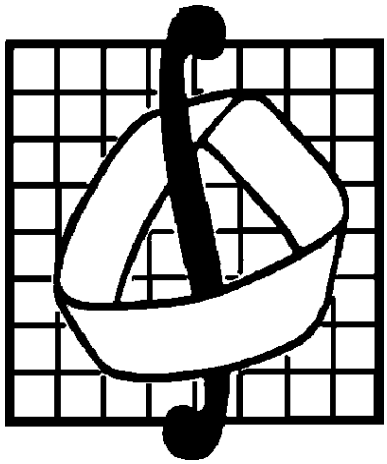
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА



Механико-математический факультет

МАТЕРИАЛЫ
IX Международной конференции "Интеллектуальные системы и компьютерные науки"

(23-27 октября 2006 г.)

ТОМ 2
часть 2

Издательство механико-математического факультета МГУ
2006

УДК 519.95, 519.14, 519.1, 519.6

Издание осуществлено при поддержке
Российского фонда фундаментальных исследований
по проекту № 06-01-10-114

Материалы IX Международной конференции "Интеллектуальные системы и компьютерные науки"(23-27 октября 2006 г.), том 2, часть 2. - М.: Изд-во механико-математического факультета МГУ, 2006.

Сборник содержит работы участников IX Международной конференции "Интеллектуальные системы и компьютерные науки", проходившей на механико-математическом факультете МГУ им. М. В. Ломоносова с 23 по 27 октября 2006 г. при поддержке Российского фонда фундаментальных исследований (проект № 06-01-10-114). Сборник адресован научным сотрудникам, преподавателям, аспирантам и студентам, работающих и интересующихся тематикой математических проблем теории интеллектуальных систем и их приложений.

Научное издание

МАТЕРИАЛЫ IX МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ "ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И КОМПЬЮТЕРНЫЕ НАУКИ"(23-27 октября 2006 г.) Под общей редакцией академика Садовничего В. А., проф. Кудрявцева В. Б., проф. Михалева А. В.

В составлении и редактировании сборника принимали участие: Строгалов А. С., Носов В. А., Уварова Т. Д., Холоденко А. В., Галатенко А. А.

Ответственный за выпуск Строгалов А. С.

© Механико – математический факультет МГУ, 2006

Прогнозирование по трендовой модели временного ряда, не подчиняющегося нормальному распределению[1]

Кульчицкий К. А., *e-mail: cd82@list.ru*
Ставропольский государственный университет,
355042 г. Ставрополь ул. 50 лет ВЛКСМ 40/3 кв.45,

Традиционные методы прогнозирования временных рядов [1] в большинстве своем состоят в разложении последовательности временного ряда на составляющие компоненты, прогнозировании отдельно каждой составляющей и объединении полученных результатов.

В общем случае последовательность временного ряда может быть представлена в виде суммы четырех составляющих [1]:

1. систематическая составляющая, показывающая в среднем характер изменения прогнозируемого явления во времени и называемая трендом;
2. колебания около тренда с возможно изменяющейся частотой;
3. сезонные колебания со значительно большей частотой, чем колебания около тренда;
4. случайная составляющая.

Надежный прогноз возможен лишь в случае, когда длина n временного ряда достаточна для получения достоверных выводов относительно характера изменения во времени каждой составляющей ряда. Зачастую, на практике, длина m должна составлять порядка сотен единиц, что является не всегда допустимым при прогнозировании экономических показателей предприятия, поскольку современное предприятие находится в условии воздействия множества внешних и внутренних факторов, учесть которые на протяженном временном интервале, порой, не представляется возможным. В данном случае актуальной становится задача прогнозирования короткого временного ряда [2] порядка 10 – 20 значений.

Другое, не менее важное противоречие заключается в том, что большинство разработанных методов прогнозирования временных рядов применимы в случае нормального распределения. На практике, зачастую, данное требование не выполняется. Эмпирически доказано, что большинство экономических показателей подчинены распределению Парето[3]. Рассмотренная далее методика основывается на том, что временной ряд подчиняется распределению Парето.

Общая схема методики представлена на рисунке № 1.

Первый этап – это сбор информации о каком-либо показателе на определенном временном интервале и представление его в виде временного ряда.

На втором этапе на основе алгоритма R/S анализа [4] проводится проверка на трендоустойчивость и наличие фрактальной структуры временного ряда. Наличие фрактальной структуры временного ряда является косвенным признаком того, что данный временной ряд подчиняется распределению Парето. Наличие трендоустойчивости говорит о склонности исследуемого временного ряда и в дальнейшем следовать установившейся ранее тенденции.

На третьем этапе производится выбор функции для трендовой модели и оцениваются параметры модели. Далее производится проверка адекватности модели одним из существующих методов. В случае подтверждения адекватности модели построенную модель можно использовать для получения прогнозных значений исходного временного ряда.

Для определения прогнозируемости короткого временного ряда будет рассматриваться такая характеристика временного ряда как трендоустойчивость, которая может быть получена в результате фрактального анализа [3,4]. Наличие трендоустойчивости говорит о том, что временной ряд подчиняется определенной закономерности. Определение закономерности в свою очередь позволяет прогнозировать очередные значения временного ряда.

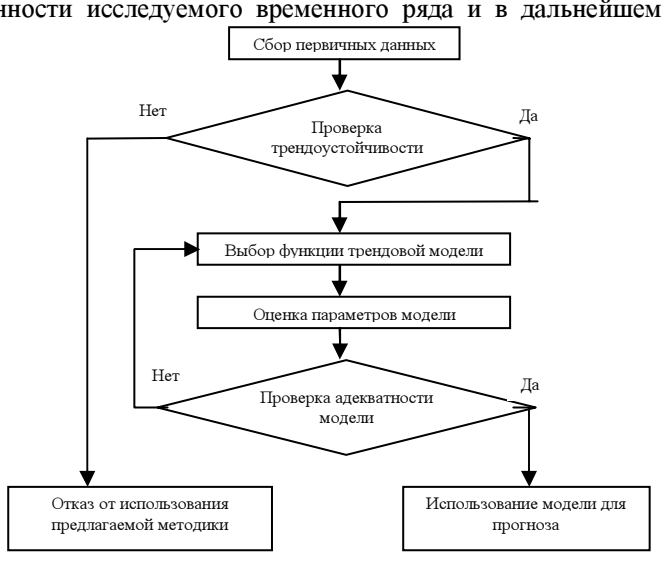


Рис. 1. Схема методики

Основным инструментарием фрактального анализа является алгоритм R/S анализа [4,5].

[1] Работа выполнена при поддержке РФФИ, проект № 06-01-00020а

В временном ряде Z последовательно выделяем его начальные отрезки $Z_\tau = z_1, z_2, ..., z_\tau$, $\tau = 3,4,...,n$, для каждого из которых вычисляем текущее среднее $\bar{z}_\tau = \frac{1}{\tau}\sum_{i=1}^{\tau} z_i$. Далее для каждого фиксированного Z_τ, $\tau = 3,4,...,n$ вычисляем накопленное отклонение для его отрезков длины t: $X_{\tau,t} = \sum_{i=1}^{t}(z_i - \bar{z}_\tau)$, $t = \overline{1,\tau}$. После чего вычисляем разность между максимальным и минимальным накопленными отклонениями $R = R(\tau) = \max_{1\le t\le\tau}(X_{\tau,t}) - \min_{1\le t\le\tau}(X_{\tau,t})$, которую принято называть термином "размах R". Этот размах нормируется, т.е. представляется в виде дроби R/S, где $S = S(\tau)$ - стандартное отклонение для отрезка временного ряда Z_τ, $3 \le \tau < n$.

Показатель Херста $H = H(\tau)$, характеризующий фрактальную размерность рассматриваемого временного ряда и соответствующий ему цвет шума, получаем из соотношения $R/S = (a*\tau)^H$, $H = H(\tau)$. Логарифмируя обе части этого равенства и полагая значение $a = 1/2$, получаем последовательность декартовых координат (x_τ, y_τ) точек H - траектории ординаты которых $y_\tau = H(\tau) = \frac{\log(R(\tau)/S(\tau))}{\log(\tau/2)}$ и абсциссы $x_\tau = \log(\tau)$, $\tau = 3,4,...,n$.

Требуемая для фрактального анализа ряда *Z R/S*-траектория представляется в логарифмических координатах последовательностью точек, абсциссы которых $x_\tau = \log(\tau)$, а ординаты $y_\tau = \log(R(\tau)/S(\tau))$. Соединяя отрезком соседние точки (x_τ, y_τ) и $(x_{\tau+1}, y_{\tau+1})$, $\tau = 3,4,...,n-1$, получаем графическое представление *R/S*-траектории (*H*-траектории) в логарифмических координатах (в обычных декартовых координатах).

Предположим, что известны предыстория прогнозируемого явления в виде последовательности результатов наблюдений $\tilde{z}_t, t = 1,..., n,$ и модель

$$F^*(t;\theta) = (\theta;\varphi(t)). \tag{1}$$

Задача состоит в отыскании прогнозной последовательности, наиболее согласованной с результатами наблюдений.

Соответствие модели (1) результатам наблюдений определяется тем, насколько вычисленные по модели значения временного ряда близки к наблюдаемым [6]. В качестве меры близости примем сумму

$$J(\theta) = \sum_{t=1}^{m} \Omega(\tilde{z}_t - (\theta, \varphi(t))), \tag{2}$$

где Ω – некоторая строго выпуклая функция, например, квадрат или модуль вещественного числа [7].

В результате приходим к задаче минимизации

$$\min_{\theta_i} J(\theta). \tag{3}$$

Решая задачу (2), (3), находим вектор оценок параметров тренда $\hat{\theta} = (\hat{\theta}_1, ..., \hat{\theta}_K)$, а затем, строим прогнозную последовательность

$$\hat{z}_t = (\hat{\theta}, \varphi(t)), t = n+1, ..., n+m. \tag{4}$$

Построим доверительный интервал I_p, соответствующий доверительной вероятности *p*. Эмпирически доказано, что экономические временные ряды подчиняются распределению Парето [3]. И в данном случае будем считать, что значения временного ряда z_i подчиняются распределению Парето.

Функция распределения Парето: $F(x) = \begin{cases} 1 - x^{-a}, x \ge 1 \\ 0, x < 1 \end{cases}$ $a > 0$.

Вероятность *p* попадания $\hat{z}_t$ в интервал (α, β) будет определяться формулой:

$$P(\alpha < z_t < \beta) = F(\beta) - F(\alpha), \tag{5}$$

где *F(x)*- функция распределения Парето.

Выражение (5) может быть переписано следующим образом:

$$P(|\hat{z}_t - z_t| < \varepsilon) = p,$$
$$P(|\hat{z}_t - z_t| < \varepsilon) = F(\hat{z}_t + \varepsilon) - F(\hat{z}_t - \varepsilon).$$

В случае, когда каждый элемент $z_i > 1$ получаем:

$$P(|\hat{z}_t - z_t| < \varepsilon) = (\hat{z}_t - \varepsilon)^{-a} - (\hat{z}_t + \varepsilon)^{-a} = (\hat{z}_t - \varepsilon)^{-\frac{1}{H}} - (\hat{z}_t + \varepsilon)^{-\frac{1}{H}},$$

где $a = \frac{1}{H}$ - фрактальная размерность пространства вероятностей, характеризующая толщину хвостов в функции плотности вероятности.

Таким образом, величина z_t с вероятностью p будет попадать в доверительный интервал, который равен $I_p = (\hat{z}_t - \varepsilon, \hat{z}_t + \varepsilon)$.

Саму величину ε можно найти из выражения $(\hat{z}_t - \varepsilon)^{-\frac{1}{H}} - (\hat{z}_t + \varepsilon)^{-\frac{1}{H}} = p$.

Для проверки адекватности полученной модели экспериментальным данным необходимо провести анализ остатков (остаточных ошибок) модели [8]. Остатком называется разность между исходным и предсказанным значением временного ряда: $\eta_i = z_{i+n} - F(\hat{\theta}, \varphi(i+n)), i = 1,...,k$.

Модель (1) можно считать удовлетворительной, если остатки независимы и распределены (приблизительно) по нормальному закону.

Для проверки независимости остатков существует множество различных тестов: критерий восходящих и нисходящих серий; критерий серий, основанный на медиане выборки; критерий квадратов последовательных отношений. Они пригодны при любом законе распределения. Также некореллированость остатков при расчете автокорреляционной функции будет говорить об их статистической независимости. Сама автокорреляционная функция может быть рассчитана по следующей формуле [8]: $\rho(l) = \sum_{i=1}^{\tau-l} \eta_i \eta_{i+l} \Big/ \sum_{i=1}^{\tau} \eta_i^2, l = 0,1,...,K < \tau$,

где τ - глубина полученного прогноза. В предположении о независимости остатков, значения $\rho(l)$ должны быть распределены по нормальному закону и лежать в интервале $\pm 2\sigma_\eta / \tau$ с вероятностью 0,95.

Для проверки нормальности распределения остатков можно использовать, например, критерий χ^2.

Список литературы

1. Введение в анализ макроэкономической динамики переходного периода / В. А. Бессонов. – М., 2003. – 151 с.

2. Гречкин В. А., Кульчицкий К. А., Николаева А. В. Актуальность прогнозирования коротких временных рядов // XIV Международная конференция "Математика. Экономика. Образование". IV международный симпозиум "Ряды Фурье и их приложения". Труды. Ростов н/Д, изд-во ООО "ЦВВР", 2006. - С. 134 - 135.

3. Петерс Э. Хаос и порядок на рынках капитала. Новый аналитический взгляд на циклы, цены и изменчивость рынка. – М.: Мир, 2000. – 333 с.

4. Перепелица В. А., Беляков С. С., Овчаренко Н. Ф. Фрактальный анализ временных рядов объемов инвестиций в основной капитал региона // Региональное приложение к журналу «Современные наукоемкие технологии». – 2004. - №2. – С.19-23.

5. Инструментарий фрактального анализа для выявления косвенных факторов, выявляющих несоответствие налогового потенциала прибыли предприятия и его фактических налоговых отчислений / Кульчицкий К. А., Николаев А. В., Овчаренко Н. Ф.; Ставропольский государственный университет – Ставрополь, 2005.– 45 с. – Библиогр.: 14 назв.– Рукопись деп. в ИНИОН РАН № 59454 07.10.2005

6. Прогнозирование временных рядов по разнородной информации / В.Б. Головченко. – Новосибирск: Наука. Сибирская издательская фирма РАН, 1999. – 88 с.

7. Гречкин В. А., Кульчицкий К. А., Николаев А. В. Метод прогнозирования коротких временных рядов // VII международная конференция "Кибернетика и высокие технологии XXI века". Воронеж, 16-18 мая 2006 г. - Воронеж: Изд-во ВГУ, 2006. - С. 853 - 859.

8. Безручко Б. П., Смирнов Д. А. Математическое моделирование и хаотические временные ряды. Саратов: ГосУНЦ «Колледж», 2005. – 320 с.

A-тестовая система и оценка знания с помощью метрических функций в пространстве F-чисел

Курбанов Ф. И., *fuadig@yahoo.com*
Бакинский Государственный Университет

Introduced new type of testing system. The main device built by means of fuzzy numbers, linguistic variable and metric functions in area of fuzzy numbers included earlier by authors. Was shown effectiveness suggested form of testing in comparison with usual. Was prepared program facilities for realization of suggested testing system.

Key words: F - numbers, metric, metric function, testing system, linguistic variable, functional, knowledge estimation, strategy.

В процессе обучения, как известно, основной задачей обучающего является эффективная, качественная и полная передача собственного знания и навыков обучаемым. Для повышения качества обучения одним из основных принципов является наличие обратной связи у обучающего. Другими словами, при передаче новой порции знания необходимо оценивать и контролировать усвоенное, к текущему моменту времени, знания. Объем и глубина знания, которую обучающий намерен передать при очередном занятии процесса обучения, существенно зависит от знания обучаемого. Для контроля знания существуют много различных методов [1]. В последнее время довольно широкое распространение получил метод тестирования закрытого типа, как метод объективного оценивания знания у обучаемых.

Обычно в таких тестовых системах обучаемым задается n вопросов ($1 \leq n \leq 100$), каждому из которых приводится k ($2 \leq k \leq 5$) возможных вариантов ответа. Причем, всегда только один из этих вариантов бывает правильным. Таким образом, в таких тестовых системах обучаемый за правильный ответ получает 1, а за ложный ответ 0 баллов. Другими словами, правильность каждого вопроса оценивается элементами множества L состоящего из двух элементов:

$$L = \{0,1\}$$

Пусть Q_i ($1 \leq i \leq n$) есть i-ый вопрос тестовой системы. Множество всех вопросов обозначим $Q = \{Q_i\}_{i=1}^{n}$. Тогда тестовую систему можно рассматривать как некоторый функционал f осуществляющий следующее отображение:

$$f : Q \rightarrow L$$

Итоговая оценка знания обучаемого в таких тестовых системах вычисляются применением некоторого алгоритма к значениям функционала f. Например:

$$S = \sum_{i=1}^{n} f(Q_i)$$

Характерной особенностью таких тестовых систем является принадлежность значений функционала f множеству L ($f(Q_i) \in L$).

В дальнейшем набранные обучаемым баллы обрабатываются одним из существующих [1] алгоритмов. Таким образом получается итоговая оценка для знания обучаемого.

Обозначим через ZO знание обучаемого, которую необходимо оценивать. После применения указанного выше метода знание обучаемого будет выражено некоторым натуральным числом. Другими словами, обычный метод закрытого тестирования (crisp test) есть оператор, который отображает знание в множество состоящего из натуральных чисел и нуля:

$$CT : ZO \rightarrow N \cup \{0\}, \qquad (1)$$

здесь N означает множество всех натуральных чисел.

Возникает естественный вопрос: На сколько полученное в результате тестирования натуральное число адекватно отражает знание обучаемого?

Известно, [2], [3], что рассмотренный выше crisp тестирование обладает некоторыми недостатками. Для устранения основных недостатков этого вида тестирования в работе [2] предложено A – тестовая система, которая более адекватно позволяет оценивать знание обучаемого.

A – тестовая система является синтезом двух видов тестирования. Первое из них, так называемое F (fuzzy) тестирования [4], используя преимущества fuzzy технологий позволяет моделировать неопределенности возникающие в оценке истинности высказываний обучаемого. Это позволяет устранять ряд недостатков crisp тестирования. Другой вид тестирования [5], так называемое P тестирование опирается на идеи теории вероятностей и помогает оценивать другой вид неопределенностей знания обучаемого. Здесь моделируется распределение мнения обучаемого

относительно вариантов ответа на заданный вопрос. Объединение этих двух видов тестирования в одной системе А-тестирования позволяет более точно и адекватно оценивать знание обучаемого. Так как при этом учитывается различные грани знания, которые не рассматриваются в crisp тестировании.

Прежде всего отметим, что обычная crisp тестовая система CT, обладает следующими недостатками:

1. Несмотря на то, что подготовленные вопросы почти всегда имеют разные уровни сложности, верные ответы на них всегда оцениваются одинаковыми баллами.

2. Ответы на вопросы, которые считаются неверными отличаются тем, что некоторые из них могут быть гораздо ближе к истине чем другие. CT система не учитывает и не может отличить обучаемых, которые более близко подошли к правильному ответу.

3. Часто мнение обучаемого распределяется между несколькими вариантами ответов. Вынужденный выбирать только одну из этих ответов обучаемый может случайно допустить ошибку. CT система не учитывает распределение мнения обучаемого над вариантами ответов.

4. CT система является неустойчивой системой. Другими словами, малое отклонение от истины сразу приводит к нулевому результату.

Рассмотрим конкретные примеры:

I. Следующие примеры взяты из банка задач ГКПСА (Государственная Комиссия приема студентов Азербайджана)

A) Вычислить: $\frac{1}{4} - \frac{1}{5} - \frac{1}{6}$

B) При каких значениях параметра a уравнение $2a|x| + 2 = 2ax - 3$ имеет решение?

Сразу заметно, что задача B) гораздо труднее чем задача A). Для решения задачи A) достаточно знание стандартных процедур вычисления дробей. Тогда как, задача B) требует более детального анализа. Здесь требуется нестандартное мышление. Однако правильное решение обоих задач оцениваются одинаковыми баллами. А это значительно отражается в итоговом балле S. Таким образом, неучитывание уровней сложности тестов приводит к неадекватности итоговых оценок уровня знания обучаемых.

II. В какое время дня англичане при встрече пользуются словом «Good afternoon?»

Варианты ответов:

1. С 9^{00} до 12^{00}
2. С 11^{00} до 14^{00}
3. С 12^{00} до 17^{00}
4. С 15^{00} до 18^{00}
5. С 19^{00} до 22^{00}

Какое из этих ответов отражает действительность? Пусть правильный ответ расположен на третьей строке. Другими словами, она отражает полную истину. Пятый вариант ответа очевидно, что совершенно неверно. Но, для других вариантов этого нельзя утверждать, поскольку у них есть верная часть. Другими словами, они содержат долю истины. Обучающие, которые выбирают ответы 4 и 5 обладают разными уровнями знания. Обычная CT система тестирования не учитывает долю истины имеющиеся в вариантах ответа. Тогда как, для адекватного и точного оценивания знания необходимо отличить обучаемых выбирающих совсем неправильный ответ от тех кто выбирает частично правильный ответ.

Допустим, что мнение обучаемого распределяется следующим образом. Он считает верными либо второй, либо третий варианты ответов. Остальные варианты отбрасываются. При этом, по мнению обучаемого, 51% возможно верен второй, а 49% третий вариант ответов. Таким образом, выбирая второй ответ в CT системе он получает ноль баллов. Тогда как, другой обучаемый со стопроцентной уверенностью выбравший 2-ой вариант также получит ноль баллов. Легко заметить, что их понимание и знания предметной области существенно отличаются.

III. Чему равно число π, отношение длины окружности к его диаметру?

Варианты ответов:

1. 3.14159
2. 3.14159….
3. 4
4. 15
5. 10000

Как известно, правильным является второй из предлагаемых вариантов. Но разница между первым и вторым вариантом меньше чем 10^{-5}. Допуская столь незначительную ошибку, обучаемый получает ноль баллов. Отметим, что можно построить тест, где такого рода ошибка будет меньше чем 10^{-n} для любого натурального n. Тогда как, другой обучаемый, который не имеет ни малейшего представления о числе π, выбирает пятый вариант и также получает ноль баллов.

С другой стороны, допустим, что обучаемый, отбрасывая все другие варианты оставил только первые два варианта. Но он колеблется и не знает какой из них выбирать. При этом у него сформировалось мнение, по которому 51% - возможно верно первый вариант, и 49% - возможно верно

второй вариант. Таким образом, выбирая первый вариант, обучаемый, опять получит ноль баллов по СТ системе. Другой обучающий, полностью уверенный в верности пятого варианта, также получит ноль баллов.

Указанные выше примеры, в общем демонстрируют основные слабые стороны обычной тестовой системы. Особенно заметна неустойчивость таких систем. С их помощью невозможно достаточно точно оценить знания. Они дают только грубую оценку.

Построим тестовую систему, которая не имеет вышеуказанных недостатков. Для этого вместо обычной СТ системы, сначала будем строить FT и PT системы. В дальнейшем, их объединяя получим AT систему. При построении этих систем будем использовать понятия fuzzy множества, лингвистической переменной и F чисел [6],[7].

Определение 1: Пусть, U множество произвольной природы (в дальнейшем будем называть его универсумом). Нечетким или fuzzy множеством Ã называется совокупность упорядоченных пар вида <x, $\mu_{\tilde{A}}(x)$>, где $x \in U$, $\mu_{\tilde{A}}(x) \in [0,1]$ для всех x. Функция $\mu_{\tilde{A}}(x)$ называется функцией принадлежности fuzzy множества Ã ($\mu : U \rightarrow [0,1]$).

Определение 2: Лингвистической переменной, называется переменная, значениями которой могут быть слова и предложения естественного или искусственного языка.

Определение 3: F-числом называется fuzzy множество Ã универсум которого есть множество вещественных чисел R. Кроме того, выполняются следующие условия :

1. Ã – нормально, т.е. существует $x_o \in R$ такое, что $\mu_{\tilde{A}}(x_o) = 1$.
2. Ã – выпукло, т.е. для любого x, $y \in R$ и $\lambda \in [0,1]$ справедливо неравенство: $\mu_{\tilde{A}}(\lambda x+(1-\lambda)y) \geq \min(\mu_{\tilde{A}}(x), \mu_{\tilde{A}}(y))$.
3. Носитель множества Ã- ограничено, т.е. существуют $A,B \in R$ такие, что $A \leq x \leq B$, если $x \in S_{\tilde{A}}$, где $S_{\tilde{A}} = \{x| \mu_{\tilde{A}}(x) > 0, x \in R\}$.

Определение 4: F – тестовой системой (FT), будем называть тестовую систему в которой:

A) Уровень сложности различных вопросов тестовой системы, оцениваются обучающим с помощью F чисел.

B) Правильность вариантов ответов, для каждого вопроса, оцениваются обучаемым и обучающим, с помощью F чисел.

C) Итоговая численная оценка знаний обучаемого вычисляется по некоторому алгоритму, которая обрабатывает все F числа, полученные в предыдущих пунктах.

В общем случае, F тестирование можно рассматривать как некоторый оператор FT, отображающий знание обучаемого в множество положительных вещественных чисел.

$$FT : Z0 \rightarrow R^{+} \qquad (2)$$

По сравнению с отображением (1), отображение (2), дает более точную оценку знаний обучаемого. Отметим, что результат работы алгоритма, указанного в пункте C) определения 4 будет в виде F числа. Однако, как известно, [9], F число может быть дефаззифицировано с помощью одного из известных методов. В соотношении (2) имеется в виду, что результату алгоритма применена дефаззификация. Однако, более точный и адекватный результат несомненно будет в виде F числа.

В процессе реализации F тестирования, обучаемый и обучающий, каждый в отдельности субъективно оценивают долю истины для каждого из вариантов ответа на вопросы тестовой системы. Кроме этого, обучающий оценивает уровень сложности каждого вопроса. Другими словами, в этом процессе оценивания отдельные объекты тестовой системы получают некоторый вес значимости. Эти веса могут быть, как в виде обычного вещественного числа, так и F числа. Определим отдельные разновидности F тестирования.

Определение 5: F тестированием называется тестирование вида F1, если все используемые в нем веса, вещественные числа.

Определение 6: F тестированием называется тестирование вида F2, если хотя бы один из использованных весов есть F число.

Отметим, что F тестирование вида F2 – более широкое понятие, чем F1. Так как любое вещественное число можно рассматривать, как F число, носитель которого состоит из одной точки, то можно сказать, что F тестирование вида F1 – есть частный случай F тестирования вида F2.

Таким образом, на первом этапе F тестирования обучающий или эксперт оценивает сложности всех вопросов тестовой системы и расставляет веса, S_i ($1 \leq i \leq n$), для каждого из них.

На втором этапе, обучающий или эксперт, по своему усмотрению оценивает долю истины имеющийся в каждом из вариантов ответа на вопросы тестовой системы.

Ответы на i-й вопрос Q_i тестовой системы можно рассматривать, как значение некоторой лингвистической переменной L_i . В пространстве рассуждений обучающего, каждая из значений переменной L_i имеет значение истинности, которая достаточно хорошо формализуется с помощью F чисел. Обозначим эти числа $\tilde{E}_{ij}$ ($1 \leq i \leq n$, $1 \leq j \leq k$). На втором этапе, обучающий задает все эти числа.

На третьем этапе, обучаемый оценивает ответы на заданные вопросы тестовой системы, ставит соответствующие значения истинности для каждого ответа. Обозначим эти числа $\tilde{O}_{ij}$ ($1 \leq i \leq n$, $1 \leq j \leq k$). Они

будут соответствовать значению истинности лингвистических переменных L_i ($1 \le i \le n$) для каждого ответа, и отражать субъективное мнение обучаемого.

Численное значение знания обучаемого в FT системе, вычисляется следующей формулой :

$$\hat{S} = \sum_{i=1}^{n} (\hat{S}_i - \sum_{i=1}^{n} \sum_{j=1}^{k} \rho(\tilde{E}_{ij}, \tilde{O}_{ij})) \qquad (3)$$

Можно рассматривать и другие алгоритмы для обработки массива чисел $\hat{S}_i$, $\tilde{E}_{ij}$,$\tilde{O}_{ij}$. Формула (3) является одним из простых, наглядных и интуитивно понятных способов обработки, полученных в предыдущих этапах F чисел.

Здесь, ρ – есть метрика между F числами. Для ее вычисления можно использовать различные известные метрики в пространстве F чисел. Наибольший эффект дает следующая метрика [8] :

$$\rho(\tilde{A}, \tilde{B}) = \sum_{i=1}^{n} \beta_i \rho_i(\tilde{A}, \tilde{B}) \qquad (4)$$

где $0 \le \beta_i \le 1$, для всех i ($1 \le i \le n$), $\sum_{i=1}^{n} \beta_i = 1$

Теперь, рассмотрим, другой вид тестирования, в котором учитывается распределение мнения у обучаемого относительно вариантов ответа на заданные вопросы.

Определение 7: P тестовой системой (PT) будем называть систему, в которой:

A) Обучаемому предоставляется возможность численно распределять свое мнение относительно верности вариантов ответов на заданные вопросы.

B) Итоговая численная оценка знания обучаемого вычисляется по некоторому алгоритму, в котором обрабатываются и учитываются числа полученные в пункте A).

В общем случае, P тестирования, также как и F тестирование, можно рассматривать как некоторый оператор PT, отображающий знание обучаемого в множество положительных вещественных чисел.

$$PT : Z0 \rightarrow R^{+} \qquad (5)$$

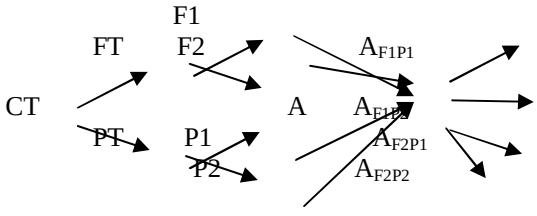
Например, можно взять какое-то фиксированное число P (например, P = 100), и предложить обучаемому распределить это число между вариантами ответов. Если обучаемый убежден в правильности одного единственного варианта ответа, то он может положить на этот вариант все имеющиеся у него шарики (Предположим, что обучаемый имеет в распоряжении P шариков). Другими словами, сделать максимальную ставку на один выбранный вариант. В других случаях, каждый вариант получит столько шариков, сколько имеет шансов быть правильным по мнению обучаемого. Например, в примере (III) первый обучаемый будет ставить 51 шариков на первый вариант, и 49 шариков на второй. Тогда как, второй обучаемый все 100 шариков поставит на пятый вариант. Обычное тестирование обоим обучаемым поставит 0 баллов. Но, PT система сможет их отличить и оценит разными количествами баллов.

Определение 8: PT система называется системой вида P1, если все используемые в нем веса – вещественные числа.

Определение 9: PT системой называется система вида P2, если хотя бы один из используемых весов – есть F число.

Определение 10: A тестовой системой (AT) будем называть систему, в которой одновременно выполняются все условия F и P тестовых систем.

Другими словами AT система – есть синтез FT и PT систем. Можно отличить 4 разновидности AT системы.



Из диаграммы видно, что существует 4 вида AT систем. Самая сильная из них система A_{F2P2} .

Таким образом, вместо обычной CT системы можно рассматривать 8 видов более адекватных тестовых систем. Из них 2 вида F, два вида P и четыре вида A тестовых систем. Несомненно, самой адекватной системой является A тестовая система, которая является синтезом F2 и P2 тестовых систем.

Численное значение знания обучаемого в A тестовых системах, можно вычислить по следующей формуле:

$$\hat{S} = \sum_{i=1}^{n} (\hat{S}_i - \sum_{i=1}^{n} \sum_{j=1}^{k} \tilde{P}_{ij} \rho (\tilde{E}_{ij}, \tilde{O}_{ij})),$$

где $\tilde{P}_{ij}$ – коэффициенты уверенности обучаемого на правильность j – го варианта ответа на i – ый вопрос. Соответственно для P1 и P2 тестовых систем имеемρ следующие соотношения:

$$\sum_{j=1}^{k} P_{ij} = P, \qquad (7)$$

$$\sum_{j=1}^{k} \tilde{P}_{ij} = \tilde{P}, \qquad (8)$$

где P заранее выбранное фиксированное число (например P=1 или P=100), $i \in [1 \ldots n]$, а $\tilde{P}$ его фаззифицированный аналог.

Здесь ρ есть метрика между F числами. Для ее вычисления можно использовать формулу (4). Метрики ρ_i $(1 \le i \le m)$ в (4) являются линейно независимыми компонентами общей метрики ρ. Каждая из них есть отдельная метрика вычисляющая расстояние между числами $\tilde{A}$ и $\tilde{B}$ в некотором смысле. Коэффициенты β_i зависят от стратегии проверки знания обучаемого. Можно сказать, что вектор $\beta = (\beta_1, \beta_2, \ldots, \beta_n)$ определяет стратегию проверки знания. Для дефаззификации F числа $\hat{S}$ можно применять различные методы [10] . В итоге мы можем получить crisp число для оценки знания обучаемого:

$$S = Def(\hat{S}) \qquad (9)$$

Здесь, Def означает оператор дефаззификации. Таким образом $\hat{S}$ и S являются численными выражениями знания ZO обучаемого полученного с помощью A тестирования. В итоге мы имеем следующее отображение:

$$AT : ZO \rightarrow R^{+} \qquad (10)$$

В процессе реализации тестирования вида F2 и P2 можно использовать либо F числа стандартного вида, либо заданные пользователем F числа произвольного вида. Для этого необходимо использовать соответствующее программное обеспечение, позволяющее произвести арифметические операции над F числами.

Необходимо отметить , что гораздо проще программно реализуется F числа следующих стандартных видов:

1. Интервальные
2. Треугольные
3. Трапециевидные
4. Гауссовые
5. Колоколообразные

Эти числа легко управляются визуально, либо с помощью параметров. Для работы с такими числами подготовлена программная система Flafa.

Тестирование реализовано в программной системе Flames.

Список литературы

1. Г. А. Атанов, Н. Н. Пустынникова «Обучение и искусственный интеллект или основы современной дидактики высшей школы». Донецк 2002.

2. Ф. И. Курбанов «Адекватное тестирование и оценка знания с помощью метрических функций». Труды II Республиканской научной конференции «Современные проблемы информатизации, кибернетики и информационных технологий». Баку 2004.

3. Ф. И. Курбанов, Н. Г. Мамедова «Модель трех векторов и A тестирование в педагогике». Тезисы докладов на Республиканской научно-практической конференции «Перспективы развития педагогической науки на современном этапе». Баку 2004.

4. Ф. И. Курбанов «F тестирование и применение метрических функций в пространстве fuzzy чисел к оценке задания тестируемых». Journal of Gauges university p.137. Баку 2004.

5. А. З. Юсифов «Тестовая система с вероятностным выбором». Journal of Gauges p.171. Баку 2004.

6. Л. Заде «Понятие лингвистической переменной и его применение к принятию приближенных решений». «Мир».Москва 1976.

7. Li-Xin Wang «A course in fuzzy systems and control». The Hong-Kong University of Science and technology – 1997.

8. Ф. И. Курбанов «Параметрические метрики в пространствах нечетких чисел» SCM 2004. Сборник докладов. Санкт-Петербург 2004.

9. Ф. И. Курбанов «Банахово пространство нечетких чисел с обобщенным центром тяжести». Современные проблемы прикладной математики. Баку – 2002.

Автоматизация информационных потоков кролиководческой фермы на базе технологии RFID

Латыпов Рустам Хафизович
д.т.н., декан факультета ВМК КГУ, профессор, 420008, Кремлевская 18, Roustam.Latypov@ksu.ru
Крамин Артем,
Казанский государственный Университет,
420110, Казань, ул. Зорге 57/29 – 167,
тел. 229-64-08, факс 229-64-08, kramin@wrsa.ru
Васис Оксана,
Казанский государственный Университет, 420066, Казань, Ямашева 31 – 486, тел. 517-81-52,
exquisite_lady@rambler.ru

Введение.

Технология RFID в настоящее время является одним самых быстрорастущих сегментов в сфере IT. Уникальные возможности бесконтактной радиочастотной идентификации по достоинству оценили крупнейшие розничные торговцы WalMart и METRO.

Основной принцип работы RFID-систем — бесконтактный обмен данными между транспондером и считывателем через радиочастотный канал. Радиочастотная метка представляет собой устройство, в состав которого входит микросхема, обеспечивающая функциональность метки, и антенна для приема/передачи радиосигнала. Каждая этикетка содержит уникальный неизменяемый номер, а так же некоторый объем памяти для хранения дополнительной информации об объекте идентификации.

Наш проект приоткрывает новые перспективы в развитии технологии RFID – сельское хозяйство. Разработанная система - это инновационный инструмент для зооинженеров кроликофермы, базирующийся на использовании технологии RFID. Система позволяет вывести на качественно новый уровень работу, связанную с зоотехническим учетом, планированием на ферме, прогнозированием, позволяет в корне изменить подход к хранению данных о поголовье фермы, истории выполненных работ, генеалогических связей поголовья.

Разработка системы ведется с 2002 года. Точкой приложения наших усилий стала одна из крупнейших кроликоферм России – «МИАКРО-КАРАТАУ». В сотрудничестве с зооинженерами этой компании мы внедряем в реальное производство нашу систему. В течение этого времени, мы прошли уже большой путь, и на сегодняшний день на 5-ти фермах республики работает сокращенная версия системы, персонал этих предприятий обучен работе на ПК и активно использует дополнительные возможности для повышения эффективности своего бизнеса.

Мы предлагаем фермерам России продукт, который выведет на новый уровень племенное кролиководство России и поможет работникам села существенно увеличить эффективность их труда.

Описание системы

Концептуальная схема

Ретро – кролиководу, работающему по старинке, приходится заниматься всем сразу: выполнять зоотехнические работы, вести генеалогические линии, заполнять паспорта, бонитировать кроликов и пр. пр. На ферме в 100 – 200 кроликов такой подход еще имеет право на существование. Но что делать, если ферма больше? Гораздо больше? Три – четыре – пять тысяч кроликов?

Мы предлагаем пользователям комплексное решение по автоматизации кролиководческой фермы. Система состоит из двух частей: оборудование и программное обеспечение. Концептуально работу

Системы можно описать следующим образом. Каждый кролик фермы получает свою уникальную радиочастотную метку (RFID), такими же метками оборудуются МИНифермы для содержания кроликов, корзинки для переноски оборудования и пр. На самой ферме развертывается беспроводная локальная сеть WiFi. Персонал фермы получает в пользование мобильные карманные компьютеры на базе Windows CE (Pocket PC). В офисе фермы устанавливаются desktop-компьютеры, на которых можно получить доступ к полной версии интерфейса Системы, в забойном цехе электронные весы, связанные с сервером Системы, в мисках с водой устанавливаются термоэлементы, которые будут контролировать температуру воды. На сервере фермы устанавливается разработанное нами приложение, которое вдыхает жизнь во все оборудование.

Система берет на себя задачи по ведению количественного учета, планированию работ, прогнозированию состояния фермы, подготовке зоотехнических отчетов, помогает наиболее эффективным образом организовать выполнение зоотехнических работ. По сути дела автоматизируется вся деятельность, связанная с информационными потоками, начиная от случки кролика до его забоя. Имеется возможность автоматически готовить информацию для сторонних бухгалтерских систем. По сути дела мы помогаем фермеру заняться его основной работой – ухаживать за животными. Все остальное сделает наша Система (рис. 1).

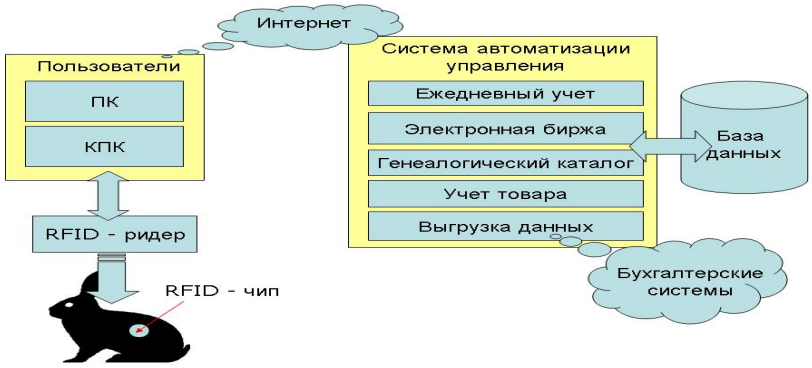


рис. 1.

Итак, каждый кролик на ферме на 1-ый день после рождения получает свой собственный RFID - метка (наклеивается на ухо). На данный момент мы ведем разработку системы, взяв за основу бесчиповые пассивные метки. Это означает, что метка не имеет собственного источника питания, и позволяет хранить только уникальный идентификатор, нанесенный на метку во время ее производства. Подобные метки являются самыми дешевыми и неприхотливыми в использовании (рабочий диапазон температур от -30ºС до +85ºС). Кроме того, подобные метки имеют маленькие размеры и весят считанные граммы, что важно при работе с однодневными животными. Идентификатор метки считывается системой с помощью RFID – ридера, который в нашем случае исполнен как простой переносной сканер. Ридер - подсоединен к ПК или КПК, и система, считав ID животного, отправляет запрос через Интернет – получая в результате некую служебную информацию в соответствии с логикой запроса.

Так, например, произведя случку кроликов, зооинженер сообщает системе (считав соответствующие номера) какие именно кролики были случены, в базе появляется информация о случке, а в каталог генеалогических связей поголовья добавляется новая запись.

Инструменты разработки. Система разработана с помощью MS Visual Studio .Net, в виде web приложения (ASP.NET). Хранение информации организовано в виде базы данных под управлением СУБД MS SQL Server 2000, что делает Систему легко масштабируемой. Отдельные части функциональности вынесены в виде веб-сервисов, что позволяет продолжать разработку и тестирование системы во время ее работы, распределить поддержку системы на нескольких независимых администраторов, снизить нагрузку на главный сервер системы, перенеся вычисления связанные с обработкой больших объемов информации на другие сервера.

Племенное разведение

В российском кролиководстве существует более глобальная проблема. За последние десять лет число племенных репродукторов упало в несколько раз и составляет на январь 2005 год – 9 хозяйств (в 1989 году только на территории Татарстана их было больше десятка!). Генеалогический учет даже на этих фермах ведется из рук вон плохо, дедовскими методами – в амбарных тетрадях. Породы постепенно вырождаются, потомство мельчает. Выход из создавшейся ситуации один – создание глобального (федерального масштаба) генеалогического каталога кроликов России.

Именно тут на помощь приходит разработанная нами Система, а ее модуль генеалогического учета. Эта та часть функционала, которая позволит объединить все фермы и скоординировать усилия по племенной работе. Оперативно получая архив данных о генеалогических связях поголовья, зооинженер сможет выбрать партнера для обмена поголовьем, операторы оперативно определят потребность фермы в свежей крови, покупатель поголовья определит – где и когда можно отдать на случку купленных животных. Генеалогическая подсистема – максимально использует возможности интернет. Фермы – разнесенные географически на сотни километров – будут получать всю необходимую информацию мгновенно.

Подобная организация племенного дела по кролику уже работает на территории Татарстана. Следующий шаг – экспансия по всей территории России.

Функциональность Системы

Если остановиться на главном, то наша система это инструмент, позволяющий:

- Полностью автоматизировать процесс создания зоотехплана;
- Хранить в удобной и легкодоступной форме данные о текущем поголовье фермы и постройках;
- Организовать обмен информацией между зоотехниками различных ферм (что очень важно в условиях информационного голода в кролиководстве в целом);
- Автоматизировать компьютерный учет подавляющего большинства операций на ферме – рассадки, отсадки, переселения, перекидки молодняка вязки, случки и пр.;
- Переложить на компьютер заботы о поддержании оптимальной температуры гнездовых отделениях и температуры в мисках с водой;
- Подготовить необходимую документацию для третьих фирм (племенные свидетельства, зоотехпланы и пр.);
- Хранить данные о генеалогических связях поголовья;
- Автоматизировать процесс списания и учета готового продукта на складе, включая подготовку сопроводительных документов для продажи (с автоматическим взвешиванием и пр.);
- Подготовить сопроводительные документы на мясо согласно новым правилам ЕС;
- Организовать электронную мясную биржу – возможность подготовки оптовой партии мяса, с последующей перепродажей;
- С точки зрения потребителя, оптимизируется система сбыта – вводится в использование продажа меха и мяса через Internet.
- Если взглянуть глазами руководителя и хозяина фермы, программа позволяет координировать работу всей фермы (выяснить, сколько корма требуется закупить, спрогнозировать убой и продажи), составить отчеты и статистики при помощи наглядных таблиц и диаграмм, а также спрогнозировать развитие всего бизнеса.
- Организовать глобальный генеалогический каталог федерального масштаба;

К тому же все эти возможности доступны не только с настольных компьютеров, но и с мобильных устройств - Pocket PC.

Мы уверены, что в течение ближайших 3-4 лет Система станет стандартом де-факто для кроликоферм России.

Заключение.

Аналогов данному предложению на мировом рынке не существует. А значит, не существует и конкурентов.

В тоже время, как на территории Татарстана, так и всей России имеется большое число мелких производителей кроличьего мяса, нуждающихся в племенном, чистокровном поголовье, квалифицированной консультационной помощи, кормах и организованном сбыте своей продукции. Внедрение нашей Системы поможет решить многие из этих проблем, и поможет сельчанам заняться своим основным делом.

В своей работе мы ощущаем поддержку Министерства сельского хозяйства Татарстана и Министерства труда и занятости, они знают о проблемах в нашей предметной области, и готовы сотрудничать в целях улучшения состояния в российском кролиководстве и обстановки на селе. Мы планируем, что к 2007 году возможностями Системы будут пользоваться более около 200 работников на крупных кроликофермах и 200 – 300 мелких фермерских хозяйств.

Список литературы

1. AIM, Inc. (2004, January). RFID connections – ROI of privacy invasion. http://www.aimglobal.org/technologies/rfid/resources/articles/jan04/0401-roispy.htm

2. K. Mayfield «Radio ID Tags: Beyond Bar Codes». *Wired News*. http://www.wired.com/news/technology/0,1282,52343,00.html 2002, May 20

3. RFID journal. RFID Business Applications. http://www.rfidjournal.com/article/articleprint/1334/-1/129/

4. A. Kramin, O. Vasis, R. Latypov "The System of administration for rabbit farms "MIAKRO"/ Proceedings of 4th International Symposium on Intelligent Manufacturing Systems, September 6-8 2004, pp.318-325

5. И. Михайлов «Акселерационные методы в кролиководстве» 2003, С.-Петербург, «Путь»

6. М. Ховард, Д. Леблан «Защищенный код» 2003, Москва, Microsoft Press

Автоматизация распределения учебной нагрузки на кафедре

Макатов В. З.,
инженер-программист, Казахстанский филиал МГУ им. М.В. Ломоносова,
Республика Казахстан, 010010, г. Астана, ул. Мунайтпасова, 7, каб. 717
тел.(факс):8(3172) 35-15-04, E-mail: mak-vz@yandex.ru
Фураева И. И.,
к.ф.-м.н., доцент кафедры
вычислительной техники и информационных систем
Евразийского национального университета им. Л.Н. Гумилева
Республика Казахстан, 010010, г. Астана, ул. Мунайтпасова, 5, каб. 309
тел.: 8(3172) 27-41-43.

С введением кредитной технологии обучения в Казахстане, изменились принципы управления учебным процессом в вузе. В некоторых вузах еще сохраняется традиционная технология подготовки специалистов лишь на отдельных факультетах, когда другие факультеты уже перешли на новую форму обучения студентов. Стремление Казахстана войти в международное образовательное пространство диктуется все повышающимися требованиями к выпускаемым специалистам с высшим образованием, с одной стороны, а с другой, требует признания национальных образовательных программ и обеспечения мобильности студентов и профессорско-преподавательского состава, их конкурентоспособности в современных условиях.

Штатное расписание профессорско-преподавательского состава кафедры в высшем учебном заведении формируется из расчета учебной нагрузки. До введения кредитной технологии обучения процесс распределения учебной нагрузки происходил на основании пятилетних (в основном) учебных планов, из которых создавался перечень учебных дисциплин для изучения по отдельной специальности и курса. Суммарное количество учебных дисциплин, разделенное по семестрам, с указанием количества учебных часов на все виды занятий поступало из учебной части на кафедру для последующего распределения этой нагрузки между преподавателями кафедры. Процесс распределения осуществлялся с учетом индивидуальных особенностей и статуса преподавателя кафедры, исходя из некоторых предпочтений и установленной минимальной ставки учебной нагрузки. В итоге распределенная нагрузка между преподавателями на кафедре поступала в учебную часть (как закрепленная) и на основании этой информации формировалось расписание учебных занятий.

Процесс распределения учебной нагрузки на кафедре для традиционной (или линейной) системы обучения студентов можно представить следующей обобщенной схемой (Рис. 1).

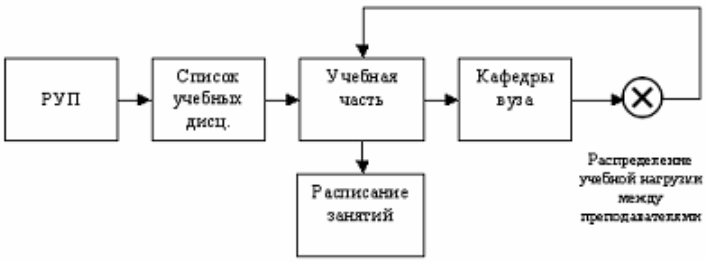


Рис. 1. Схема распределения учебной нагрузки для линейной технологии обучения

При переходе к кредитной технологии обучения процесс распределения учебной нагрузки на кафедре изменяется. Связано это с тем, что формирование учебной нагрузки для отдельного преподавателя зависит от выбора самих студентов. Студенты в кредитной технологии получили право выбирать себе учебные дисциплины из обязательной компоненты дисциплин, где они могут выбирать себе преподавателя для прохождения курса обучения, а также из компоненты дисциплин по выбору студентов, где они выбирают не только преподавателя, но и отдельный курс по своему усмотрению. На основании выбора учебных дисциплин, студенты формируют индивидуальную траекторию обучения. Вопрос распределения учебной нагрузки на кафедре переходит в плоскость определения сформировавшихся учебных подгрупп для отдельного преподавателя с определением соответствующей учебной нагрузки. Но так, как выбор учебных дисциплин основан на перечне курсов в учебном рабочем плане, учебная нагрузка формируется на основании рабочих учебных планов (РУП) и индивидуальных траекторий обучения студентов. Студенты выбирают дисциплины благодаря представленным силлабусам, разработанными ведущими преподавателями. А так, как ведущий преподаватель вправе сам формировать себе ассистирующую команду для проведения лабораторных, семинарских и/или практических занятий, то вопрос распределения учебной нагрузки при кредитной форме обучения затрагивает определение учебной нагрузки для ассистирующих групп преподавателей и дисциплины, которые остались не распределенными по каким-то причинам. Причем ведущий преподаватель может и сам выступать в роли ассистента для каких-нибудь других курсов. Здесь же возникают вопросы перераспределения учебной нагрузки для тех преподавателей, у которых количество желающих изучать дисциплину студентов оказалось ниже установленного минимума, т.е. подгруппа студентов для изучения дисциплины не набралась и должна быть расформирована, а студенты обязаны сделать выбор другой дисциплины. Процесс распределения учебной нагрузки на кафедре для кредитной технологии обучения можно представить следующей обобщенной схемой (Рис. 2).

Как видно из представленных схем, вопрос распределения учебной нагрузки для кредитной технологии существенно претерпел изменения. Если раньше учебная нагрузка для преподавателя формировалась путем ее распределения на кафедре, то в кредитной технологии этот процесс основан на выборе студентов и на добавлении той части нагрузки, которая может быть дополнена из ассистирующей компоненты и компоненты, где учебную нагрузку нужно перераспределять или из нераспределенного остатка дисциплин. Нераспределенный остаток дисциплин может появиться только для дисциплин обязательной компоненты.

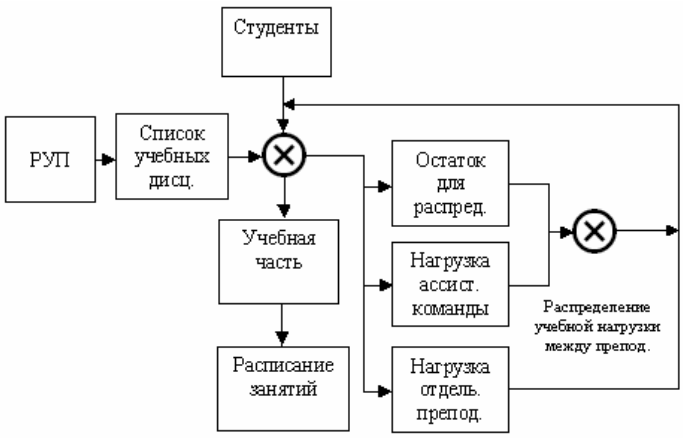


Рис. 2. Схема распределения учебной нагрузки для кредитной технологии

Введенная в Казахстане кредитная технология обучения студентов предполагает, что учебная нагрузка по каждой дисциплине состоит из следующих компонентов:

- количество лекционных часов;
- количество часов на лабораторные занятия;
- количество учебных часов на практические/семинарские занятия;
- количество часов на СРСП (самостоятельную работу студентов с преподавателем);
- количество часов на СРС (самостоятельную работу студентов).

При чем первые четыре компонента должны быть представлены в сетке учебного расписания занятий. В отличие от линейной технологии обучения, в нагрузку для кредитной технологии не входят запланированные часы на зачеты, экзамены и консультации. Эту работу осуществляет специальный отдел офис-регистратора в вузе.

В общем виде нагрузку отдельного преподавателя для кредитной технологии обучения можно представить следующей формулой:

$$N_{преп.} = N_{выб} + N_{асс} + N_{ост}$$

где, $N_{выб}$- учебная нагрузка преподавателя, сформированная на основании выборов студентами, определяется как: $N_{выб} = \sum_{i=1}^{n} N_{выб\,i}^{ОК} + \sum_{j=1}^{m} N_{выб\,j}^{КВ}$, сумма нагрузки из дисциплин обязательной компоненты - $N_{выб\,i}^{ОК}$ и дисциплин из компоненты по выбору студентов – $N_{выб\,j}^{КВ}$.

$N_{асс}$ - учебная нагрузка, образованная на основании привлечения преподавателя в качестве ассистента для проведения лабораторных, практических и/или семинарских занятий, рассчитывается по формуле: $N_{асс} = \sum_{i=1}^{n} N_{асс\,i}^{ОК} + \sum_{j=1}^{m} N_{асс\,j}^{КВ}$, как сумма из двух компонент дисциплин.

$N_{ост}$ - остаток, образованный как часть нераспределенной учебной нагрузки или вторично перераспределенной: $N_{ост} = \sum_{i=1}^{n} N_{ост\,i}^{ОК} + \sum_{j=1}^{m} N_{ост\,j}^{КВ}$, тоже состоящий из двух видов дисциплин, i=1,2,…,n и j=1,2,…,m – количество учебных дисциплин из обязательной компоненты и компоненты по выбору соответственно, зависящих от специальности, курса обучения и года поступления.

Автоматизация распределения учебной нагрузки на кафедре приводит к необходимости создания сложной автоматизированной системы формирования учебной нагрузки для преподавателя, которая должна учитывать контингент студентов, действующие рабочие учебных планы по специальности, индивидуальные траектории обучения студентов и др.

Контингент студентов, обучающихся по кредитной технологии, изменяется намного чаще, чем в традиционной системе образования. Это связано с тем, что студенты имеют право мигрировать из вуза в вуз, поэтому количество студентов на каждом курсе заранее трудно предугадать. Кроме того, студенты в кредитной технологии обучения имеют возможность мигрировать из вуза в вуз в течение учебного периода, а не только после его окончания.

Возможность выбора учебных дисциплин для изучения влечет за собой создание учебных подгрупп в рамках существующих академических групп специальностей, которые формируются временно для изучения отдельных курсов, что значительно усложняет процесс составления учебного расписания занятий. В этой связи актуальными становятся вопросы объединения малокомплектных подгрупп для изучения отдельных курсов.

Случайный характер выбора себе преподавателя и учебных курсов студентами, приводит к неравномерной нагрузке преподавателей на кафедре, некоторые из которых могут быть сильно перегружены или не догружены учебной нагрузкой. Для сглаживания таких моментов нужно использовать более утонченные методы перераспределения остатка или ассистирующего компонентов нагрузки. Актуальными становятся модели оптимизации учебной нагрузки при перераспределении между преподавателями кафедры.

Как видно процесс определения и распределения учебной нагрузки на кафедре для отдельного преподавателя в кредитной образовательной технологии становится более сложным и зависимым от многих факторов, делается более зависимым от использования автоматизированных систем.

Создание автоматизированной системы распределения учебной нагрузки на кафедре позволит:

- автоматизировать сам трудоемкий процесс расчета и распределения нагрузки;
- подготовить некоторое приемлимое количество вариантов распределения учебной нагрузки для последующего более оптимального составления расписания учебных занятий;
- оптимизировать нагрузку отдельного преподавателя с учетом равномерности его загрузки по семестрам и личных индивидуальных предпочтений;
- использовать заложенные механизмы распределения как для кредитной технологии обучения студентов, так и для линейной системы.

Основными модулями такой системы являются:

- Модуль ведения рабочих учебных планов по отдельным специальностям, курсам и годам обучения, с полным перечнем учебных дисциплин из обязательной компоненты и компоненты по выбору студентов, с разбивкой часов, выделяемых на все виды занятий;
- Модуль ведения контингента студентов, восстанавливаемых для обучения или переводимых из других вузов;
- Модуль формирования и ведения индивидуальных образовательных траекторий студентов;
- Модуль для ведения штатов профессорско-преподавательского состава кафедр с указанием основных атрибутов, влияющих на распределение учебной нагрузки;
- Модуль формирования и ведения силлабусов по каждой учебной дисциплине в отдельности.

Здесь же появляются вопросы увязки и согласования работы этих модулей в общей автоматизированной системе.

Автоматизация процесса распределения учебной нагрузки на кафедре повысит эффективность управления учебным процессом в вузе, позволит решать острые вопросы по нагрузке для отдельного преподавателя более оперативно и комплексно.

Ротация и объединение потоков в задаче автоматизированного управления рабочим учебным планом

Макатов В. З.,
инженер-программист, Казахстанский филиал МГУ им. М.В.
Республика Казахстан, 010010, г. Астана, ул. Мунайтпасова, 7, каб. 717
тел.(факс):8(3172) 35-15-04, E-mail: mak-vz@yandex.ru
Фураева И. И.,
к.ф.-м.н., доцент кафедры вычислительной техники и информационных систем
Евразийского национального университета им. Л.Н. Гумилева
Республика Казахстан, 010010, г. Астана, ул. Мунайтпасова, 5, каб. 309
тел.: 8(3172) 27-41-43.

Рабочий учебный план (РУП) является основным документом процесса обучения в вузе. Он включает в себя набор дисциплин, которые должны быть изучены за определенный период времени. Все учебные дисциплины, входящие в учебный план для кредитной технологии обучения делятся на дисциплины обязательного компонента и дисциплины компоненты по выбору студентов. Обязательные дисциплины представляют собой набор учебных курсов для обязательного изучения, дисциплины из компоненты по выбору, выбираются студентами самостоятельно. По тем и другим дисциплинам ведется строгий учет образовательных кредитов, которые должен набрать студент в период обучения. Количество учебных дисциплин в обоих компонентах может меняться в зависимости от года обучения.

Возможность выбирать себе отдельные дисциплины порождает множество учебных групп, которые должны быть представлены в учебном расписании для проведения занятий. Так как выбор учебных дисциплин студентами носит случайный характер, появление отдельных групп студентов трудно предугадать. Появление множества групп приводит к острой нехватке аудиторного фонда для проведения занятий по выбранным студентами дисциплинам. Попытки управлять аудиторным фондом в линейной образовательной системе уже предпринимались.

Так в работе [2] определены следующие рычаги в системе управления аудиторным фондом:

- разработка систем рекомендаций по составлению расписания;
- введение коэффициента сменности;
- оптимальное формирование учебных потоков;
- корректировка учебных планов;
- формирование количественного и качественного состава студентов;
- качественные изменения имеющегося аудиторного фонда;
- изменение перечня специальностей.

Для осуществления управления аудиторным фондом необходимо анализировать и оптимизировать входные данные, используемые в процессе составления расписания.

Составление расписания учебных занятий напрямую связано с рабочими учебными планами, распределением нагрузки на кафедрах и имеющимся аудиторным фондом. Одним из факторов

оптимизации использования аудиторного фонда является составление расписания учебных занятий для малокомплектных групп. Увеличение количества таких групп приводит не только к росту среднегодовой нагрузки преподавателей, но и к уменьшению общего контингента студентов, которых можно разместить для проведения аудиторных занятий, так как аудиторный фонд ограничен. Наибольшие трудности малокомплектные группы создают при составлении расписания для дисциплин, использующих аудитории не общего, а специального пользования: компьютерные классы, лингафонные кабинеты, кабинеты физики, химии и т.д.

Задача уменьшения количества малокомплектных групп или их влияния на использование аудиторного фонда является весьма актуальной для всех вузов. Одним из способов уменьшения такого негативного влияния является объединение потоков для проведения всех видов занятий: лекций, практических (семинарских), лабораторных, СРСП (самостоятельной работы студентов с преподавателем).

Традиционно РУП составляется только на основе ГОСО (государственного общеобязательного стандарта образования) без учета имеющегося контингента студентов, причем по каждой специальности и для каждого курса независимо от других. В настоящей работе рассматривается математическая модель формирования РУП, минимизирующая влияние малокомплектных групп на использование аудиторного фонда. Корректировка рабочих учебных планов производится только на текущий учебный год, так как на следующий учебный год ситуация может измениться в связи с изменением контингента студентов.

Обозначим $D^1_{i,j}$ - j-тую дисциплину для i-того потока, изучаемую в 1 семестре, $D^2_{i,j}$ – во втором семестре. Необходимо отметить, что дисциплины будут одинаковы для разных потоков только в том случае, если они не только имеют одно наименование, но и одинаковое количество часов (кредитов) и одинаковые изучаемые темы. Так, например, если дисциплина «информатика» для специальностей «информатика» и «информационные системы» имеет, согласно ГОСО, одинаковые темы для изучения, а для специальности «ПГС» другие темы при таком же общем количестве часов, поэтому для первых двух специальностей эта дисциплину рассматриваем как одну и ту же, а для специальности «ПГС» – как другую. Кроме того, следует учитывать не только общее количество аудиторных часов в неделю, но и их разбивку по типам занятий. Так дисциплина «информатика» с недельной разбивкой часов 2+2+2 и с 2+0+4 на разных потоках считаются различными дисциплинами.

Так как объединение потоков для проведения учебных занятий производится для студентов одного факультета и одного курса, то, не теряя общности, будем рассматривать множество дисциплин 1 и 2 семестров для каждого курса и языка обучения отдельно.

Для дальнейшего уменьшения количества отдельных аудиторных занятий в рассматриваемых группах имеется возможность переноса в РУП дисциплин из одного семестра в другой. Для этого выберем из множеств U^1_i, U^2_i дисциплины, не связанные пререквизитами, то есть для которых возможен перенос в другой семестр. Обозначим их T^1_i, T^2_i.

Задача формирования РУП ставится следующим образом: выбрать среди T^1_i, T^2_i такие дисциплины, что:

$$T_i^1 \cap (\bigcup_{l=1}^{k_1} SP_l^2) \neq \otimes, T_i^2 \cap (\bigcup_{l=1}^{k_1} SP_l^1) \neq \otimes$$

и перемещение дисциплин произвести таким образом, чтобы общее количество часов недельной нагрузки не превышало максимально допустимое.

Если количество часов в множествах заменяемых дисциплин одинаково, то есть $\sum_l K_l(P^1_{i,l}) = \sum_m K_m(P^2_{i,m})$ то такой обмен возможен и проблема объединения всех видов занятий для малокомплектных групп полностью решена. В общем случае $\sum_l K_l(P^1_{i,l}) \neq \sum_m K_m(P^2_{i,m})$.

Рассмотрим дисциплины $U^1_i - P^1_i$, $U^2_i - P^2_i$, которые нельзя объединить ни с какими дисциплинами 1 и 2 семестров полнокомплектных групп. Среди них выберем дисциплины не связанные междисциплинарными связями (пререквизитами), которые допускают перенос в другой семестр. Обозначим эти дисциплины Z^1_i, Z^2_i.

Если $\alpha > 0$, то среди дисциплин Z^2_i необходимо найти одну или несколько дисциплин, таких что $\sum_t K_t(Z^2_{i,t}) = \alpha$. Если $\alpha < 0$, то такие дисциплины необходимо найти среди Z^1_i.

Обозначим через R^1_i, R^2_i общее количество аудиторных часов в первом и втором семестрах на i-том потоке. По условию формирования РУП $R^1_i \le K\max$, $R^2_i \le K\max$, где $K\max$ - максимально

допустимое количество аудиторных часов в неделю (36 для линейной системы обучения). Перенос дисциплин из одного семестра в другой должен производиться таким образом, чтобы выполнялись неравенства:

$$R_i^1 - \alpha + \sum_t \delta_t * K_t(Z_{i,t}^2) \le K\max,\ \alpha > 0, \qquad R_i^2 - |\alpha| + \sum_t \delta_t * K_t(Z_{i,t}^1) \le K\max,\ \alpha < 0,$$

где $\delta = 1$, если дисциплина выбрана для переноса и $\delta = 0$ в противном случае.

Таким образом, задача свелась к поиску для каждого потока (для каждого i) таких дисциплин, которые следует поменять в РУП в семестрах.

Если среди дисциплин из множества Z_i^2 нет ни одной такой, чтобы выполнялось неравенство (2), то задача в такой постановке не имеет решения.

Так как возможен перенос дисциплин из обоих множеств Z_i^1, Z_i^2, то можно усложнить постановку задачи: определить δ_{il}^1, δ_{il}^2 таким образом, чтобы $\sum_l \delta_{il}^1 + \sum_y \delta_{iy}^2 \to \min$ при условии

$$\begin{cases} R_i^1 - \alpha + \sum_j (1 - 2\delta_{ij}^1) * K_j(Z_{i,j}^1) + \sum_j \delta_{ij}^2 * K_j(Z_{i,j}^2) \le K\max \\ R_i^2 + \alpha + \sum_j (1 - 2\delta_{ij}^2) * K_j(Z_{i,j}^2) + \sum_j \delta_{ij}^1 * K_j(Z_{i,j}^1) \le K\max \end{cases}$$

Вопросу объединения малокомплектных групп при формировании учебного расписания в кредитной технологии обучения еще мало уделено внимания, т.к. предполагается, что объединение потоков и групп должно производится административными методами при составлении учебного расписания учебными отделами вузов. Но переход на кредитную образовательную технологию порождает массу проблем из-за нехватки аудиторного фонда. Особенно эти проблемы остро себя проявляют там, где вузы к кредитной технологии обучения перешли не полностью, а лишь отдельными факультетами. Поиск решение проблемы видится в осуществлении комплексных мероприятий по оптимизации учебного процесса, где одним из элементов может быть рассматриваемая выше математическая модель ротации и объединения потоков.

Использование автоматизированных систем для решения вопросов управления и организации учебного процесса в вузе позволит снизить часть трудозатрат при формировании учебного расписания, позволит более эффективно управлять аудиторным фондом вуза.

Список литературы.

1. Гусаков В. П., Шпак А. В. Автоматизированная система поддержки принятия решений распределения аудиторного фонда. //XIV Международная конференция- выставка «Информационные системы в образовании» (ИТО-2004), Москва 2004 г.

2. Шевчук Е. В., Шпак А. В. Экспертная система определения качества использования аудиторного фонда вуза. //Материалы 3-й Алматинской Международной научно-практической конференции. Сборник статей, - Алматы: Кенже-Пресс-Медиа, 2002 г., 170с.

3. Шпак А. В. Математическая модель формирования учебных потоков с целью оптимизации использования лекционно-практического аудиторного фонда. //Научно технический журнал «Автоматика и информатика», № 1-2, 2002 г., 51с.

База данных и программный комплекс электронной энциклопедии по экологии

Малинина К. О. *(malinina@viniti.ru),*
Жаров А. В.,
Шамаев В. Г.,
Всероссийский институт
научной и технической информации (ВИНИТИ) РАН,
125315, Москва, ул. Усиевича, 20

Разработана и находится в стадии опытной эксплуатации электронная версия энциклопедии по экологии. Конечной целью является ее публикация в Internet. В процессе работы разработаны структура

базы данных на СУБД Microsoft SQL Server 2000, включая хранимые процедуры и пользовательские хранимые функции, и программа ведения БД энциклопедии, включая загрузку данных из исходных файлов, выгрузку данных в файлы различной структуры, обработку, редактирование и сортировку статей энциклопедии, заполнение поисковых таблиц.

Обеспечение ссылочной целостности данных и поддержки бизнес-правил в основном возложено на сервер БД, однако для верификации данных используются также служебные опции клиентской программы. Статьи энциклопедии загружаются из файлов Microsoft Word 2000, HTML-файлов, а также файлов оригинал-макета предыдущего издания энциклопедического словаря по экологии в издательской системе Ventura 2.0.

Для публикации экологической энциклопедии в сети Internet разработана витрина данных: база данных на СУБД MySQL и PHP-сценарии доступа к ней, что позволяет использовать в качестве Web-сервера Apache под ОС семейства Unix. Разработаны механизмы передачи данных из БД Microsoft SQL Server 2000 в БД MySQL. Передача текстов статей для публикации печатного издания энциклопедического словаря по экологии реализована как выгрузка в QuarkXPress.

Структура базы данных

Выбор Microsoft SQL Server 2000 в качестве СУБД обусловлен широкими возможностями обеспечения ссылочной целостности и поддержки бизнес-правил, наличием хранимых процедур и функций, поддержкой различных типов сравнения строк, а также удобством администрирования.

Энциклопедический словарь рассматривается как множество (набор) статей, упорядоченных по алфавиту. Текст статьи хранится в HTML-формате, что позволяет избежать перекодировки при просмотре статьи в Internet. Не предполагается наличия в энциклопедии двух статей с одинаковыми заголовками, что контролируется уникальным ключом на уровне сервера. В качестве первичного ключа используется идентификатор, формируемый по фиксированному шаблону, что дает возможность не нарушать ссылочную целостность при коррекции заголовка статьи.

Сортировка статей, добавление иллюстраций, разбиение энциклопедии на главы по буквам, формирование ссылок и алфавитного указателя, получение ключевых слов и т.д. решаются программным путем с использованием подчиненных таблиц и других объектов БД (представления, хранимых процедур и функций). Данные, по которым проводится обработка, а именно заголовок статьи (термин энциклопедического словаря), его перевод, синонимы, сокращение, отсылка к другой статье, вынесены в отдельные поля.

В БД электронной энциклопедии используется регистронезависимая русскоязычная сортировка строковых полей.

При разработке БД предполагалось, что размер основной таблицы не превосходит 20 тыс. записей, тогда как размер одной статьи может достигать десятков килобайт.

Таблица статей энциклопедии снабжена служебными полями: технологическим статусом, размером текста статьи и т.д.

Для обеспечения быстрого поиска статей по ключевым словам в массиве данных энциклопедии созданы и автоматизировано заполняются поисковые таблицы: список ключевых слов и инвертированный список статей по ключевым словам. Для рубрикации статей предусмотрена таблица-рубрикатор. Получение списка идентификаторов статей-ссылок по их заголовкам для произвольного числа терминов в поле «Ссылки» реализовано как табличная хранимая функция.

В качестве иллюстраций к статьям могут использоваться как рисунки, так и дополнительный текст (например, таблицы). Иллюстрации вынесены в отдельную таблицу БД, где в дополнение к изображению хранятся его название, формат, размеры и подпись к рисунку.

Связи между объектами БД, как между двумя статьями, так и между статьями и их иллюстрациями, фиксируются в таблице связей в виде отношения «многие–ко многим». Для обеспечения ссылочной целостности используется механизм внешних ключей.

Все таблицы БД снабжены полями, в которых фиксируются пользователь и дата последнего изменения записи, их поддержка обеспечивается на уровне сервера определением значений по умолчанию и триггерами. Поскольку количество объектов БД невелико, они созданы от имени владельца БД и имеют префикс владельца dbo.

Программа ведения базы данных энциклопедии

Программа «БД Энциклопедия» ведения базы данных электронной энциклопедии разработана в среде Borland Delphi 6.0 Enterprise. Связь клиентского приложения с базой данных осуществляется на основе ADO технологии через ODBC DSN. В DSN определяется имя SQL-сервера. Имя БД, login и пароль пользователя задаются при обращении к программе.

Электронная энциклопедия по экологии имеет два основных источника данных: файлы, подготовленные в Microsoft Word 2000 и сохраненные в HTML-файл, и файлы Word 6 для MS-DOS с разметкой текста издательской системы (ИС) Ventura 2.0. Реализован перевод разметки текста из ИС Ventura в HTML с сохранением результата преобразования в файл. Исходный файл содержит статьи

энциклопедии на заданную букву. Производится «разбор» файла с отслеживанием статуса состояния и постатейная загрузка в БД. Формируется протокол загрузки, для анализа корректности разбиения файла на статьи заполняется служебное поле «Число абзацев». В отдельные поля выделяются заголовок статьи, его перевод, синонимы термина, ссылки на другие статьи, сокращения и т.д. Присутствие в файле одноименных статьей считается ошибкой, второй вариант статьи отбрасываются. Поскольку данные из файлов Word для Windows считаются более свежими, чем те, что уже находятся в БД, существующая запись полностью замещается новой информацией. Файлы с разметкой ИС Ventura 2.0 являются более старыми, поэтому при их обработке дубли отбрасываются.

Реализованы просмотр статей в виде HTML-страниц через встроенный обозреватель, редактирование текста статьи, включая вставку HTML-разметки текста и специальных символов из заданного набора, поиск статьи по вхождению заданной подстроки, поиск вхождений подстроки в текст статьи, фильтрация по префиксу заголовка, установка «закладки» на статью и возврат на нее, вставка в текст статьи ссылки на другую статью, выгрузка текста статьи в файл и загрузка текста из HTML-файла, минуя процедуру разнесение по полям.

Рисунки из файлов форматов JPEG, BMP и GIF, подписи к ним, и дополнительный текст (таблицы) загружаются в БД и привязываются к статьям. Иллюстрация помещается в текст статьи либо как HTML-тег подключения изображения с подписью к нему, либо как ссылка на статическую (HTML-файл) или динамически формируемую (PHP-скриптом) HTML-страницу, содержащую название, тот же тег и подпись к рисунку.

Поисковые таблицы заполняются на основании поля «Ключевые слова» таблицы статей, которое в свою очередь формируется из термина (заголовка статьи), перевода, синонимов, ссылки, сокращения, полного варианта термина для сокращения.

Сортировка статей энциклопедии по алфавиту с присвоением числового кода с заданным шагом дает возможность сохранить порядок следования статей энциклопедического словаря при переносе БД на другую платформу, в том числе, не поддерживающую выбранного варианта сравнения строк (например, вне зависимости от установок СУБД MySQL), а также позволяет принудительно изменить его в том случае, когда лексиграфическая сортировка не дает требуемого результата.

Статический вариант Web-публикации энциклопедии состоит из набора HTML-страниц двухуровневого оглавления (алфавита и листов оглавления по буквам, содержащих по 45 терминов), HTML-файлов со статьями и файлов с иллюстрациями. Он может рассматриваться и как самостоятельный продукт, поскольку не требует ни наличия Web-сервера, ни подключения к Internet.

Для экспорта электронной энциклопедии в витрину данных используются коммуникативные форматы: CSV-файлы и пакетные файлы SQL-скриптов для загрузки в БД MySQL. Перечень и порядок выгружаемых полей настраиваются. Если выгружается часть статей, то в пакетный файл SQL-скриптов для MySQL, кроме описаний и текстов статей, добавляются данные из поисковых таблиц и описания иллюстраций для отобранных статей. Критерии отбора: дата последнего изменения записи, префикс заголовка статьи, диапазон порядковых номеров идентификаторов.

Для передачи текстов статей в ИС QuarkXPress формируется текстовый файл с тегами XPress. Для заголовка, перевода, синонимов и т.д. используются шрифтовые выделения. Выгрузка осуществляется с разбиением на главы по буквам. Порядок статей определяется кодом сортировки.

Ряд дополнительных операций предназначен для ускорения работы PHP-скриптов, формирующих HTML-страницы по запросам пользователей в Internet. Реализованы процедуры пакетной обработки статей: замена разделителей и приведение к верхнему регистру поисковых полей, проверка корректности и коррекция идентификаторов, установление и сброс перекрестных ссылок между статьями и пр. Осуществляется сбор статистических данных по наполнению БД: подсчет количества статей по каждой букве, поиск ссылок на несуществующие статьи с подсчетом общего числа таких случаев.

Публикация электронной энциклопедии в сети Internet

Web-публикация энциклопедии по экологии с возможностью поиска статей по ключевым словам реализована на основе базы данных на СУБД MySQL с использованием PHP-сценариев динамического формирования оглавления, поиска статей по ключевым словам и формирования текста статьи по данным из БД.

Состав и структура таблиц баз данных на СУБД Microsoft SQL Server 2000 и на СУБД MySQL аналогичны, но имеется ряд отличий, обусловленных разным назначением БД: первая (хранилище данных) используется для загрузки, внесения изменений и долговременного хранения данных, а вторая (витрина данных) – для поиска статей и публикации в Internet. Разделение хранилища и витрины – залог сохранности данных. БД состоит из таблицы статей, поисковых таблиц и таблицы описаний иллюстраций. Загрузка данных в БД MySQL производится запуском SQL-скриптов, сформированных программой ведения БД энциклопедии.

Кроме базы данных MySQL Web-публикация энциклопедии содержит статические HTML-страницы (главная страница, страница с формой поиска терминов и т.д.), статические HTML-страницы, формируемые программой ведения БД энциклопедии, PHP-скрипты формирования динамических

HTML-страниц, CSS-файлы (каскадные таблицы стилей) для настройки вида HTML-страниц и файлы с изображениями, иллюстрирующими статьи энциклопедии.

Список литературы

1. Экологический энциклопедический словарь. – М.: Издательский дом "Ноосфера", 1998. – 930 с.

2. Интернет-энциклопедия по экологии. 1. Представление в Интернете. Жаров А. В., Лосев К. С., Малинина К. О., Шамаев В.Г. ВИНИТИ. – М., 2006. 17 с.: Библиогр. 2 назв., ил. 9. – Рус. – Деп. в ВИНИТИ 20.04.2006, № 518-В2006

3. Интернет-энциклопедия по экологии. 2. Программно-аппаратный комплекс подготовки экологической энциклопедии к публикации в Интернете. Жаров А. В., Малинина К. О., Шамаев В. Г. ВИНИТИ. – М., 2006. 78 с.: Библиогр. 1 назв., ил. 33. –Рус. –Деп. в ВИНИТИ 20.04.2006, № 519-В2006.

4. Интернет-энциклопедия по экологии. 3. Выгрузка статей энциклопедии для загрузки в издательскую систему QuarkXPress. Жаров А.В., Малинина К. О., Шамаев В. Г. ВИНИТИ. – М., 2006. 7 с.: Библиогр. 2 назв., ил. 3. – Рус. – Деп. в ВИНИТИ 20.04.2006, № 520-В2006.

5. Интернет-энциклопедия по экологии. 4. Структура баз данных. Жаров А. В., Малинина К. О., Шамаев В. Г. ВИНИТИ. – М., 2006. 22 с.: Библиогр. 3 назв. – Рус. – Деп. в ВИНИТИ 20.04.2006, № 521-В2006.

Архитектура системы VITA-II для прототипирования учебных курсов on-line на основе онтологий

Малиновская Ольга, e-*mail: malinol@mail.ru*
СПбГПУ, КИТвП
Санкт-Петербург, Политехническая ул., д. 29

С помощью новых компьютерных технологий современный преподаватель может легко создавать дистанционные системы обучения, доступные через сеть Internet.

Однако существует ряд проблем, касающихся создания электронных обучающих курсов. Первая проблема состоит в том, что не все преподаватели имею одинаково хорошие знания в области компьютерных технологий, необходимые для создания и управления электронными обучающими курсами.

Вторая проблема, затрагивающая создание электронных обучающих курсов – это то, что не все пользователи обучающих курсов обладают одинаковым уровнем знаний, а также психологическими, социальными и другими характеристиками (имеют различные модели пользователя), поэтому и материал, предоставляемый разным пользователям должен отличаться. Таким образом, для обеспечения наилучшего процесса обучения необходимо, чтобы обучающая система была в определенной степени адаптивной, то есть подстраивала процесс обучения под конкретную модель пользователя.

В связи с вышеизложенными факторами была сформирована следующая задача – необходимо разработать универсальный программный инструментарий разработчика обучающих систем, имеющий простой интерфейс и позволяющий преподавателю, обладающему минимальным набором знаний в области информационных технологий, создать адаптивный обучающий курс на основе модели пользователя, соответствующий мировым стандартам в области систем дистанционного обучения (данная работа ведется в рамках российско-белорусского проекта ТАИС 06-01-81005). Проектируемый программный инструментарий был назван VITA II.

Система VITA II наследует основные преимущества инструментария «VITA» [1], в частности наличие представления данных в двух видах – в форме группы параграфов (далее – модульное обучение) и онтологии (далее – сетевое обучение).

Одним из основных недостатков системы «VITA» является то, что онтологии, предоставляющие учебные ресурсы конечным пользователям, являются чрезмерно громоздкими и сложными для восприятия, поэтому в систему «VITA II» вводится поддержка многоуровневых онтологий.

Для поддержки многоуровневых вложенных онтологий предлагается следующая схема разделения онтологий на три уровня:

a. Фрагмент контента (Content fragment (CF) [2] – онтологии нижнего уровня (в листьях содержат конечные концепты, которые не могут быть раскрыты в онтологию – например, текст, видео, аудио, изображение, таблица);

b. Объект контента (Content Object (CO) [2] – онтологии среднего уровня (содержат набор CF, CO и навигацию);

c. Обучающий объект (Lerning object (LO) [2] – онтологии верхнего уровня (коллекции CO и связи между ними).

При этом онтологии верхнего уровня (Learning object) должны быть описаны по стандартам IEEE 1484.12.1 [3] – 2002 и ADL SCORM Version 1.3 [4], то есть должны быть легко расширяемы, доступны другими обучающими системами и независимы от программной платформы и контекста.

Проектируемый программный инструментарий предполагает возможность работы пользователя с разными типами информации, например текст, html страницы, изображения, таблицы, диаграммы, онтологии, мультимедиа и другие. При этом каждому типу ресурсов должен соответствовать свой программный модуль, осуществляющий работу с данным ресурсом.

Полный список требований к проектируемой системе VITA II описан в [5].

Для воплощения в проектируемом инструментарии заданных требований, была разработана следующая архитектура – в состав системы входят четыре основных модуля:

- модуль управления информационными ресурсами (Resource Manager);
- модуль управления обучающими объектами (LOM Manager);
- модуль управления онтологиями (Ontology Manager);
- модуль адаптации (Adaptation Manager) – привязка групп и онтологий к стереотипной модели пользователя.

Полная архитектура системы представлена на Рис. 1, краткое описание каждого модуля находится ниже.

Модуль управления информационными ресурсами (Resource Manager)

Модуль управления информационными ресурсами представляет из себя древовидную структуру, где элементами верхнего уровня детализации являются разделы типов ресурсов, например изображения, текст, гипертекст, аудио, видео, а элементами нижнего уровня – сами обучающие ресурсы.

В модуль управления информационных ресурсов происходит загрузка исходных данных обучающего курса, таких как текст, html-страница, изображения различных форматов, презентации, аудио, видео и другие ресурсы.

Помимо функции организации информационных ресурсов менеджер информационных ресурсов позволяет формировать группы («скелет» поля знаний обучающего курса). Формирование базиса курса осуществляется при помощи добавления узлов (параграфов) к структуре курса и определения их параметров. Добавляемый элемент является потомком узла, к которому он добавляется. Каждый узел впоследствии может быть раскрыт как глава обучающего курса (html-текст), несущая определённую справочную информацию.

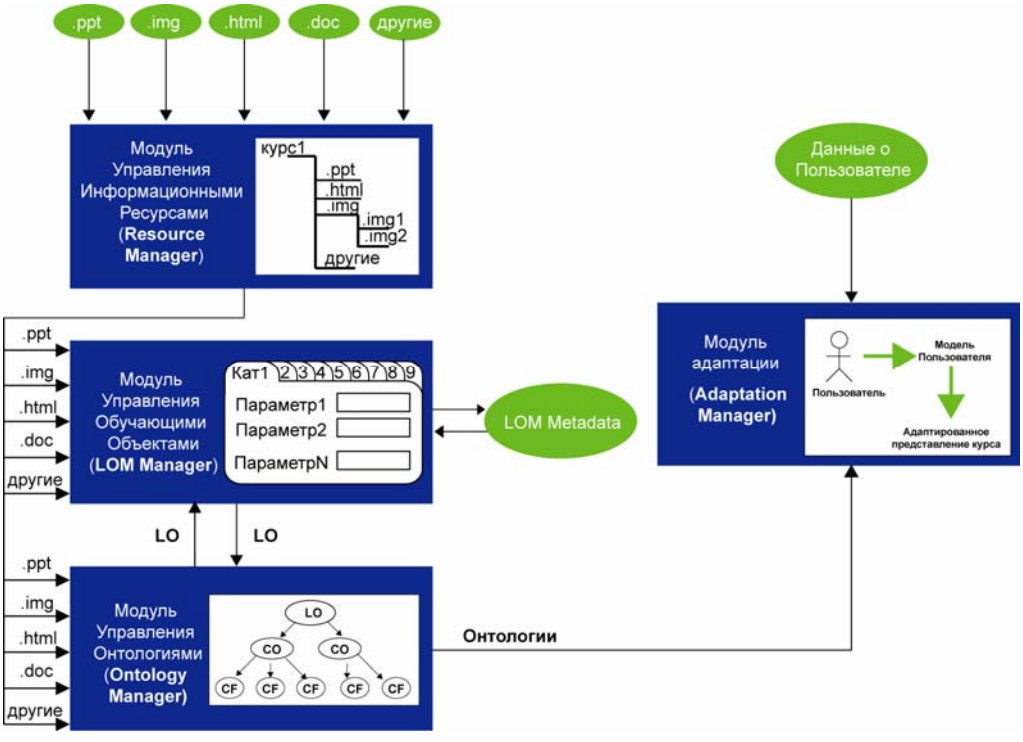


Рис. 1. Архитектура системы VITA II

Разработка групп позволяет реализовать модульную технологию обучения путем связывания определенной группы параграфов из сформированного ранее базиса, предоставляемых обучаемому, с конкретной моделью пользователя. Для определения этого набора необходимо использовать такие технологии, как сокрытие и аннотирование ссылок (параграфов) [6].

Разработчик курса определяет требования для фрагмента страницы и для страницы в целом. Требования являются булевскими выражениями над значениями концептов. Когда требования для фрагмента страницы принимают значение истина, тогда фрагмент включается для отображения в курсе. В ином случае фрагмент исключается. Также разработчик определяет для каждой группы значения набора оформительских параметров. Оформительские интерфейсные параметры характеризуют отображение информации на экране и степень участия в этом пользователя [7].

Все эти параметры должны быть однозначно определяемы по модели пользователя и иметь возможность динамически изменяться. Таким образом, модель пользователя должна содержать не только список параграфов и частоту их посещения, а и список оформительских параметров с конкретными значениями и уникальный идентификатор группы, соответствующей данной модели пользователя.

Модуль управления онтологиями (Ontology Manager)

Модуль управления онтологиями предоставляет возможности создания и визуализации многоуровневых онтологий, содержащих три уровня вложенности, структура которых была описана выше.

Для поддержки сетевой технологии обучения необходимо к каждой модели пользователя привязать онтологию, по которой будет осуществляться навигация по обучающему курсу.

Модуль Адаптации (Adaptation Manager)

Для того, чтобы разрабатываемая система была адаптивной необходимо, чтобы она обеспечивала формирование и коррекцию модели пользователя.

Модель пользователя будет представлять собой вектор значений, характеризующих данного пользователя. Эти значения должны динамически меняться в зависимости от поведения обучаемого.

Наиболее общая интерпретация этих значений заключается в том, что значение определяет уровень знаний пользователя в определённой области курса.

Каждый раз, когда пользователь посещает страницу, значение коэффициента, соответствующего номеру страницы увеличивается и соответственно увеличивается объём знаний пользователя в изучаемой области.

Модель пользователя должна ссылаться на уникальную группу и уникальную онтологию.

Модуль управления обучающими объектами (LOM Manager)

При моделировании модуля управления обучающими объектами LOM Manager использованы международные стандарты в области электронного обучения, введенные ADL Initiative, LOM [3], SCORM [4].

Это сделано для того, чтобы обучающие объекты, представляющие собой небольшие контекстно-независимые элементы курса, могли быть легко заимствованы и загружены в другие обучающие системы. Таким образом, для того чтобы сделать любой объект различной степени иерархии доступным и понятным многим системам, к нему необходимо присоединить его метаописание. Метаописание задается в соответствии с международными стандартами LOM [3] и SCORM [4]. Стандарт LOM задает элементы данных, описывающие обучающий объект, и разбивает их на девять категорий.

Модуль управления обучающими объектами LOM Manager используется для описания различных информационных ресурсов в соответствии с международными стандартами LOM и SCORM и представляет собой диалоговое окно с девятью закладками. Каждая закладка содержит информацию об одной из девяти категорий, описанных в стандарте LOM, в виде списка пар «Название поля формы»::Значение. Для формирования полного метаописания обучающего объекта в соответствии со стандартом LOM необходимо заполнить все поля, представленные на девяти закладках формы достоверной информацией, характеризующий описываемый обучаемый объект.

Основными функциями модуля является генерация метаописания информационных ресурсов и распознавания типа информационного ресурса в соответствии с его метаописанием.

Генерация метаописания информационных ресурсов происходит с помощью заполнения полей на форме и по нажатию клавиши происходит генерация XML файла заданной структуры. Распознавание типа информационного ресурса происходит посредством чтения привязанного к ресурсу XML файла.

При функционировании в режиме генерации метаописания обучающего объекта, на вход модуля могут поступать ресурсы совершенно различной структуры, такие как презентации, изображения, гипертекстовые документы, текстовые документы, онтологии, другие ресурсы.

Таким образом, с помощью стандарта LOM может быть унифицировано содержание метаописания обучающего объекта. Для стандартизации самой структуры и вида представления этой информации используется стандарт SCORM, хранящий данные метаописания объекта, заданные в соответствии со стандартом LOM, в виде XML файла.

Выводы

В результате работы со спроектированной системой разработчик дистанционных обучающих курсов получает адаптивный обучающий курс, представленный в виде суммы онтологий (сетевое обучение) или в виде группы параграфов (модульное обучение).

Сетевое обучение происходит на основе онтологий, на самом нижнем уровне которых находятся информационные ресурсы, имеющие различные типы (текст, изображения, мультимедиа, тесты и др.). Обучающие материалы соответствуют международным стандартам и могут быть заимствованы из уже сформированных сторонними разработчиками обучающих курсов, а также могут быть использованы другими разработчиками для генерации своих обучающих курсов.

Адаптация под конечного пользователя происходит через привязку конкретных онтологии и группы к модели пользователя, идентификаторы которых задается в модели пользователя.

Список литературы.

1. Гаврилова Т. А, Гелеверя Т. Е. «Программный инструментарий Vita. Версия 2.1. Техническая документация», 2002.

2. Jovanovic' J., Gasevic' D., "Ontology of Learning Object Content Structure", 2005.

3. Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, 15 July 2002.

4. ADL SCORM Version 1.3 WORKING DRAFT 0.9, November 27, 2002.

5. Гелеверя Т., Малиновская О., Гаврилова Т., Курочкин М., «Система VITA-II для прототипирования учебных курсов on-line на основе онтологий», сборник конференции «MEL-2006».

6. Брусиловский П. Л., 1996 "Технологии и методы адаптивной гипермедиа", User Modeling and User Adapted Interaction, v 6, n 2-3, стр. 87-129, http://ifets.ieee.org/russian/depository/Brusil_1996.zip

7. Васильева Е. "Проблемы интерфейсной адаптации в обучающих системах".

Применение нечеткой логики в системе оценки кадрового потенциала

Мамедова М. Г., *д.т.н. зав.отдел.,*
Джабраилова З. Г., *к.т.н., зав.сектор.*
Институт Информационных Технологий Нац.АН Азербайджана
370141, Баку, ул. Ф.Агаева, 9,
e-mail: depart15@iit.ab.az, тел.:(99 412) 399739.

Для глубокого и систематического изучения кадрового потенциала широко используются комплексные системы оценки персонала с применением различных методов. При этом содержание и методы оценки могут быть различными в зависимости от того, какие стратегические цели преследует руководство организации и какие конкретные задачи решаются для достижения поставленных целей.

В настоящей работе поставлена задача разработки системы оценки деятельности кадрового потенциала научно-исследовательских учреждений на основе подхода системного анализа, базирующейся методов теории нечеткой логики.

В качестве конкретного объекта исследования рассмотрены научно-исследовательские учреждения (научно-исследовательские институты и центры, опытные и специальные конструкторские бюро и др.). Повышение эффективности управления развитием НИУ невозможно без надежных критериев и методов оценки научной деятельности сотрудников. В данном случае непосредственным объектом оценки является деятельность человека, поэтому дать однозначную количественную оценку последней довольно проблематично. Более того, деятельность человека характеризуется многими факторами, критериями и показателями. Таким образом, задача оценки человеческого потенциала, с одной стороны, является многофакторной и многокритериальной, с другой стороны, поскольку в процессе оценки учитывается мнение различных экспертов, эти факторы, критерии и показатели в большинстве носят качественный характер и к тому же неравнозначимы. Еще одной особенностью является иерархичность оценок критериев, характеризующих трудовую деятельность человека, выраженная в том, что каждый отдельный критерий верхнего уровня основывается на агрегировании частных критериев ближайшего нижнего уровня.

Ключевыми вопросами при решении задачи оценки деятельности кадрового потенциала НИУ являются: 1) формирование структурной схемы системы оценки, т.е. структуры и набора применяемых ограничений, критериев, показателей оценки; 2) выбор методов получения и обработки первичной оценочной информации (подбор экспертов, экспертная оценка, разработка количественных и качественных градаций критериев); 3) разработка метода принятия решения, позволяющего от набора частных оценок по критериям переходить к интегральной оценке результатов.

Учитывая необходимость оперирования с информацией как качественного, так и количественного характера, в работе для формализации критериев и оценки деятельности научного потенциала НИУ использованы элементы теории нечетких множеств.

Предлагаемая методика оценки деятельности сотрудников НИУ базируется на методах принятия решений на основе нечеткой реляционной модели представления знаний.

Согласно нечеткой реляционной модели представления знаний, если $X=\{x_1, x_2, \dots, x_n\}=\{x_i, i=\overline{1,n}\}$ множество альтернатив, которые подлежат оценке и ранжированию, а $K=\{k_1, k_2,..., k_m\}=\{k_j, j=\overline{1,m}\}$ множество критериев, характеризирующих эти альтернативы, то степень удовлетворения альтернативы x_i критерию κ_j представляется функцией принадлежности $\varphi_{k_j}(x_i)\to[0,1]$, т.е.: $\varphi_{k_j}(x_i): X\times K\to[0,1]$.

Дадим формальную **постановку задачи**. Пусть $X=\{x_1, x_2, \dots, x_n\}=\{x_i, i=\overline{1,n}\}$ - множество сотрудников, работающих в каком-либо НИУ, деятельность которых характеризируется множеством неравнозначимых критериев: $K=\{K_1, K_2,..., K_m\}=\{K_j, j=\overline{1,m}\}$.

Каждый критерий K_j, входящий во множество критериев K, в свою очередь, характеризуется подмножеством частных критериев, т.е. $K_j=\{k_{j1}, k_{j2},..., k_{jT}\}=\{k_{jt}, t=\overline{1,T}\}$, причем элементы этих подмножеств также неравнозначимы.

Целью задачи является получение по результатам проведенной оценки деятельности кадрового потенциала систематизированного списка индивидуальных заключений на сотрудников, ранжированного от лучшего к худшему: $X: K\to X^*$, где X^*- систематизированный список сотрудников.

Методика оценки деятельности сотрудников сводится к выполнению следующих шагов:

1. С помощью агрегирования частных критериев нижележащего уровня оценивается каждый критерий верхнего уровня [2], т.е. если $\{\varphi_{k_{j1}}(x_i), \varphi_{k_{j2}}(x_i), \dots, \varphi_{k_{jT}}(x_i)\}=\{\varphi_{k_{jt}}(x_i), t=\overline{1,T}\}$ – функции принадлежности альтернативы x_i частным критериям $k_{j1}, k_{j2}, \dots, k_{jT}$ и $w_{j1}, w_{j2}, \dots, w_{jT}$ коэффициенты относительной важности этих частных критериев, то построением свертки этих неравнозначимых частных критериев определяется функция принадлежности альтернативы x_i обобщенному критерию K_j: $\varphi_{K_j}(x_i)=\sum_{t=1}^{T} w_{jt}\ \varphi_{k_{jt}}(x_i)$.

2. Определяется функция принадлежности альтернативы x_i обобщенному критерию K: $\varphi_K(x_i)=\sum_{j=1}^{m} w_j \varphi_{K_j}(x_i)$, где w_j, $j=\overline{1,m}$ соответствующий коэффициент относительной важности критерия K_j, $j=\overline{1,m}$.

3. Выбирается альтернатива, имеющая максимальную степень принадлежности обобщенному критерию K: $\varphi(x^*)=\max\{\varphi_K(x_i), i=\overline{1,n}\}$.

n — число альтернатив. Выбранная альтернатива является наилучшей альтернативой.

Для определения коэффициентов относительной важности критериев и их элементов может быть использованы: а) метод попарного сравнения одновременно рассматриваемых критериев; б) метод определения коэффициентов важности одновременно рассматриваемых критериев на основе 10-бальной системы экспертной оценки.

Для определения степени удовлетворения альтернатив частным критериям, т.е. функций принадлежности альтернативы x_i частным критериям $k_{j1}, k_{j2}, \dots, k_{jT}$, где $K_j=\{k_{j1}, k_{j2},..., k_{jT}\}=\{k_{jt}, t=\overline{1,T}\}$, каждому элементу подмножества критериев K_j присваиваются качественные градации, соответствующие принятым лингвистическим оценкам («хорошо», «нормально», «плохо» и т.д.), и определяются их нечеткие соответствия.

Таблица 1 на примере частного критерия «участие в научно-исследовательских работах», входящего в подмножество критерия «научно-теоретическая деятельность», иллюстрирует описанную процедуру.

Таблица 1. Задание качественных градаций и их нечетких соответствий для частного критерия «участие в научно-исследовательских работах», являющегося элементом подмножества критерия «научно-теоретическая деятельность»

Качественные градации частного критерия «участие в научно-исследовательских работах (НИР)», являющегося элементом подмножества критерия «научно-теоретическая деятельность»	Лингвистическая оценка	Область нечетких соответствий
1) Очень активно участвует в выполнении НИР;	Хорошо	[0,8-1]
2) Участвует в выполнении НИР;	Нормально	[0,56-0,79]
3) Частично участвует в выполнении НИР ;	Слабо	[0,20-0,55]

Далее членами экспертной группы для каждой градации определяются значения индивидуальных нечетких оценок из соответствующей области определения, входящей в интервал [0,1]. Объединение индивидуальных оценок отдельных экспертов в единую, коллективную оценку определяет окончательную нечеткую оценку градаций. Следует отметить, что в данном случае объединение индивидуальных экспертных оценок в коллективную производится не по правилу пересечения нечетких множеств или по правилу их объединения, а представляет собой некоторую новую операцию над такими множествами, а именно их согласованный выбор. Согласно данному подходу в качестве коллективной выбирается та индивидуальная оценка, которую дает некоторый специально сконструированный «наиболее предпочтительный» эксперт. Очевидно, что такой эксперт должен в каждой точке области всех возможных альтернатив выбирать в качестве меры принадлежности этой точки коллективной оценки ту из высказанных экспертами мер ее принадлежности индивидуальным оценкам, которая в общем случае удалена от крайних оценок, имеющихся в данном коллективе, и занимает некоторое «среднее» положение.

На основе предложенной методики разработана система управления кадровым потенциалом НИУ, реализованная в Институте информационных технологий Национальной академии наук Азербайджана для оценки трудовой деятельности сотрудников.

Разработанная система реализована в Visual FoxPro 6.0.

Предложенная методика, наряду с использованием для оценки деятельности кадрового потенциала научных учреждений, может быть успешно применена также для повышения эффективности управления персонала в государственных и коммерческих организациях, предприятиях и учреждениях.

Матричное представление нечеткой логики

Марценюк М. А., *заведующий кафедрой,*
профессор кафедры компьютерных систем и телекоммуникаций,
Пермский государственный университет.
Россия, 614990, г.Пермь, ул. Букирева, 15.
E-mail: mrcn@psu.ru

Построено матричное представление нечеткой логики, являющееся естественным обобщением «четкой» логики и тензорной логики, введенной [Мизраджи]. Логические переменные представляются 2-мерными векторами, логические операции между парами переменных – как векторные «произведения». Показано, что один из сомножителей произведения можно представить в виде некоторой 2×2 матрицы, а сама операция сводится к умножению матрицы на вектор, что дает в результате также вектор. В том случае, когда матрица не вырождена, ее можно обратить, что дает возможность обратить логическую операцию (найти решение логического уравнения). Таким образом, логические выводы сводятся к решению некоторых матричных уравнений. В четкой логике компоненты x_0, x_1 2-вектора $\boldsymbol{x}$ логической переменной могут принимать значения, равные лишь 0 или 1. В нечеткой логике x_0, x_1 могут быть любыми числами, лежащими в интервале $[0,1]$, причем их сумма удовлетворяет условию $x_0 + x_1 = 1$. Показано, что нечеткий логический вывод также может быть сведен к решению матричных уравнений. Показано также, что нечеткие операции могут быть реализованы в виде электрических цепей, причем сама нечеткая переменная реализуется как делитель тока. В каждом узле цепи выполняется закон сохранения заряда (первый закон Кирхгофа). Интерпретируя компоненты нечеткого вектора как вероятности, можно приписать ему информационную меру нечеткости, которая вычисляется по формуле Больцмана-Шеннона.

Нечеткая логика нашла повсеместное использование в интеллектуальных системах с неполной информацией. Предложено большое число моделей нечеткой логики и нечеткого логического вывода (далее для краткости Н-логика и Н-вывод) [Заде, Рыжов, Круглов и др.]. Эти модели сводятся, во-первых, к обобщению на область нечетких переменных логических операций (конъюнкции, дизъюнкции, отрицания и др.), определенных над «четкими» логическими переменными. В этом смысле они считаются нечеткими логическими переменными (НЛП). Во-вторых, для построения моделей лингвистических переменных НЛП интерпретируются как степени принадлежности точек универсального множества некоторому нечеткому множеству и, соответственно, над ними вводятся операции объединения, пересечения, дополнения и другие операции, которые обобщают известные операции над «четкими» множествами. В этом смысле говорят о нечетких переменных принадлежности (НПП). Для построения алгоритмов Н-вывода необходимо развить алгебру операций над нечеткими переменными, понимаемыми и как НЛП и как НПП.

Нечеткие векторы. Целью данной работы является построение матричной модели Н-логики. Отправной точкой выбрано тензорное представление Н-логики (Т-логика), предложенное в работах [Мизраджи]. В Т-логике логические переменные x, y представляются 2D векторами: $\boldsymbol{x} = (x_0, x_1)^{\mathsf{T}}$, $\boldsymbol{y} = (y_0, y_1)^{\mathsf{T}}$ (здесь и далее надстрочный символ T обозначает матричное транспонирование). В «четкой» логике компоненты векторов могут принимать только два значения: 0 или 1. Считается, что вектор $\boldsymbol{x}$ принимает значение «истина (true)», если $\boldsymbol{x} = \boldsymbol{e}^{(1)} = (0, 1)^{\mathsf{T}}$ и «ложь (false)», если $\boldsymbol{x} = \boldsymbol{e}^{(0)} = (1, 0)^{\mathsf{T}}$. При переходе к Н-логике могут быть отличны от нуля обе компоненты вектора $\boldsymbol{x}$.

В соответствии с известными понятиями теории информации мы можем ввести неопределенность (энтропию) системы $S(\boldsymbol{x})$ в состоянии, задаваемом вектором $\boldsymbol{x}$, следующим образом:

$$S(\boldsymbol{x}) = -x_0 \log_2 x_0 - x_1 \log_2 x_1 .$$

Легко видеть, что в «четких» состояниях энтропия равна нулю, а в состоянии крайней степени неопределенности, когда $x_0 = x_1 = 1/2$, энтропия максимальна и равна единице (1 бит). Если в задаче используется N нечетких логических переменных, каждая из которых задается своим вектором $\boldsymbol{x}_i$, где $i = 1, 2, ..., N$, то нечеткость этого множества переменных может быть найдена суммированием нечеткостей по каждой из переменных $\boldsymbol{x}_i$: $S(X) = \sum_{i=1}^{N} S(\boldsymbol{x}_i)$, где символом X обозначено множество логических переменных $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$, которое можно мыслить как «векторное поле», заданное в точках некоторого множества $A = \{a_1, a_2, ..., a_N\}$.

Операции над нечеткими векторами. В отличие от Т-логики логическая операция над векторами $\boldsymbol{x}, \boldsymbol{y}$ интерпретируются как операции «векторного» умножения определенного типа P: $\boldsymbol{x} \circ \boldsymbol{y} = \boldsymbol{z}$. Каждому логическому 2D вектору $\boldsymbol{x}$ в произведении $\boldsymbol{x} \circ \boldsymbol{y} = \boldsymbol{z}$ можно сопоставить 2×2 матрицу (квадратную матрицу второго порядка) $X^{(P)}$, компоненты которой однозначно выражаются через компоненты вектора $\boldsymbol{x}$, а сам вид матрицы зависит от типа операции P. С помощью матрицы $x^{(P)}$ логическая операция может быть представлена в матричном виде как $x^{(P)} \cdot \boldsymbol{y} = \boldsymbol{z}$, где точка « $\cdot$ » символизирует обычное матричное умножение, а векторам $\boldsymbol{y}$ и $\boldsymbol{z}$ сопоставлены матричные столбцы. В развернутом виде это соотношение может быть записано следующим образом

$$\begin{pmatrix} x_{00}^{(P)} & x_{01}^{(P)} \\ x_{10}^{(P)} & x_{11}^{(P)} \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} z_0 \\ z_1 \end{pmatrix}.$$

Такая форма представления логических операций особенно удобна для численной реализации нечеткого логического вывода. Будем считать, что в равенстве $\boldsymbol{x} \circ \boldsymbol{y} = \boldsymbol{z}$ нечеткие переменные – вектора $\boldsymbol{x}, \boldsymbol{z}$ заданы, а вектор $\boldsymbol{y}$ неизвестен. Тогда, обращая матрицу $x^{(P)}$ (при условии, что она не вырождена!), находим $\boldsymbol{y} = \left(x^{(P)}\right)^{-1} \cdot \boldsymbol{z}$.

Компоненты матриц конъюнкции и дизъюнкции $x^{(C)}$ и $x^{(D)}$, а также и матрица импликации $x^{(I)}$ могут быть найдены в следующем виде

$$x^{(C)} = \begin{pmatrix} 1 & x_0 \\ 0 & x_1 \end{pmatrix}; \quad x^{(D)} = \begin{pmatrix} x_0 & 0 \\ x_1 & 1 \end{pmatrix}; \quad x^{(I)} = \begin{pmatrix} x_1 & 0 \\ x_0 & 1 \end{pmatrix}.$$

Для дальнейших рассуждений используется более удобное обозначение матриц конъюнкции и дизъюнкции $C(\boldsymbol{x}) = x^{(C)};\ D(\boldsymbol{x}) = x^{(D)};\ I(\boldsymbol{x}) = x^{(I)}$. Матрица импликации может быть выражена через матрицу дизъюнкции следующим образом: $I(\boldsymbol{x}) = D(\boldsymbol{x}^{\neg})$, где $\boldsymbol{x}^{\neg}$ – вектор, полученный отрицанием вектора $\boldsymbol{x}$. Операция отрицания сводится к перестановке компонент вектора: если $\boldsymbol{x} = (x_0 \quad x_1)^T$, то $\boldsymbol{x}^{\neg} = (x_1 \quad x_0)^T$.

Логические операции над «четкими» переменными обладают рядом алгебраических свойств, часть из которых не выполняется для нечетких переменных. Анализ операций можно найти в литературе (см., например, [Рыжов]). При переходе к матричному представлению анализ упрощается за счет того, что свойства логических операций (например, ассоциативность конъюнкции и дизъюнкции) следуют из свойств умножения матриц. В нечеткой логике не выполняются законы дистрибутивности $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ и $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ имевшие место для четких переменных.

Матричная реализация нечеткого логического вывода. В четкой логике используется силлогизм, называемый правилом вывода modus ponens. Если x истинно (true), $x \to y$ – тоже истинно (true), то из этого следует, что и y истинно. Это правило можно рассматривать как решение «системы уравнений» $x \to y = t$, $x = t$. Обобщение метода решения логических уравнений на область нечетких переменных достигается переходом к матричному представлению логических операций. Рассмотрим нечеткое правило логического вывода модус поненс. Задача состоит в том, чтобы найти нечеткость переменной y, решая уравнение $x \to y = z$ при заданных нечетких переменных x, z. Уравнение $x \to y = z$ может быть записано в матричном виде как $D(\boldsymbol{x}^{\neg}) \cdot \boldsymbol{y} = \boldsymbol{z}$, где $D(\boldsymbol{x}^{\neg})$ – дизъюнктивная матрица, целиком определяемая вектором $\boldsymbol{x}^{\neg}$ (см. выше). Считая векторы $\boldsymbol{x}, \boldsymbol{z}$ известными, можно найти решение этой системы линейных уравнений. Определитель $\det\left(D(\boldsymbol{x}^{\neg})\right)$ матрицы $D(\boldsymbol{x}^{\neg})$ равен x_1 и отличен от нуля, если $x_1 \neq 0$. Решение уравнения $D(\boldsymbol{x}^{\neg}) \cdot \boldsymbol{y} = \boldsymbol{z}$ может быть найдено в виде $y_0 = \frac{z_0}{x_1}; \quad y_1 = 1 - \frac{z_0}{x_1}$. Решение имеет смысл, если $y_1 \geq 0$, откуда получаем дополнительное условие (кроме $x_1 \neq 0$) $x_1 \geq z_0$, накладываемое на переменную x_1. Таким образом, из матричной формы первого уравнения $x \to y = z$ получены явные формулы для вычисления компонент неизвестного нечеткого вектора $\boldsymbol{y}$. Второе уравнение «четкой» логики $x = t$ появляется в нечеткой логике как условие неравенства нулю определителя системы уравнений – $x_1 \neq 0$. Отметим, что при условии $x_1 = 1$ и $z_0 = 0$ получаем известное решение системы уравнений модус поненс «четкой» логики: $y_0 = 0;\ y_1 = 1$.

Операции над лингвистическими переменными. Для применений нечеткой логики вводят так называемые лингвистические переменные (Л-переменные). Каждая Л-переменная X описывается совокупностью нечетких переменных $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$, которую можно мыслить как «векторное поле», заданное в точках некоторого множества $A = \{a_1, a_2, ..., a_N\}$. Компоненты вектора $\boldsymbol{x}_i$ $(i = 1, 2, ..., N)$ из множества X задают степень принадлежности точки a_i множества A данной Л-переменной. Степень нечеткости Л-переменной описывается энтропией. Часто приходится решать следующую задачу. Рассмотрим некоторую точку $a \in E$, которая с разной степенью может принадлежать нескольким нечетким множествам $A, B, C, ...$ Соответственно, одной и той же точке $a \in E$ приписываются нечеткости $\mu^{(A)}(a), \mu^{(B)}(a), \mu^{(C)}(a), ...$ Чтобы найти степени принадлежности $\mu^{(A \cup B)}(a)$, $\mu^{(A \cap B)}(a), ...$ точки $a \in E$ к множествам $A \cup B, A \cap B$, являющимся объединением и пересечением множеств используются операции Заде $\mu^{(A \cup B)}(a) = \max\left(\mu^{(A)}(a), \mu^{(B)}(a)\right)$, $\mu^{(A \cap B)}(a) = \min\left(\mu^{(A)}(a), \mu^{(B)}(a)\right)$.

Далее мы будем обозначать операции над нечеткими векторами, рассматриваемыми степени принадлежности знаками «$\dot{+}$» и «$\dot{\times}$»: $\boldsymbol{x}^{(A \cup B)}(a) = \boldsymbol{x}^{(A)}(a) \dot{+} \boldsymbol{x}^{(B)}(a)$; $\boldsymbol{x}^{(A \cap B)}(a) = \boldsymbol{x}^{(A)}(a) \dot{\times} \boldsymbol{x}^{(B)}(a)$.
Эти операции также могут быть представлены в матричной форме.

Приведем пример нечетких рассуждений. Рассмотрим лингвистические переменные L, M, заданные на двух универсальных множествах E, F. Каждая переменная может иметь несколько признаков и может описываться несколькими нечеткими подмножествами $L = (A_1, A_2,..); M = (B_1, B_2,...)$, заданными на E, F соответственно. Введем описание нечетких множеств с помощью наборов нечетких векторов $X_1 = (\boldsymbol{x}_1^{(A1)}, \boldsymbol{x}_2^{(A1)},..., \boldsymbol{x}_{N1}^{(A1)})$, $X_2 = (\boldsymbol{x}_1^{(A2)}, \boldsymbol{x}_2^{(A2)},..., \boldsymbol{x}_{N1}^{(A2)})$, … и $Y_1 = (\boldsymbol{y}_1^{(B1)}, \boldsymbol{y}_2^{(B1)},..., \boldsymbol{y}_{N2}^{(B1)})$, $Y_2 = (\boldsymbol{y}_1^{(B2)}, \boldsymbol{y}_2^{(B2)},..., \boldsymbol{y}_{N2}^{(B2)})$,… Для простоты множества E, F считаются дискретными и содержащими по $N1, N2$ элементов соответственно. Между Л-переменными L, M должны быть установлены зависимости в виде нечетких логических правил (формул) вида «если …, то…», например, «если $L = A_i$, то с определенной долей нечеткости $M = B_j$ » и т.п. Количественно эти правила могут быть представлены как нечеткие логические уравнения вида $\boldsymbol{x}_i^{(Ak)} \to \boldsymbol{y}_j^{(Bm)} = \boldsymbol{z}_{ij}^{(km)}$ или в матричном виде как $D(\boldsymbol{x}_i^{(Ak)\neg}) \cdot \boldsymbol{y}_j^{(Bm)} = \boldsymbol{z}_{ij}^{(km)}$. Если считать величины $\boldsymbol{x}_i^{(Ak)}$, $\boldsymbol{y}_j^{(Bm)}$ заданными, то матричное уравнение дает возможность найти величины $\boldsymbol{z}_{ij}^{(km)}$, которые определяют на множестве $E \otimes F$ некоторые нечеткие отношения $R^{(km)}$, являющиеся нечеткими подмножествами $R^{(km)} \subset E \otimes F$. Описанная система исходных посылок может использоваться следующим образом. Предположим, что по некоторым данным стало известно значение (номер) точки i множества E, описывающего первую лингвистическую переменную L. Спрашивается, какое значение j можно при этом приписать точке множества F, описывающего вторую Л-переменную M? Поскольку индекс i считается фиксированным, то можно считать, что переменные $\boldsymbol{z}_{ij}^{(km)}$ задают некоторые нечеткие подмножества $C_i^{(km)}$ на множестве M. Объединяя эти подмножества, мы получим подмножество $\Psi_i = \bigcup_{km} C_i^{(km)}$, заданное на F, центр «тяжести» которого можно принять за искомую координату j.

Список литературы.

1. Заде Л. А. Понятие лингвистической переменной и его применение к принятию приблизительных решений. – М.: Мир, 1976.

2. Рыжов А. П. Элементы теории нечетких множеств и ее приложений. – М. МГУ, 2003

3. Круглов В. В., Дли М. И. Интеллектуальные информационные системы. Компьютерная поддержка систем нечеткой логики и нечеткого вывода. – М. ИФМЛ, 2002.

4. Mizraji E. Modalities in Vector Logic. // Notre Dame Journal of Formal Logic. V. 35, N.2, 1994 Pp. 272-282.

Некоторые вопросы кластеризации ситуаций в алгоритмах реального времени

Мельников Б. Ф., e-mail: *B.Melnikov@tltsu.ru,*
профессор каф. Прикладная математика и прикладная информатика
Тольяттинского государственного университета,
445667 г. Тольятти ул. Белорусская, 14
Мельникова Е. А., *elenamel@rambler.ru,*
старший преподаватель каф. Телекоммуникационные технологии и сети
Ульяновского государственного университета,
432970 г. Ульяновск ул. Л.Толстого, 42

Данная работа является продолжением нескольких предыдущих работ авторов, посвящённых специальному мультиэвристическому подходу к задачам дискретной

оптимизации: рассматриваются те вопросы, которые связаны с проблемами кластеризации ситуаций (классические работы о применении алгоритмов кластеризации к задачам искусственного интеллекта, прежде всего – к ситуациям, - [1], о решении задач «по аналогии» – [2])

Описание алгоритма

Приведём математическое описание алгоритма из группы “Hierarchical Clustering Algorithms”. По мнению авторов, именно этот вариант алгоритма наиболее удачен для рассматриваемых нами задач кластеризации ситуаций. Итак, пусть R_{ij} – минимальное расстояние между элементами, входящими в кластеры с номерами i и j, а r_i – максимальное расстояние между элементами кластера i. Стандартные алгоритмы кластеризации, а также алгоритмы для оценки её качества, обычно используют значения

$$f(\min_{i,j}(R_{ij})) + g(\max_{i}(r_i)),$$

где f и g – специальные возрастающие функции (выбираемые в зависимости от конкретной задачи, конкретного алгоритма и др.), а i и j принимают все возможные значения.

В отличие от стандартных алгоритмов авторы используют ту же самую формулу – но с другим смыслом значений r_i. А именно, пусть:

- i – рассматриваемый кластер,
- $m_1,...,m_n$ – все его элементы,
- l_{mn} – расстояние между его элементами m и n,
- $(M_1,..., M_N)$ – некоторая конечная последовательность, состоящая из элементов кластера $m_1,...,m_n$ и включающая каждый из них по крайней мере 1 раз. Тогда мы считаем, что

$$r_i = \min_{(M_1,...,M_N)} \left(\max_{0<K<N} (l_{M_K M_{K+1}}) \right).$$

Заметим, что данное определение не является алгоритмом, однако простейшие из возможных алгоритмов вычисления значений r_i, построенные на его основе, несложны.

Итак, мы привели формальное описание одного специального алгоритма кластеризации. В качестве одной из возможных областей его применения авторы рассматривают задачу формирования первой страницы результатов поиска в Интернете [5]. Другая возможная область применения этого алгоритма – задачи дискретной оптимизации [6].

Практическое программирование, решение нескольких конкретных задач показало, что этот алгоритм лучше других отражает возможности расположения ситуаций. Причём здесь мы имеем в виду ситуации, как для некоторой задачи дискретной оптимизации, так и для игровых программ. Мы стараемся рассматривать ситуации не только в пределах одного кластера, но и в некотором порядке – этот порядок как раз и соответствует возможной последовательности $(M_1,…M_N)$ в описанных выше формулах, и, что ещё более важно, выделению в этой последовательности ситуаций, встречающихся несколько раз (назовём их ключевыми ситуациями). Расположение ситуаций в указанном порядке производится для последовательного рассмотрения этих ситуаций, решения подзадач, им соответствующих, в похожем порядке – причём мы начинаем решение подзадач именно с рассмотрения ключевых ситуаций. При этом, т.е. при расположении ситуаций в указанном порядке, мы удачно рассматриваем решение подзадач «по аналогии» – поскольку ситуации, близкие по «хорошей» метрике, с большой вероятностью имеют похожие решения. Здесь мы в первую очередь имеем в виду не просто возможность сделать одинаковый шаг, а эвристический выбор, получение одинакового разделяющего элемента для такого шага в двух близких по метрике подзадачах.

Метрики для ситуаций одной задачи дискретной оптимизации

Для кластеризации по этому алгоритму нужно иметь метрику на множестве ситуаций – множестве подзадач. Опишем метрику для задачи вершинной минимизации недетерминированных конечных автоматов Рабина-Скотта (в принципе подобные эвристики могут применяться для самых разных задач дискретной оптимизации).

Рассмотрим постановку задачи (подробнее в [3]). Некоторая прямоугольная матрица заполнена числами 1 и 0. Блоком назовём некоторую пару подмножеств строк и столбцов этой матрицы, удовлетворяющую одновременно следующим двум условиям:

- на пересечении каждой строки и каждого столбца данных подмножеств находится 1;
- к данным подмножествам нельзя добавить ни строку, ни столбец без нарушения первого условия.

При этом из всего множества блоков надо выбрать некоторое подмножество, включающее все элементы 1 заданной матрицы. А для получения необходимого условия вершинной минимизации недетерминированных конечных автоматов из всех таких подмножеств необходимо выбрать содержащее минимальное число элементов-блоков. Далее мы будем рассматривать несколько различных метрик – и сначала введём общее обозначение. Пусть X и Y – некоторые множества, $n=|X\cap Y|$ – число элементов их пересечения, $N=|X\cup Y|$ – число элементов их объединения. Тогда будем писать $\Omega(X,Y)=1–n/N$.

Рассмотрим эвристику для определения метрики на множестве блоков. В результате практического программирования, включавшего проверку нескольких альтернативных вариантов, для данной задачи наиболее удачной признана следующая. Пусть X_1 и X_2 – множества номеров строк 1-го и 2-го блока, Y_1 и Y_2 – аналогичные множества для столбцов. Тогда в качестве метрики используем значение $(\alpha\cdot\Omega(X_1,X_2)+(1-\alpha)\cdot\Omega(Y_1,Y_2))$, где α – некоторый коэффициент, настраиваемый с помощью применения генетических алгоритмов.

На основе введённой ранее метрики на блоках определим метрику на подзадачах. Для этого сначала определим метрику для двух произвольных массивов блоков. Пусть $X=\{x_1,\ldots,x_m\}$ – первый массив, $Y=\{y_1,\ldots,y_n\}$ – второй. Тогда сама метрика есть $\rho(X,Y) = (\sum_{i,j}\Omega(x_i, y_j)) / (m\cdot n)$, где i и j принимают все возможные значения. Далее, пусть теперь X_1 и Y_1 – массивы для первой подзадачи, X_2 и Y_2 – соответствующие массивы для второй. Обозначим $A = \rho(X_1,X_2)$, $B_1 = \rho(X_1,Y_2)$, $B_2 = \rho(X_2,Y_1)$, и в качестве расстояния между подзадачами будем использовать значение $(1+2\cdot\alpha)\cdot A - \alpha\cdot(B_1+B_2)$, где α, как и ранее, – некоторый коэффициент, настраиваемый с помощью генетических алгоритмов.

При использовании этой метрики для решения подзадач «по аналогии», о котором было сказано выше, получаются удачные версии программ. Однако данная метрика требует предварительного вычисления расстояний для всех пар блоков, сгенерированных к данному моменту. Поэтому авторами чаще используется другая метрика, в чём-то аналогичная метрике, введённой выше для самих блоков.

Итак, приведём ещё один вариант метрики на множестве подзадач. Пусть P_1 – множество клеток матрицы первой подзадачи, имеющих значение 1; P_2 – то же для второй подзадачи. Тогда в качестве метрики используем значение $\Omega(P_1,P_2)$. Отметим, что при этом сами формулы получаются значительно проще, чем в предыдущем варианте, однако, если на множестве блоков вычисления метрики проведены заранее, то вычисления, соответствующие второму варианту определения метрики на подзадачах, работают существенно дольше.

Список литературы

[1] V. Lifschitz. “Frames in the Space of Situations”, Artificial Intelligence 46 (1990) 365–376.

[2] J. Carbonell. “Derivational Analogy and Its Role in Problem Solving”, AAAI (1983) 64–69.

[3] B. Melnikov. “Once More About the State-Minimization of the Nondeterministic Finite Automata”, The Korean Journal of Computational and Applied Mathematics, Vo.7, No.3. (2000) 655–662.

[4] B. Melnikov. “Discrete Optimization Problems – Some New Heuristic Approaches”, 8th International Conference on High Performance Computing and Grid in Asia Pacific Region, IEEE Computer Society Press Ed. (2005) 73–80.

[5] Мельникова Е. А., Радионов А .Н. Построение классификационной иерархии документов на основе лексико-частотных характеристик с применением функций риска. //Труды II Всеросс. научной конф. «Методы обработки информации» стр. 103-106 , М., Изд-во МГУ, 2005

[6] Белозёрова А. Р., Мельников Б. Ф. Применение комплекса эвристик в задаче составления схемы нуклидных превращений. //Труды II Всеросс. научной конф. «Методы и средства обработки информации» стр. 208–212, М., Изд-во МГУ, 2005

Специальные методы принятия решений для разработки экспертных систем в области микробиологии

Мельников Борис Феликсович

доктор физ.-мат. наук, профессор каф. прикладной математики

Тольяттинский государственный университет г.Тольятти

Адрес: 445667, г. Тольятти, Белорусская, 14, ТГУ

e-mail: bormel@mail.ru

Тарасова Елена Геннадьевна

к.т.н., вед. программист ОГС ЦТКиТИ

Ульяновский государственный университет г.Ульяновск

Адрес: 432059, г.Ульяновск, ул. 40летия Победы 5-113

e-mail: tarasova@uven.ru

В статье рассмотрено применение эвристических методов для разработки систем принятия решений, опирающихся на опыт и знания экспертов. Ранее используемый в работе метод (метод функций риска) применялся для решения различных задач дискретной оптимизации: коммивояжера, минимизации

недетерминированных конечных автоматов, выбора конкретного варианта ветвления в незавершенном методе ветвей и границ. В данной статье представлено решение одной из прикладных задач области микробиологии.

В настоящей статье рассматриваются вопросы исследования проблем области микробиологии, связанные с дисбиотическими изменениями кишечника. Известно, что при физиологической норме организм содержит сотни видов бактерий, а также вирусы и простейшие, составляющие в сумме нормальный микробиоценоз. Под воздействием дестабилизирующих факторов микрофлора кишечника реагирует качественными и количественными изменениями в популяциях микроорганизмов, анализ и оценку которых необходимо осуществить в ходе исследования. Современные представления о дисбактериозах требуют поиска новых подходов к их оценке и характеристике.

В проведенном исследовании было предложено использование метода функций риска для осуществления кластеризации и оценки, определенных в задаче групп данных различных случаев (контрольный, переходный, критический). Данные по количественному и качественному составу бактерий получены на основе запатентованного бактериологического анализа экспертами области.

Существующие подходы к анализу данных в рассматриваемой предметной области сводятся как правило к оценке эффективности каких-либо препаратов (либо совокупности) и их воздействия на организм. Для этого используются различные корреляционные методы и определяются оценки значимости полученных результатов.

Данное исследование основано на знаниях экспертов, полагается что они имеют схемы лечения. Задача сводится к построению системы принятия решений для оценки данных по конкретным больным и этапам развития дисбиоза, с накоплением всевозможных состояний. Для этого используются разработанные автором алгоритмы и реализованные на их основе программные комплексы в составе экспертной системы для получения оценок этапов развития дисбактериоза.

Таким образом, можно перечислить основные цели для решения поставленной задачи:

- Разработка подхода к построению систем поддержки принятия решений как способа анализа и оценки значимых этапов изучаемого процесса для решения задач определенных на этапе идентификации.

- Разработка модели анализа и оценки ситуаций, включающей:

  - функциональную модель, представляющую собой записанные с помощью эвристического и математического аппаратов зависимости между параметрами значимых этапов исследуемых процессов;
  - соответствующее алгоритмическое обеспечение;
  - программные средства, реализующие разработанные алгоритмы и выполняющие функцию анализа и оценки параметров, работу с базами данных в системе.

Разработанный для данного исследования подход – это способ анализа и оценки ситуаций, основных этапов процесса, а также его изучение. Согласно подходу, в первую очередь выявляются и анализируются наиболее характерные и значимые этапы изучаемого процесса, определяющие дальнейшее функционирование и развитие системы принятия решений. На таких значимых этапах построения системы принятия решений, как идентификация, выполнение и тестирование, происходит непосредственное общение с экспертами областей – для получения знаний и экспертной оценки полученных результатов.

В основу методики получения знаний от экспертов положен механизм извлечения экспертного знания путем сбора и обработки опыта всех участников по решению конкретной практической задачи. Опыт эксперта представляется проведенными разработками с использованием локальных методов, характерных для данной области, а также принятыми решениями на различных этапах экспериментов.

Методика получения знаний от экспертов представляет собой совокупность следующих процедур:

- сбор первичной информации, на основе данных диагностических методов, получивших положительную оценку эксперта в результате практической деятельности;
- использование опыта и знаний экспертов для совместной разработки и создания ситуационной модели, а также определение правил исследования;
- использование опыта и знаний экспертов для оценки полученных результатов исследования и целесообразности развития и использования системы принятия решений.

Схожая методика используется для создания прецедентных экспертных систем, обеспечивающих возможность поддержки принятия решения по инциденту путем предоставления пользователю сведений об аналогичных случаях. «В основе поддержки принятия решения методом прецедентов находится последовательность трех процессов: классификации полного пространства прецедентов, идентификации нового прецедента и извлечении аналогов (аналога), адаптации (применения) решений выделенных аналогов (аналога) для нового прецедента»[1].

К преимуществам использования опытных данных экспертов относятся следующие аспекты:

1. Рассуждения на основе опыта позволяют избежать прошлых ошибок и использовать прошлые успехи. В итоге получается достаточно практичная модель.
2. Не требуется всестороннего анализа знаний о данной области. Рассуждения на основе опыта обеспечивают аддитивную модель приобретения знаний.
3. На основе опыта выявляются отличительные особенности целевой проблемы, и выбирается соответствующий случай из базы данных.

Итак, подход построения системы на основе приобретения знаний от экспертов содержит такие этапы как идентификация, концептуализация, формализация, реализация, тестирование. Индивидуальность подхода заключается в принципах, которые реализуются на каждом из перечисленных этапов.

На этапе идентификации для разработки ситуационной модели определяются задачи, которые подлежат решению. Выявляются цели разработки, состоящие в получении динамической системы принятия решений для анализа и оценки ситуаций в поставленной задаче. Определяются вычислительные ресурсы и группы экспертов в области.

На этапе концептуализации производится анализ проблемной области с участием экспертов, создается ситуационная модель. Инженером по знаниям (автором статьи) определяется метод решения задач (эвристический метод функций риска), проектируется структура баз данных для задачи.

На этапе формализации создается модель анализа и оценки ситуаций, содержащая функциональную модель на основе эвристического метода функций риска и локальных методик области, ситуационную модель для задачи данной прикладной области, базу данных области, алгоритмы, которые реализуют функциональную модель области.

На этапе реализации выбираются язык программирования и платформа, происходит преобразование формальных знаний в работающие программы по анализу и оценке ситуаций, а также по работе с используемыми структурами данных. Осуществляется наполнение базы знаний системы.

На этапе тестирования проводится совместная с экспертом проверка работы созданного варианта системы на репрезентативных задачах. В процессе тестирования анализируются возможные источники ошибок.

Для достижения целей исследования разработана модель анализа и оценки ситуаций в соответствии с подходом к построению систем принятия решений. При разработке данной модели учитывались все принципы подхода. Таким образом, разработанная автором модель анализа и оценки ситуаций состоит из:

- Ситуационной модели, уникальной для задачи и предметной области.
- Функциональной модели, построенной на основе эвристического метода функций риска, – как способа оценки ситуаций в предметной области.
- Модели системы принятия решений в исследуемой области.

Первоначальная задача инженера по знаниям и эксперта предметной области заключается в создании ситуационной модели по имеющимся экспертным фактам конкретной задачи и заполнении баз данных.

Создание ситуационной модели (СМ), индивидуально для каждой решаемой задачи. В ее основе лежит принцип группировки данных о значимых этапах изучаемого процесса экспертами на основе личного опыта и результатов практической деятельности. Ситуация – это описание состояния процесса, явления. Построение СМ в каждой области зависит от таких качеств эксперта как профессионализм, опыт и желание сотрудничать. Ситуационные модели в каждой задаче основаны, прежде всего, на базе знаний (БЗ), которая содержит базу ситуаций и базу решений. База ситуаций (БС) напрямую связана с базой решений (БР), каждой ситуации соответствует свое решение (система состояний). Каждая ситуационная модель в процессе работы системы принятия решений будет расширяться за счет появления новых ситуаций, их решений (а именно, оценок), а в итоге за счет увеличения базы знаний.

На этапе получения знаний от экспертов, при построении ситуационной модели, программисту не всегда удается точно записать мнение эксперта. «Основной трудностью создания ЭС становится получение знаний от эксперта, так как этот процесс зависит от многих факторов: предметной области, методики получения знаний, квалификации эксперта и способности выявлять его собственные знания»[2]. Однако подобный процесс необходимо осуществить на первом шаге построения систем принятия решений – для возможности дальнейшего создания полноценной системы, более того, такой процесс, по-видимому, является наиболее важным этапом создания экспертной системы.

После построения ситуационной модели инженером по знаниям должен быть проведен предварительный анализ ситуаций в задаче, который состоит в следующем:

- Совместно с экспертами области производится отбор первоначальных групп, характеризующих наиболее значимые ситуации процесса. Ранее было сказано, что каждый эксперт области имеет некоторый набор локальных методик, позволяющих диагностировать процессы, явления (в данной работе заболевание) и получать числовые данные по конкретным параметрам, анализ и оценку этих параметров необходимо произвести программисту.
- После того как создана ситуационная модель, инженером по знаниям (экспертом по оценке ситуаций) принимается решение о том, что все группы (ситуации) в ситуационной модели определяют детерминированную задачу с конкретным количеством групп.
- Затем экспертом по оценке реализуется эмпирический выбор функций определения весовых коэффициентов, согласно принципам метода, с оповещением экспертов областей.
- Полученные результаты рассматриваются и оцениваются экспертами областей, с учетом имеющихся личных разработок и опыта. А также принимают решение о целесообразности дальнейших исследований с использованием разработанной системы.

Для корректной работы эвристического метода нами производился эмпирический выбор класса функций для определения весовых коэффициентов. После проведенного предварительного процесса статической оценки можно считать, что система отработала с помощью инженера по знаниям один из важных этапов обучения. Вся информация, полученная на этом шаге, является основополагающей для дальнейшего развития динамической системы принятия решений и заносится в базу данных. Поэтому данный этап разработки является наиболее трудоемким как для экспертов по оценке ситуаций, так и для экспертов областей.

Функциональная модель представлена в работе записанными с помощью эвристического и математического аппаратов зависимостями между параметрами исследуемых процессов. В основу алгоритма решения задачи положен эвристический метод принятия решений - метод функций риска, который позволил получить объективную оценку ситуаций в поставленной задаче.

«На практике наиболее часто встречаемая при расчетах средних ошибка заключается в игнорировании весов в тех случаях, когда эти веса в действительности необходимы»[3]. Метод функций риска позволяет учитывать веса при оценке ситуаций, тем самым приближая полученные результаты к реальным. В ходе исследования нами эмпирически определяется класс функций, а именно выбраны квадратичные функции, для предварительного анализа групп в задаче.

Для реальной оценки ситуаций критические и нестабильные реализации фактов стоит считать наиболее вероятными, чем контрольные. Так в рассматриваемой области наиболее вероятными являются проблемные случаи (случаи заболевания), которые представляют интерес для исследования, как для экспертов области, так и для экспертов по оценке ситуаций.

Ситуация определена в контексте метода, если для нее заданы функции определения весовых коэффициентов (функции риска). На основе возможных значений весовых коэффициентов и исследуемого вариационного ряда получаем набор оценок исследуемого параметра (оценка ситуации). Представленная Эвристика (1) используется для решения задачи исследуемой области, что позволяет анализировать и оценивать определенные экспертами области ситуации.

Эвристика 1. $$\bar{x} = \frac{\sum_i x_i * f(x_i)}{\sum_i f(x_i)},$$

как x_i представлены сгруппированные данные исследуемой области, имеющие общую размерность;

$f(x_i)$ – весовая оценка, представленная функцией риска или функцией определения весовых коэффициентов.

Для улучшения оценки групп достаточно предположить оценку для различных ситуаций. Так, если ситуация близка к контрольной (рассматривается здоровый организм), то функция для определения весовых коэффициентов выбирается убывающая (отношение к ситуации пессимистическое). Если ситуация ухудшается, то функция риска приближается к константе. А если ситуация является критической (обнаружено заболевание), то функция риска для определения весовых коэффициентов выбирается возрастающая.

Выбор конкретного вида функций связан с распределением значений весовых коэффициентов. Так при выборе возрастающих функций (рис. 1) для критических ситуаций большие значения весовых коэффициентов присваиваются большим значениям исследуемого параметра (что позволяет задать

критическую ситуацию в модели), т.е. значения весовых коэффициентов возрастают с увеличением значений исследуемого параметра.

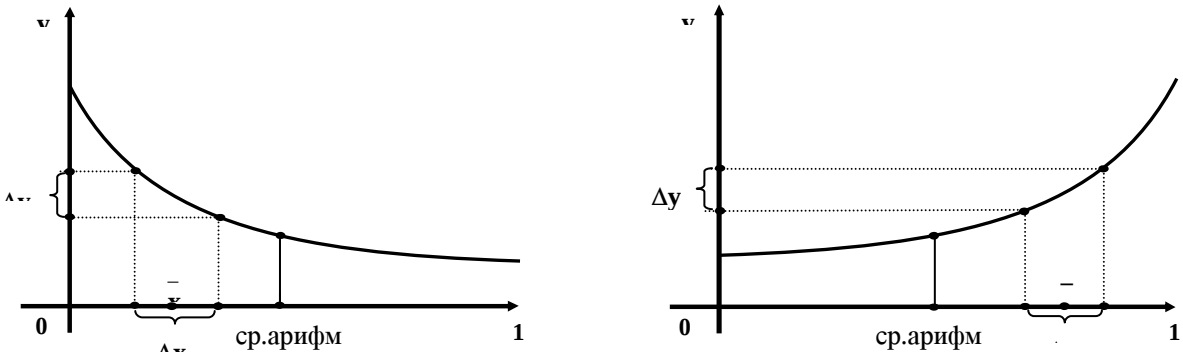


Рис.1 Функции риска. Слева функции для положительного исхода или переходного случая.

Справа функции для критического случая.

Для построения полноценной системы принятия решений необходимы эвристические методики, способные без участия экспертов оценивать ситуации и производить сопоставления с имеющимися в базе результатами предыдущих экспериментальных исследований. В работе вводится понятие динамических функций риска. На основе предварительной оценки групп, состоящей из оценок наиболее значимых параметров процесса (стадий), должна строиться оценка вновь появляющихся групп, предлагаемых к анализу экспертами области.

«Динамическая оценка позиции определяется на основе всех детерминированных оценок, полученных для всех возможных значений ситуации. Полученные значения детерминированных (обычно-статических) оценок специальным образом усредняются - это и есть получаемая динамическая оценка. С физической точки зрения такое усреднение даёт положение центра тяжести одномерной системы тел с массами, определяемыми специально подобранной функцией (функцией риска)» [4]. Важно то, что применение динамической функции риска может быть проведено более одного раза, что может улучшить динамическую оценку позиции поскольку конкретная функция риска выбирается на основе всё более достоверной априорной информации.

В результате выполненной работы получена система поддержки принятия решений, содержащая базу для хранения данных ситуационной модели и результатов экспериментальных исследований, алгоритм анализа и оценки этапов дисбиоза, программное обеспечение на основе разработанного алгоритма. Реализованная система позволяет производить анализ и оценку представленных экспертами области ситуаций, накапливать информацию и осуществлять модернизацию (преобразование) имеющихся данных (оценок) на основе поступающей все более достоверной информации.

Список литературы

1. Берман А. Ф., Николайчук О. А., Юрин А. Ю., Павлов А. И. Идентификация прецедентов для определения причин отказов // Материалы Всероссийской конференции «Инфокоммуникационные и вычислительные технологии и системы». – Улан-Удэ. – Издательство Бурятского Госуниверситета. – 2003. – Часть 1.– С. 48-53.

2. Андреев Ю. В., к.т.н., Попов А. В. Экспертные системы – современное состояние и перспективы развития // Тезисы докладов международной конференции «Копьютерные технологии в образовании и предпринимательстве», ЧГТУ, 2001.

3. Теория статистики: Учебник / Под ред. Проф. Г. Л. Громыко. - 2-е изд. - М.: ИНФРА-М, 2005. – 476с.

4. Б.Мельников. Эвристики в программировании недетерминированных игр. – «Програмирование» (Известия РАН), 2001, №5, с.63–80.

5. Melnikov B. Discrete optimization problems – some new heuristic approaches // 8th International Conference on High Performance Computing and Grid in Asia Pacific Region. – 2005. – P.73–80.

Верификация функциональных программ на основе построения их графовых моделей

Миронов А. М.
Институт Проблем Информационной Безопасности МГУ
E-mail: mironov@iisi.msu.ru

Введение

Одна из возможных постановок задачи верификации функциональных программ (ФП) имеет следующий вид: доказать, что две ФП вычисляют одну и ту же функцию.

Понятие функциональной программы изложено во многих источниках (отметим, например, [1]). В настоящем исследовании мы рассматриваем упрощённое понятие функциональной программы, в котором не используются сложные типы данных, конструкции `let` и `where`, сопоставления с образцом, и лямбда-определения. Под функциональной программой (ФП) мы будем понимать определение некоторой совокупности функций при помощи системы функциональных уравнений.

Переменные, константы, функциональные символы

Мы предполагаем, что заданы

- множество $\mathcal{D}$, элементы которого называются **значениями**,
- множество Con **констант**, причём каждой константе $c \in Con$ сопоставлено некоторое значение, обозначаемое тем же символом c,
- множество Var, элементы которого называются **переменными**, причём каждой переменной $x \in Var$ сопоставлено множество $\mathcal{D}_x \subseteq \mathcal{D}$ значений, которые может принимать эта переменная,
- множество Fun, элементы которого называются **функциональными символами (ФС)**, причём каждому $f \in Fun$ сопоставлено натуральное число $ar(f)$, называемое **арностью ФС** f.

Некоторые ФС называются **базовыми**. Каждому базовому ФС сопоставлена некоторая частичная функция, которая обозначается тем же символом f, и имеет вид $f : \mathcal{D}^{ar(f)} \to \mathcal{D}$.

Мы будем предполагать, что множество `Fun` содержит базовый ФС `if_then_else` арности 3. Соответствующая этому символу функция сопоставляет каждой тройке $(d_1, d_2, d_3) \in \mathcal{D}^3$

- элемент d_2, если $d_1 = 1$
- элемент d_3, если $d_1 \neq 1$.

Для каждой тройки $(d_1, d_2, d_3) \in \mathcal{D}^3$ знакосочетание `if_then_else` (d_1, d_2, d_3) будет сокращённо записываться в виде $d_1 \ ?\ d_2 : d_3$. Ниже символ Fun^+ обозначает совокупность всех ФС, отличных от `if_then_else`.

Выражения, подстановки, означивания

Выражения строятся из переменных, констант и функциональных символов. Множество выражений определяется следующим образом.

1. Каждая переменная или константа является выражением.
2. Для каждого списка выражений $e_1, \ldots, e_n$, и каждого ФС f арности n, знакосочетание $f(e_1, \ldots, e_n)$ является выражением.

Подстановка – это знакококосочетание θ вида

$$\theta \ = \ [x_1 := e_1, \ldots, x_k := e_k] \tag{1}$$

где $x_1, \ldots, x_k$ – список различных переменных, и $e_1, \ldots, e_k$ – выражения.

Для каждой подстановки θ вида (1) и каждого выражения e знакосочетание θe обозначает выражение, получаемое из e заменой для каждого $i = 1, \ldots, k$ каждого вхождения переменной x_i в e на выражение e_i.

Означиванием называется частичная функция вида $\xi : Var \to \mathcal{D}$.

Для каждого выражения e **значение** $\xi(e)$ выражения e на означивании ξ считается определённым, если имеет место один из следующих случаев.

1. $e = x \in Var$, и $\xi(x)$ определено. В этом случае $\xi(e) \stackrel{\text{def}}{=} \xi(x)$.

2. $e = \ \in Con$. В этом случае $\xi(e) \stackrel{\text{def}}{=} c$.

3. e имеет вид $e_1 \ ? \ e_2 \ : \ e_3$, причём $\xi(e_1)$ определено. В этом случае

   - если $\xi(e_1) = 1$, то $\xi(e) \stackrel{\text{def}}{=} \xi(e_2)$
     (т.е. значения $\xi(e)$ и $\xi(e_2)$ либо оба не определены, либо оба определены и совпадают)
   - если $\xi(e_1) \neq 1$, то значение $\xi(e) \stackrel{\text{def}}{=} \xi(e_3)$

4. e имеет вид $f(e_1, \ldots, e_k)$, причём

   - $f \in Fun^+$
   - символу f сопоставлена некоторая частичная функция
   - значения $\xi(e_1), \ldots, \xi(e_k)$ определены,
   - значение функции f на списке $(\xi(e_1), \ldots, \xi(e_k))$ определено.

   В этом случае $\xi(e) \stackrel{\text{def}}{=} f(\xi(e_1), \ldots, \xi(e_k))$.

Для каждого означивания ξ, каждой переменной $x \in Var$ и каждого значения $d \in \mathcal{D}_x$ знакосочетание $\xi[x := d]$ обозначает означивание, которое отличается от ξ только значением на x:

$$\forall y \in Var \qquad \xi[x := d](y) \stackrel{\text{def}}{=} \begin{cases} \xi(y), & \text{если } y \neq x \\ d, & \text{если } y = x \end{cases}$$

(Для каждой пары ξ_1, ξ_2 означиваний, и каждой переменной $x \in Var$ знакосочетание $\xi_1(x) = \xi_2(x)$ означает, что значения $\xi_1(x)$ и $\xi_2(x)$ либо оба не определены, либо оба определены и совпадают).

Функциональные программы

Функциональная программа (ФП) имеет вид системы уравнений

$$\begin{cases} f_1(x_{11}, \ldots, x_{1k_1}) = e_1 \\ \ldots \\ f_n(x_{n1}, \ldots, x_{nk_n}) = e_n \end{cases} \tag{2}$$

и представляет собой определение некоторого множества функций, где

- $f_1, \ldots, f_n$ – ФС, являющиеся именами определяемых функций,
- $x_{11}, \ldots, x_{nk_n}$ – различные переменные, являющиеся формальными параметрами определяемых функций, и
- $e_1, \ldots, e_n$ – выражения, обладающие следующими свойствами: для каждого $i = 1, \ldots, n$
  - совокупность всех переменных, входящих в выражение e_i, совпадает с множеством $\{x_{i1}, \ldots, x_{ik_i}\}$,
  - каждый ФС, входящий в e_i,
    - или является базовым,
    - или содержится в множестве $\{f_1, \ldots, f_n\}$.

Функции, определяемые функциональными программами

Функции, определяемые функциональными программами, вычисляются стандартной рекурсией: если требуется вычислить значение функции f_i, которая определяется системой уравнений (2), на списке аргументов $(d_1, \ldots, d_{k_i})$, то для этого вычисляется значение выражения

$$[x_{i1} := d_1, \ldots, x_{ik_i} := d_{k_i}]e_i \tag{3}$$

Вычисление значения выражения (3) происходит по следующему рекурсивному правилу.

- Если данное выражение равно константе, то его значение равно значению этой константы.
- Если данное выражение имеет вид
$$u_1 \; ? \; u_2 \; : \; u_3 \tag{4}$$
то сначала вычисляется значение выражения u_1, и
  - если значение выражения u_1 не определено, то значение всего выражения (4) считается неопределённым,
  - если значение выражения u_1 равно 1, то вычисляется значение выражения u_2, и
    - если значение выражения u_2 не определено, то значение выражения (4) не определено,
    - в противном случае – значение выражения u_2 по определению равно значению выражения (4)
  - если значение выражения u_1 отлично от 1, то вычисляется значение выражения u_3, и
    - если значение выражения u_3 не определено, то значение выражения (4) не определено,
    - в противном случае – значение выражения u_3 по определению равно значению выражения (4)
- Если данное выражение имеет вид
$$f(u_1, \ldots, u_m) \tag{5}$$
где f – базовый ФС, отличный от `if_then_else`, то вычисляются значения выражений $u_1, \ldots, u_m$, после чего вычисляется значение функции f на списке значений выражений $u_1, \ldots, u_m$, которое по определению считается равным значению выражения (5). Если
  - одно из значений выражений $u_1, \ldots, u_m$ не определено, или
  - значение функции f на списке значений u_1, …, u_m не определено

  то значение всего выражения (5) считается неопределённым.
- Если данное выражение имеет вид $f_j(u_1, \ldots, u_{k_j})$, где f_j – ФС, являющийся именем одной из определяемых функций в системе (2), то его значение равно значению выражения
$$[x_{j1} := u_1, \ldots, x_{jk_j} := u_{k_j}]e_j \tag{6}$$
которое вычисляется по тому же правилу, по которому вычисляется значение выражения (3).

Если описанный выше процесс вычисления значения $f_i(d_1, \ldots, d_{k_i})$ не завершается, то мы будем считать, что значение f_i на данном списке аргументов не определено.

Графовые модели функциональных программ

Понятие графовой модели ФП

По ФП (2) можно построить граф, называемый **графовой моделью (ГМ)** ФП (2), в котором

- каждая вершина имеет метку, совпадающую с некоторым подвыражением одного из выражений, входящих в (2), и
- каждое ребро имеет метку одного из следующих типов:
  1. **условие**: метка данного типа имеет вид φ ? или $\neg\varphi$?, где φ либо некоторое подвыражение одного из выражений $e_1, \ldots, e_n$, либо константа 1.

2. **подстановка**: метка данного типа имеет вид $[x_1 := u_1, \ldots, x_m := u_m]$, где $x_1, \ldots, x_m$ – некоторые различные переменные, входящие в (2), и $u_1, \ldots, u_m$ – некоторые подвыражения выражений, входящих в (2).
3. **ссылка на подвыражение**: метка данного типа имеет вид (i), и имеет следующий смысл: для каждого ребра с меткой (i) метка вершины, являющейся началом данного ребра, имеет вид $f(u_1, \ldots, u_m)$, и метка вершины, являющейся концом данного ребра, равна u_i.

ГМ строится по ФП (2) следующим образом.

- Множество её вершин находится во взаимно-однозначном соответствии с множеством всех подвыражений выражений, входящих в (2). Меткой каждой вершины является соответствующее ей подвыражение.
- Для каждого уравнения $f_i(x_{i1}, \ldots, x_{ik_i}) = e_i$, входящего в (2), ГМ содержит ребро с меткой 1? из вершины с меткой $f_i(x_{i1}, \ldots, x_{ik_i})$ в вершину с меткой e_i.
- Для каждой вершины N с меткой вида $e_1 \; ? \; e_2 \; : \; e_3$, ГМ содержит
  - ребро из N в вершину с меткой e_2, метка этого ребра равна $e_1?$, и
  - ребро из N в вершину с меткой e_3, метка этого ребра равна $\neg e_1?$.
- Для каждой вершины N с меткой вида $f_i(u_1, \ldots, u_{k_i})$, где f_i – имя одной из функций, определяемых в ФП (2), ГМ содержит ребро из N в вершину с меткой $f_i(x_{i1}, \ldots, x_{ik_i})$, метка этого ребра имеет вид $[x_{i1} := u_1, \quad \ldots, \quad x_{ik_i} := u_{k_i}]$.
- Для каждой вершины N с меткой вида $f(u_1, \ldots, u_m)$ где $f \in Fun^+$, и каждого $i = 1, \ldots, m$ ГМ содержит ребро из N в вершину с меткой u_i, метка этого ребра равна (i).

Верификация ФП при помощи ГМ

Пусть заданы две ФП, в первой из которых определяются функции $f_1, \ldots, f_n$, а во второй - $g_1, \ldots, g_m$. Если предположительно известно, что функции f_1 и g_1, определяемые этими ФП, совпадают (при условии, что аргументы этих функций удовлетворяют некоторому соотношению, которое мы ниже будем называть **предусловием**), и требуется доказать это формально, то это можно сделать следующим способом.

Сначала переименуем переменные в этих ФП, так, чтобы каждая входящая в них переменная входила только в одно уравнение одной из этих ФП. Каждое из этих переименований происходит только в пределах одного из уравнений, и заключается в замене всех вхождений некоторой переменной в это уравнение на новую переменную, не входящую в эти ФП. Затем построим графовые модели обеих ФП.

Теорема.

Пусть существует совокупность дуг, соединяющих вершины первой ГМ с вершинами второй ГМ, и удовлетворяющих следующим условиям.

1. *Каждая из этих дуг имеет метку, представляющую собой булевозначное выражение, зависящее от переменых, входящих в метки вершин, которые соединяет эта дуга.*
2. *Среди этих дуг присутствует дуга,*
   - *которая соединяет вершину в первой ГМ с меткой вида $f_1(x_{11}, \ldots, x_{1n_1})$, с вершиной во второй ГМ с меткой вида $g_1(y_{11}, \ldots, y_{1m_1})$, и*
   - *метка которой изображает предусловие.*
3. *Если*
   - *в одной из ГМ есть ребро с меткой θ из некоторой вершины v_1 в вершину v_1',*
   - *v_2 – некоторая вершина другой ГМ*
   - *вершина v_1 соединена с вершиной v_2 дугой с меткой α,*

   то вершина v_1' соединена с вершиной v_2 дугой с меткой α', такой, что истинна импликация $\alpha \to \theta\alpha'$.

4. *Если*

   - *в одной из ГМ есть ребро с меткой* φ*? из некоторой вершины* v_1 *в вершину* v_1'*,*
   - v_2 *– некоторая вершина другой ГМ*
   - *вершина* v_1 *соединена с вершиной* v_2 *дугой с меткой* α*,*

   то вершина v_1' *соединена с вершиной* v_2 *дугой с меткой* α'*, такой, что истинна импликация* $(\alpha \wedge \varphi) \to \alpha'$*.*

5. *Если*

   - *вершина* v_1 *в одной ГМ соединена с вершиной* v_2 *в другой ГМ дугой с меткой* α*,*
   - *из* v_1 *и* v_2 *не выходит ни одного ребра*
   - *метками* v_1 *и* v_2 *являются выражения* e_1 *и* e_2 *соответственно*

   то истинна импликация $\alpha \to (e_1 = e_2)$*.*

6. *Если*

   - *вершина* v_1 *в одной ГМ соединена с вершиной* v_2 *в другой ГМ дугой с меткой* α*,*
   - *метка* v_1 *представляет собой базовый ФС*

   то в этом случае

   - *метка* v_2 *представляет собой тот же базовый ФС,*
   - *пусть*
     - *список концов рёбер, выходящих из* v_1*, имеет вид* $v_{11}, \ldots, v_{1n}$*, причём для каждого* $i = 1, \ldots, n$ *вершина* v_1 *соединена с вершиной* v_{1i} *ребром с меткой* (i)
     - *список концов рёбер, выходящих из* v_2*, имеет вид* $v_{21}, \ldots, v_{2n}$*, причём для каждого* $i = 1, \ldots, n$ *вершина* v_2 *соединена с вершиной* v_{2i} *ребром с меткой* (i)*,*

     тогда для каждого $i = 1, \ldots, n$ *вершина* v_{1i} *соединена с вершиной* v_{2i} *дугой с меткой* α_i*, причём истинна импликация* $\alpha \to (\alpha_1 \wedge \ldots \wedge \alpha_n)$*.*

7. *Если*

   - *в одной из ГМ есть*
     - *ребро с меткой* θ *из некоторой вершины* v_1 *в вершину* v_1'*,*
     - *из вершины* v_1 *выходят также рёбра с метками* $(1), \ldots, (n)$ *в вершины соответственно* $v_{11}, \ldots, v_{1n}$
     - *из вершины* v_1' *выходят рёбра с метками* $(1), \ldots, (n)$ *в вершины соответственно* $v_{11}', \ldots, v_{1n}'$
   - v_2 *– некоторая вершина другой ГМ, которая для каждого* $i = 1, \ldots, n$ *соединена с вершиной* v_{1i} *дугой с меткой* α_i

   то для каждого $i = 1, \ldots, n$ *вершина* v_2 *соединена с вершиной* v_{1i}' *дугой с меткой* α_i'*, такой, что истинна импликация* $\alpha_i \to \theta\alpha_i'$*.*

Тогда, если обе ФП завершают свою работу после конечного числа шагов на всех значениях аргументов, удовлетворяющих предусловию, то значения функций f_1 *и* g_1 *на этих значениях аргументов совпадают.*

Список литературы

1. Филд А., Харрисон П.: Функциональное программирование. М., Мир, 1993. - 637 с.

О приближении непрерывных функций нейронными сетями

Морнев М. Л.,
Математико-Механический Факультет,
г. Екатеринбург, ул. Ленина, 51
Уральский Государственный Университет им. А. М. Горького,
e-mail: max.mornev@gmail.com

Колмогоров в [1] доказал, что произвольная действительнозначная функция, непрерывная на n-мерном единичном кубе, точно представима суперпозицией непрерывных функций одной переменной и сложения. Это представление можно интерпретировать как нейронную сеть. К сожалению, теорема Колмогорова не дает информации о типе функций активации такой сети. Требованиям теории нейронных сетей больше отвечают аппроксимационные теоремы, использующие конкретный тип функций активации. В данном тексте приведен вариант такой теоремы для весьма широкого класса функций.

Пусть функция $h\colon R \to R$ такова, что существуют, конечны и различны пределы $A = \lim_{x\to-\infty} h(x)$, $B = \lim_{x\to+\infty} h(x)$, и для всякого x имеет место $\min(A,B) \leqslant h(x) \leqslant \max(A,B)$. Тогда для этой функции справедливы следующие две теоремы.

Теорема 1. *Для любой непрерывной функции $f\colon I \to R$ и любого $\varepsilon > 0$ можно так выбрать натуральное число m и действительные коэффициенты $a_1,\dots,a_m,b_1,\dots,b_m,c_1,\dots,c_m$, что для функции*

$$g(x) = \sum_{p=1}^{m} a_p h(b_p + c_p x)$$

будет иметь место $\sup_{x\in I} |f(x) - g(x)| < \varepsilon$.

Теорема 2. *Для любой непрерывной функции $f\colon I^n \to R$ и любого $\varepsilon > 0$ можно так выбрать натуральное число m и действительные коэффициенты $a_1,\dots,a_m,b_1,\dots,b_m,c_1,\dots,c_m,d_{11},\dots,d_{mn},e_{11},\dots,e_{mn}$, что для функции*

$$g(x_1,\dots,x_n) = \sum_{p=1}^{m} a_p h\Big[b_p + c_p \sum_{q=1}^{n} h(d_{pq} + e_{pq}x_q)\Big]$$

имеет место $\sup_{(x_1,\dots,x_n)\in I^n} \big|f(x_1,\dots,x_n) - g(x_1,\dots,x_n)\big| < \varepsilon$.

Отметим, что функция $h\colon R \to R$ не предполагается непрерывной: она может иметь разрыв в каждой точке.

Список литературы

[1] *А. Н. Колмогоров.* О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения. В кн.: ДАН СССР, 1957, т. 114, є5, с. 953-956.

Функциональная модель интеллектуальной системы составления расписания учебных занятий

Низамова Г. Ф., *кафедра Информатики*
Ковтуненко А. С.,
Уфимский государственный
авиационный технический университет
г. Уфа, ул. К-Маркса, 12
e-mail: nizamova_guzel@mail.ru, kobtyhehko@yandex.ru

Данная работа посвящена разработке структуры интеллектуальной системы составления расписания учебных занятий. Основной функцией данной системы является поиск оптимального расписания с помощью описанного в работе [2] генетического алгоритма. Кроме того, система должна выполнять и ряд других функций, в том числе позволять вводить необходимую при составлении расписания занятий информацию, обрабатывать данную информацию, заносить ее в специально созданную базу данных, использовать ее при составлении расписания, выбирать оптимальное расписание (если в ходе решения получено несколько вариантов) и предложить его пользователю. Важным также является учет опыта и знаний ЛПР (экспертов-методистов) для учета ряда требований, обусловленных спецификой конкретного образовательного учреждения.

Поэтому при выборе структуры данной системы представляется целесообразным использовать такие методы, которые наиболее полно отразят ее специфические особенности. Исходя из этого, система должна отвечать 2 основным группам требований (общим и специальным):

1) Общие требования: простота освоения и применения, универсальность (удобный для пользователя интерфейс), интегрируемость

2) Специальные требования: возможность решения многокритериальных задач генетической оптимизации; возможность решения информационно-поисковых задач, связанных с хранением, поиском и обработкой исходной для составления расписания информации.

Наиболее удобным при описании структуры таких систем, выполняющих несколько функций, является использование так называемых функциональных моделей. Функциональная модель любой системы представляет собой графическое описание этапов работы системы, при котором модель (структура) системы строится как иерархия диаграмм, описывающих процесс преобразования информации от ее входа до выдачи конечному пользователю. В данной работе в качестве инструментария для разработки функциональной модели системы составления расписания используется BPwin. Первый функциональный блок (рис. 1.) представляет анализируемый процесс в виде единого модуля (нулевой уровень), который отражает цель задачи проектирования, входные данные, используемые при решении задачи, механизмы, с помощью которых поставленная цель может быть достигнута и управление.

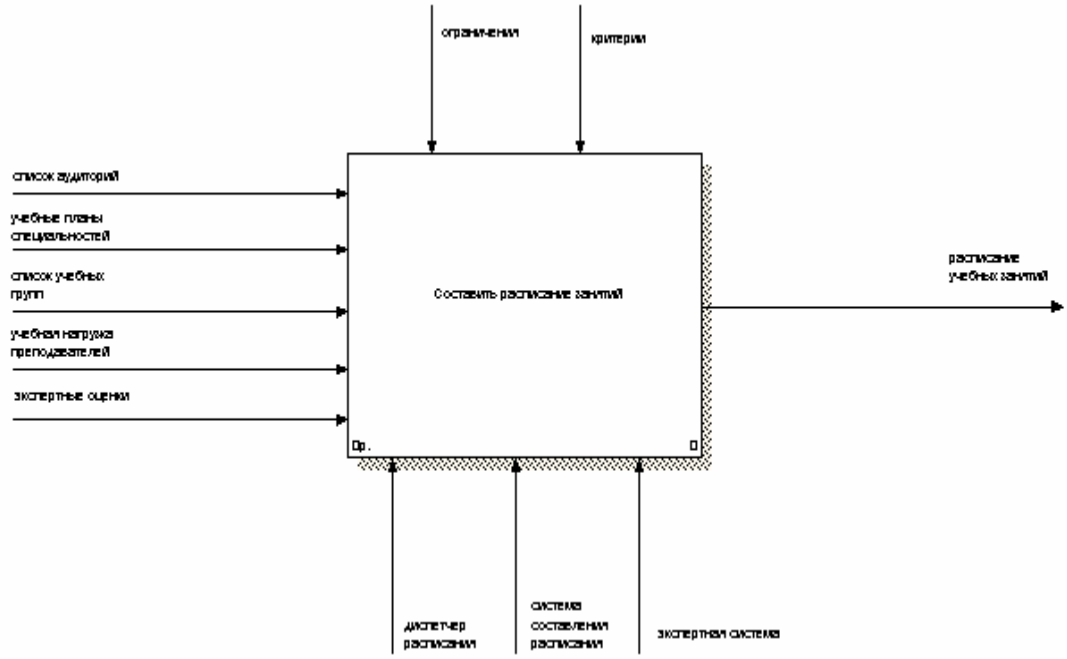


Рис. 1 Контекстная диаграмма интеллектуальной системы составления расписания учебных занятий

Представленная на рис. 1 диаграмма детализируется на другой диаграмме рис. 2 (первого уровня) с помощью трех блоков, которые описывают основные этапы, выполняемые системой при составлении расписания учебных занятий.

Этап 1. Ввод исходных данных. Под исходными данными понимаются сведения об учебных группах, дисциплинах, преподавателях, аудиториях и парах. На этом этапе также осуществляется приведение исходных данных к виду, необходимому для решения задачи (например, на данном этапе формируются блоки (циклы) занятий) [2].

Этап 2. Обработка экспертной информации.

Этап 3. Составление расписания учебных занятий.

Этап 4. Возможность выбора ЛПР конечного варианта расписания и вывод результатов.

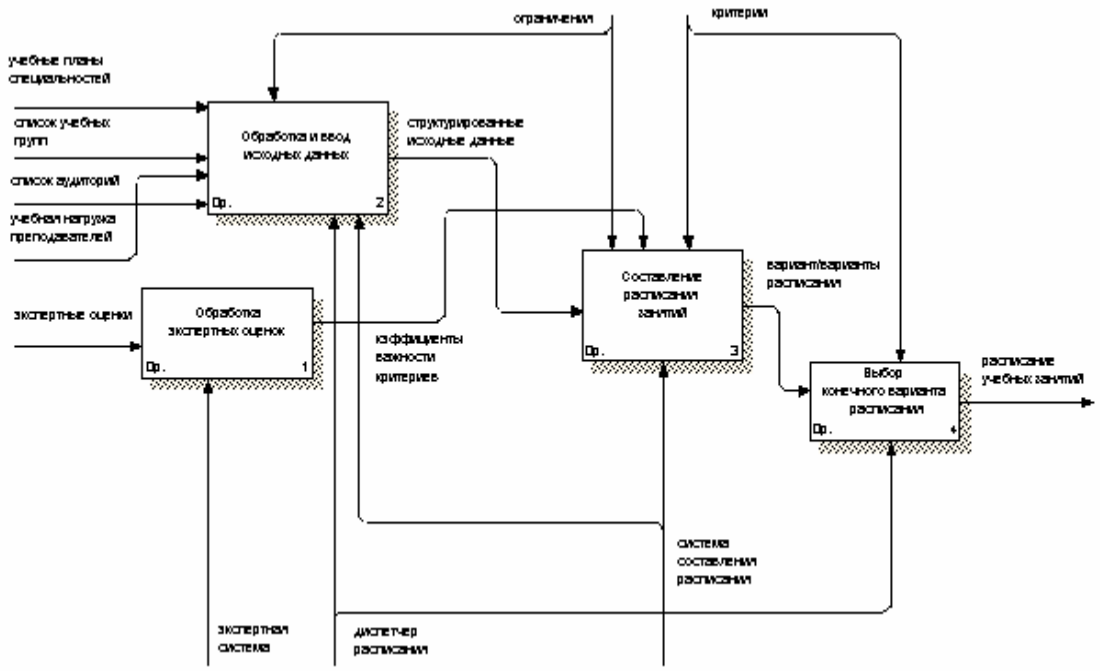


Рис. 2 Диаграмма первого уровня интеллектуальной системы составления расписания учебных занятий

Среди перечисленных этапов, реализуемых системой составления расписания, более детального рассмотрения заслуживает блок, который описывает процедуру составления расписания. Необходимо отметить, что реализация этапа 3 происходит с использованием агрегативного генетического алгоритма [3], схема которого приведена на рис. 3

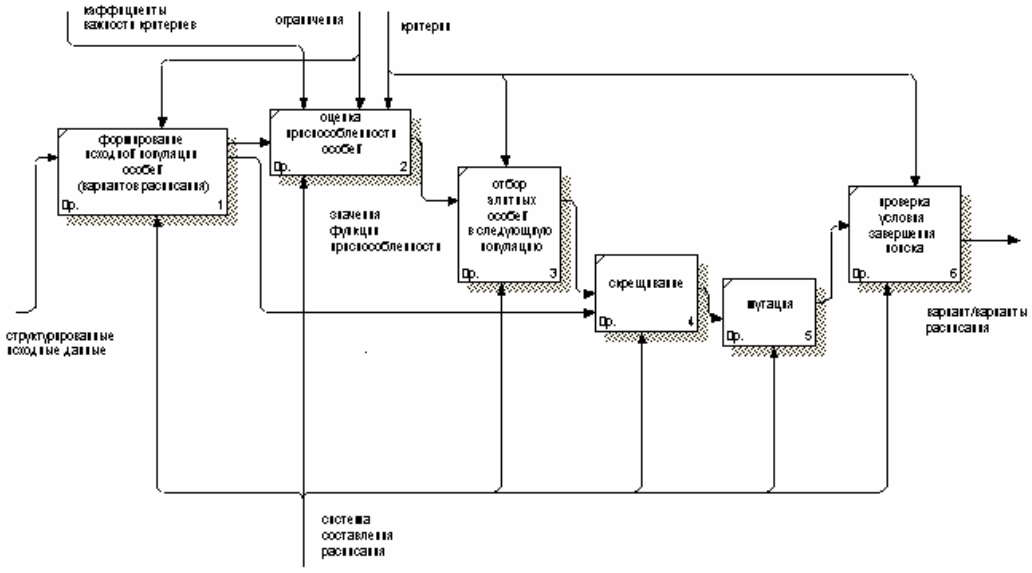


Рис. 3 Диаграмма-декомпозиция блока “Составление расписания занятий”

Следует сказать, что каждая особь генетического алгоритма представляет собой решение задачи и состоит из двух хромосом. Число генов первой и второй хромосомы равно числу выделенных циклов занятий. Информационным наполнением первой хромосомы являются номера аудиторий, допустимых для проведения планируемых занятий, второй хромосомы – номера пар для проведения данных занятий.

На следующем уровне (рис. 4) представлена укрупненная схема определения коэффициентов важности частных критериев оптимальности расписания. Напомним, что данный блок включает в себя процедуру коррекции коэффициентов важности с помощью аппарата лингвистических и вероятностных переменных.

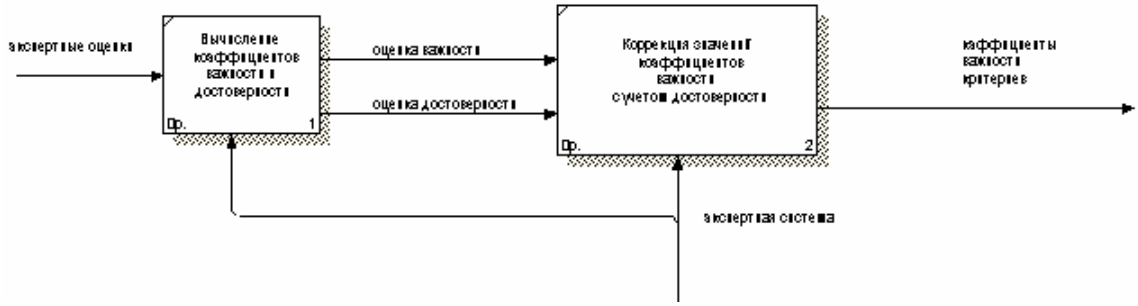


Рис. 4 Схема процедуры коррекции коэффициентов важности

Представленная в данной работе функциональная модель детально и последовательно описывает структуру и функции интеллектуальной системы. Система реализована на языке Object Pascal в среде Borland Delphi 7 и апробирована на разных наборах данных. Результаты показали эффективность применения разработанной системы для решения задачи составления расписания учебных занятий.

Список литературы.

1. E. D. Barraclough The application of a digital computer to the construction of timetables. The Computer Journal, 1965, № 8, с. 136-146.

2. Кабальнов Ю. С., Шехтман Л. И., Ковтуненко А. С. Генетический алгоритм составления расписания учебных занятий, основанный на структуризации исходной информации // Деп. научн. работа в ВИНИТИ, УГАТУ, 2006, 27 с.

3. Низамова Г. Ф. Генетический алгоритм составления расписания работы сложных систем (на примере образовательных систем массового обучения) // Интеллектуальные системы обработки информации и управления: Материалы Региональной зимней школы-семинара аспирантов и молодых ученых, Том 1, Уфа, 2006, с. 116-122.

Исследование работы системы передачи сообщений с помощью имитационной модели

Окороченко Г. Е.,
к.т.н., доцент кафедры "Компьютерные системы и технологии" МИФИ
E-mail: vladbor@inbox.ru
Бородакий В. Ю., *МИФИ.*
115409, Москва, Каширское шоссе, 31.

В распределённых вычислительных системах широко применяют межплатформенное связующее ПО, ориентированное на передачу сообщений (Messaging-oriented Middleware)[1]. Создание и эффективное использование такого ПО – это сложная задача, требующая современных методов проектирования и мощных инструментов исследования работы сетевых систем.

Исследуемая система передачи сообщений представляет собой множество приложений, осуществляющих информационное взаимодействие путем обмена сообщениями в синхронном или асинхронном режимах. Приложения работают на отдельных компьютерах или компьютерах, входящих в состав локальных вычислительных сетей.

Доступ к службе передачи сообщений осуществляют с помощью специальной интерфейсной библиотеки, которая взаимодействует с клиентской частью системы – локальной службой.

Среду передачи образует множество взаимосвязанных серверов – менеджеров очередей, которые взаимодействуют с локальными службами, а также осуществляют промежуточное хранение и маршрутизацию сообщений. Пересылка сообщений ведется с учетом их размера и срочности доставки (приоритета).

Приложения, локальная служба, менеджер очередей могут работать совместно на одном или на разных компьютерах, взаимодействуя по внутренней или внешней сети. Сети могут быть объединены с помощью каналов дальней связи.

Целью данной работы было построение и применение имитационной модели системы передачи сообщений в качестве инструмента для поиска и выбора эффективных решений при создании системы, а также на последующих этапах её установки, настройки и эксплуатации.

В результате проведенных исследований была разработана модель, удовлетворяющая следующим требованиям:

- описание вычислительного процесса на уровне работы отдельных программных модулей, фиксация состояния различных очередей сообщений и деление сообщений в ходе моделирования их передачи на сегменты и кадры;
- отражение использования основных компьютерных и коммуникационных ресурсов;
- сбор всех необходимых статистических данных для оценки эффективности, выявления узких мест и резервов функционирования системы;
- обладание легко изменяемой структурой на основе графического интерфейса;
- обеспечение простого способа задания исходных данных и удобного представления результатов моделирования.

Для реализации модели выбрана имитационная объектно-ориентированная система моделирования OMNET++ [2]. Она относится к категории некоммерческих продуктов и имеет следующие достоинства: открытость, модульность, возможность построения многоуровневых моделей, графический интерфейс пользователя, удобство отладки и высокую производительность. Отладка включает возможности трассировки, анимации и наблюдения за списком будущих событий, что особенно важно, учитывая высокую алгоритмическую сложность создаваемой модели.

В системе OMNET++ отсутствует библиотека готовых моделей для компонентов информационно-вычислительных систем, поэтому необходима их самостоятельная разработка. Программирование выполняют на языке C++.

С учетом иерархического принципа построения имитационной модели выделены три основных уровня её блоков:

- элементарные (базовые) блоки модели, представляющие программные процессы, устройства, очереди, память и др.;
- укрупнённые блоки, имитирующие работу основных структурных элементов системы, таких как приложение, локальная служба и менеджер очередей;
- общая модель системы, построенная на основе укрупнённых блоков.

Исходные данные для моделирования получены путем натурных измерений для имеющихся версий программ, входящих в состав разрабатываемого комплекса передачи сообщений. В частности, для каждого программного модуля определена зависимость времени его исполнения от размера сообщения. Анализ показал, что в большинстве случаев эти времена пропорциональны размеру сообщения. Исследовано также влияние на указанное время объёма свободной оперативной памяти и процесса свопинга.

После сбора исходных данных проведены модельные эксперименты для различных вариантов сетевых структур системы передачи сообщений. Исследовано влияние важнейших её параметров: размера и приоритета сообщений, интенсивности отправки сообщений, скорости работы каналов связи, производительности процессоров и объема оперативной памяти менеджеров очередей на время доставки сообщений, размеры очередей в различных точках системы и загруженность её ресурсов.

Примеры результатов моделирования показаны на рис. 1 и 2. Здесь представлены характеристики функционирования системы передачи сообщений, включающей две локальные сети, в которых взаимодействуют около десятка рабочих станций через два менеджера очередей. Сети соединены каналом связи, работающим со скоростью 100 Мбит/с.

На рис. 1 дана зависимость времени передачи сообщений от их размера. Видно, что при размерах сообщений менее 100 кб времена их доставки относительно малы и составляют доли секунды.

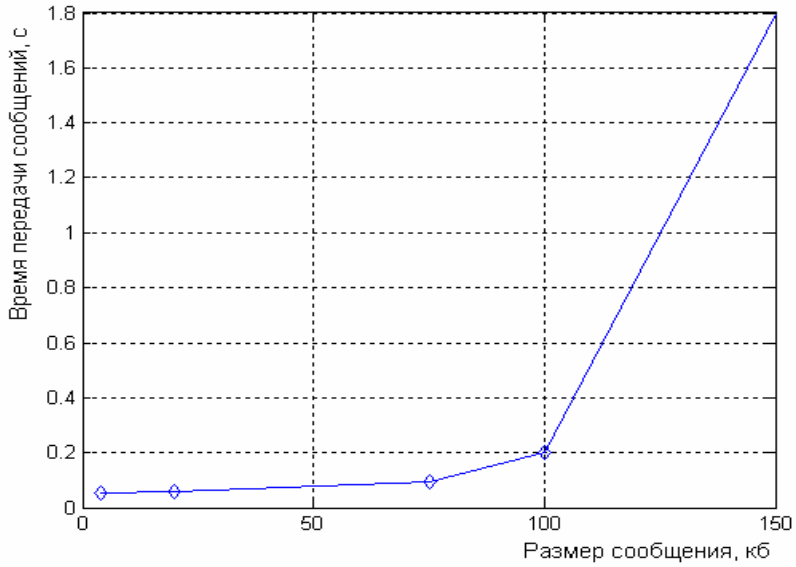


Рис. 1. Зависимость времени передачи сообщений от их размера

С увеличением размера сообщений наблюдается существенный рост времени, что связано с перегрузкой одного из ресурсов системы (канала 3, рис. 2).

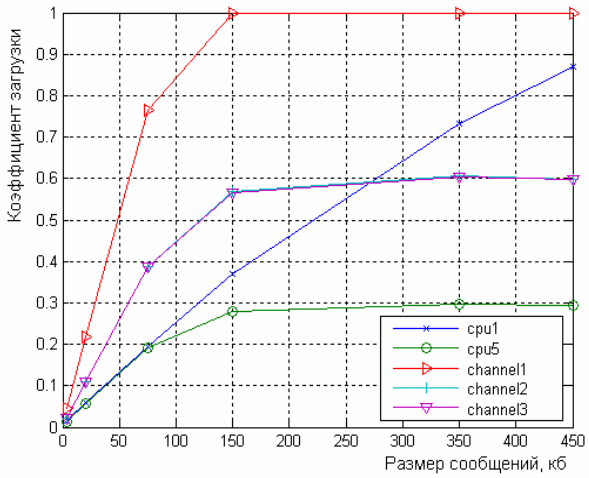


Рис. 2. Зависимость загрузки процессоров и каналов от размера сообщений

В целом, опыт разработки показал, что выбранные инструментальные средства и методика, основанные на широком применении приёмов объектно-ориентированного программирования, дали возможность быстро создать довольно сложную имитационную модель. С её помощью проведено сравнение различных вариантов построения системы передачи сообщений, выявлены узкие места и резервы в её работе. В модель заложены возможность расширения и совершенствования. Готовые её блоки могут быть использованы для построения моделей других сетевых систем.

Список литературы.

1. “Middleware: модель сервисов распределенной системы” Ф. Бернштейн. Журнал "Системы управления базами данных", №2, 1997.

2. Москаленко А. Ш., Окороченко Г. Е. Сравнительный анализ средств для имитационного моделирования вычислительных сетей. Научная сессия МИФИ-2004. Сборник научных трудов. В 15 томах. Т. 10. Телекоммуникации и новые информационные технологии в образовании. М.: МИФИ, 2004, с. 95-96.

Выделение полосовых образов заданного профиля на цифровых изображениях

Орлов А. А., *докторант*
602264, г. Муром, Владимирская обл., ул. Орловская, 23,
МИ ВлГУ, кафедра ИС
e-mail: is@mivlgu.ru

Изображение является основным источником информации для человека. Показано также, что для большинства классов изображений наиболее ценными данными является препарат протяженных линейчатых объектов. Таковыми являются, например, контуры на изображениях. По границам (контурам) человек распознает и анализирует форму объектов на рассматриваемой сцене.

Сообразно с-этим в области компьютерной обработки изображений существует огромное количество работ, связанных с выделением и анализом границ. Обычно задача выделения контуров объектов решается как детектирование перепадов яркости в области изображения на основе градиентных фильтров [1, 2].

Имеются также сцены, на которых непосредственно присутствуют изображения объектов в виде полос. Это изображения рукописей, треков, дорог, ребер на флюорограммах и многое другое. В большинстве случаев яркостные перепады протяженных образов имеют уникальную форму. Поэтому для повышения качества обработки таких изображений возникает необходимость использования специфических признаков.

Целью работы является построение фильтра для детектирования полосовых образов заданного профиля для их последующего анализа.

Разработка математической модели исходного изображения

Пусть $\varphi(x,y) = 0$ – уравнение кривой на плоскости R^2, так что функция $\varphi : R^2 \to R$ возвращает значение, модуль которого численно равен расстоянию до этой кривой.

Определение

Множество точек, расположенных вдоль кривой $\varphi(x,y) = 0$ на расстоянии, не превышающем значение d (рисунок 1),

$$S = \{(x, y) \| \varphi(x, y)| \le d\} \tag{1}$$

назовем областью полосы (или просто полосой), заданной кривой φ (сердцевиной полосы).

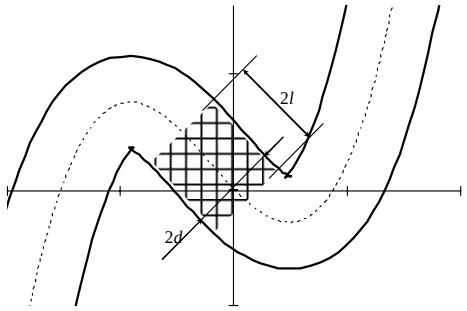


Рис. 1.

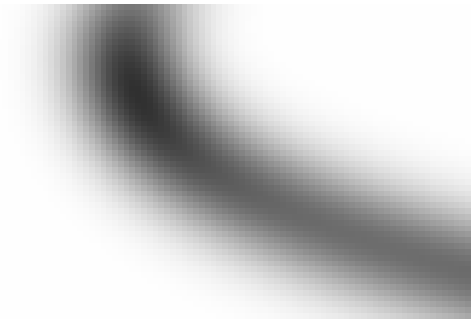


Рис. 2.

Составим модель изображения, содержащего полосу. Путь $\gamma: R^2 \to R$ – функция яркости некоторого изображения в непрерывном пространстве R^2.

Определение: Изображение, заданное функцией $\gamma: R^2 \to R$, будем называть фоном.

Определение: Изображением полосы назовем сцену, заданную характеристической функцией $\chi : R^2 \to \{0, 1\}$,

$$\chi(x, y) = \begin{cases} 1, (x, y) \in S, \\ 0 \end{cases} = \begin{cases} 1, |\varphi(x, y)| \le d, \\ 0. \end{cases} \tag{2}$$

Определение: Профилем полосы в точке (x, y) будем называть сечение поверхности $z = \chi(x, y)$ нормальной плоскостью в пространстве R^3 к кривой $\varphi(x,y) = 0$ в точке (x, y).

Ясно, что профиль изображения полосы (2) будет иметь прямоугольную форму.

Обобщим понятие полосы, видоизменив ее профиль. Для этого будем полагать, что полоса состоит (точнее, интегрируется) из бесконечного множества ее профилей расположенных вдоль кривой $\varphi(x,y) = 0$.

Пусть $A(x)$ – профиль полосы, так что $\lim_{x\to\pm\infty} A(x) = 0$. Тогда характеристическая функция изображения полосы профиля $A(x)$ определится как

$$\chi(x_0, y_0) = \int_{\varphi(x,y)=0} A\left(\sqrt{(x-x_0)^2 + (y-y_0)^2}\right) ds$$

Пример изображения полосы, где $A(x)$ - гауссиан, приведен на рисунке 2.

В качестве модели исходного изображения возьмем алгебраическую сумму

$$f(x,y) = \gamma(x,y) + k\chi(x,y) \qquad (3)$$

как наложение изображения полосы на фон, где k – коэффициент, характеризующий контраст полосы на фоне.

Вычисление признаков полосы

Решим задачу вычисления параметров полосы, заданной функцией $f : R^2 \to R$. Для этого разобьем полосу на сегменты.

Определение. Под сегментом полосы в точке (x,y) $((x,y)\in\varphi)$ будем понимать область полосы S, ограниченной прямыми, полученными параллельным переносом нормали к кривой φ в точке (x,y) на расстояние l (рисунок 1).

Определение. Значение $2l$ назовем длинной сегмента.

Таким образом, каждый сегмент полосы характеризуется шириной $2d$, длиной $2l$, позицией (x,y) и ориентацией θ.

Аппроксимируем сегмент полосы произведением функции детектора профиля полосы и гауссоиды:

$$s(x,y) = B(x)e^{-\frac{y^2}{2\sigma^2}}. \qquad (4)$$

Функция $B(x)$ несколько отличается от $A(x)$ и должна учитывать наличие множества полос и других объектов на исходном изображении. $B(x)$ следует формировать статистически по выборке профилей реальных образов.

Определение. Функцию $s(x,y)$ будем называть детектором сегмента полосы (или просто детектором полосы).

Введем соответствующую функцию комплексного переменного $s(r) = s(x+iy) = s(x,y)$, на основе которой выполним вращение поверхности $s(x,y)$ относительно оси аппликат на угол θ, умножив комплексный аргумент на $e^{i\theta}$:

$$s(x,y,\theta) = \underset{\theta}{\mathrm{Rot}}\, s(x,y) = s(r\cdot e^{i\theta}). \qquad (5)$$

где $\underset{\theta}{\mathrm{Rot}}$ – оператор вращения вокруг оси Oz на угол θ.

Сопоставим параметр σ с параметром l (т.е. возьмем $l = \sigma$).

В качестве сравнительной оценки исходного изображения в точке (x_0,y_0) с детектором полосы (4) выполним скалярное произведение (свертку)

$$h(\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-x_0, y-y_0)s(r\cdot e^{i\theta})dxdy. \qquad (6)$$

Определение. Формулу (6) назовем интегральным преобразованием по сегменту полосы.

Вследствие корреляционных свойств преобразования (6) спектральная функция $h(\theta)$ будет принимать максимальное значение в точке θ_0, тогда и только тогда, когда поверхность функции исходного изображения будет наилучшим образом повторять поверхность детектора.

Таким образом, для определения θ_0 достаточно решить следующую задачу оптимизации:

$$h(\theta) \to \max,$$
$$0 \le \theta < 2\pi. \qquad (7)$$

Разработка алгоритма

Алгоритм вычисления признаков полос будет заключаться в выполнении преобразования (6) в каждой точке изображения:

$$h(x_0, y_0, \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x - x_0, y - y_0)s(r \cdot e^{i\theta})dxdy$$

и определения θ_0 по максимуму $h(x_0,y_0,\theta)$.

При реализации алгоритма в дискретном пространстве можно помещать знания функции $s(x, y)$ в матрицы порядка $(3\sigma + 1)\times(3\sigma + 1)$. Размеры матрицы обусловлены правилом трех сигм. Однако преобразование (6) является довольно трудоемкой операцией. Поэтому разложим двойной интеграл (6) на два интеграла по перпендикулярным прямым:

$$g(x_0, y_0, \theta) = \int_{-\infty}^{\infty} f(t\sin\theta + x_0, t\cos\theta + y_0)e^{-\frac{t^2}{2\sigma^2}}dt,$$

$$h(x_0, y_0, \theta) = \int_{-\infty}^{\infty} g(-t\cos\theta + x_0, t\sin\theta + y_0)B(t)dt$$

Отсюда алгоритм выполнения преобразования примет следующий вид.

Пусть $\Gamma(y) = e^{-y^2/2\sigma^2}$, r – размер фильтра ($r = 3\sigma$).

Первый этап (формирование значений промежуточной функции $g(x,y,\theta)$)

Для $\forall\ \theta \in [\theta_{\max}, \theta_{\min}]$
  Для $\forall\ t \in [-r, r]$
    $x(t) = t\sin\theta$
    $y(t) = t\cos\theta$
  Для $\forall\ x_0 \in [r, x_{\max} - r)$
    $x(t) = x(t) + x_0\ \forall\ t \in [-r, r]$
    Для $\forall\ y_0 \in [r, y_{\max} - r)$
      $y(t) = y(t) + y_0\ \forall\ t \in [-r, r]$
      $S = 0$
      Для $\forall\ t \in [-r, r]$
        $S = S + f(x(t),y(t))\ \Gamma(t)$
      $g(x_0,y_0,\theta) = S$

Второй этап

Для $\forall\ \theta \in [\theta_{\max}, \theta_{\min}]$
  Для $\forall\ t \in [-r, r]$
    $x(t) = -t\cos\theta$
    $y(t) = t\sin\theta$
  Для $\forall\ x_0 \in [r, x_{\max} - r)$
    $x(t) = x(t) + x_0\ \forall\ t \in [-r, r]$
    Для $\forall\ y_0 \in [r, y_{\max} - r)$
      $y(t) = y(t) + y_0\ \forall\ t \in [-r, r]$
      $S = 0$
      Для $\forall\ t \in [-r, r]$
        $S = S + g(x(t),y(t),\theta)\ B(t)$
      $h(x_0,y_0,\theta) = S$

Экспериментальные результаты

На рисунке 3 представлены исходные изображения рукописи и флюорограммы, результаты их фильтрации, а также профили полос. Видно, что обработка обеспечивает подавление дефектных

областей (пятен) и устраняет разрывы в тексте рукописи, вызванные аддитивным шумом. На флюорограмме же довольно четко выделяются изображения ребер.

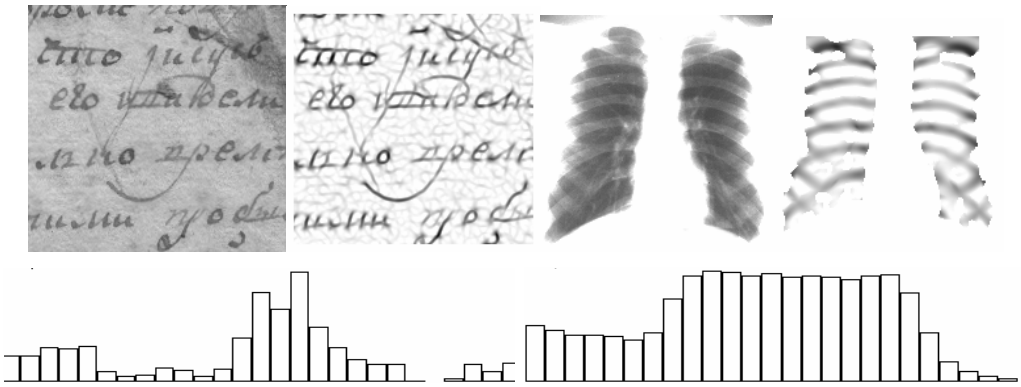


Рис. 3.

В заключение отметим, что разработанный алгоритм обеспечивает высокое быстродействие, а построенная теория интегрирования вдоль полос может служить основой для сознания более сложных и качественных алгоритмов обработки и анализа изображений.

Список литературы

1. Бакут П. А., Колмогоров Г. С. Сегментация изображений: Методы вы-деления границ областей. Зарубежная радиоэлектроника, №10, 1987, с. 25-47

2. Методы компьютерной обработки изображений. Под ред. В. А. Сойфера. – М.: ФИЗМАТЛИТ, 2003. – 784 с.

Компьютерная модель идентификации портрета Франциска Скорины

Павловская А. И. [1] **, Ревяко Г. М.** [2] **, Фисюк М. М.** [3]

1) Аспирантка кафедры ЭВМ Белорусского государственного университета информатики и радиоэлектроники, ул. П. Бровки,6, г. Минск, 220027, Беларусь,
pavlovskaya_a@mail.ru

2) Доцент кафедры ЭВМ Белорусского государственного университета информатики и радиоэлектроники, ул. П. Бровки,6, г. Минск, 220027, Беларусь,
reviako-rti@bsuir.unibel.by

3) Зам. Начальника кафедры криминалистической экспертизы Академии МВД РБ, пр. Машерова, 6, г. Минск, 220027, Беларусь

Введение

В основе построения компьютерной модели идентификации лежит проведение портретной экспертизы двух изображений великого белорусского просветителя и первопечатника Франциска Скорины [1,2].

Приведем фрагмент фрески Рафаэля Санти «Афинская школа», украшающей библиотеку Ватикана. Исследование, описанное ниже, покажет, что человек, изображенный рядом с Рафаэлем, - Франциск Скорина!

Рис. 1 – Фрагмент фрески «Афинская школа»

Прежде всего, для этого есть исторические предпосылки. Доктор искусств Скорина, прекрасно знающий итальянскую живопись, как раз в то время жил в Италии, в Падуе защищал еще одну докторскую диссертацию – по медицине. Хотя история не оставила свидетельств, что пути их действительно пересекались, но, учитывая узость круга художников и искусствоведов, можно предположить, что они знали друг друга. Искусствоведы всегда консультировали заказчиков, к какому художнику обратиться, и мастера кисти жизненно были заинтересованы в таком знакомстве.

Что касается росписи ватиканской библиотеки, то Папа Римский, как отмечали современники, проявил тогда измотавшую всех его помощников чрезмерную требовательность. Он долгое время отвергал предлагавшихся ему художников и их творческие замыслы, пока не сделал выбор в пользу Рафаэля. Естественно, люди из окружения Папы нашли молодого тогда мастера не сами, а по подсказке кого-то разбирающегося в живописи. Вероятно, это был Франциск Скорина. Оказанная услуга в получении престижного заказа, с которого и началось восхождение Рафаэля к вершине славы, способна была подвигнуть художника на такой жест, как увековечивание своего благодетеля.

На эту мысль наталкивает и другое обстоятельство. В роли Сократа, Птолемея, Гераклита и других исторических личностей на фресках представлены легко узнаваемые современники Рафаэля – итальянские общественные деятели, ученые, представители правящей элиты. И только портрет человека, которого художник расположил рядом с собой, нарушив технологию, то есть написав его вживую на штукатурке, не используя утверждаемый заказчиком картонный эскиз, так и остался не идентифицированным.

Необходимо отметить, что профессиональные художники дают положительное заключение о том, что человек на фреске и Франциск Скорина – одно и то же лицо.

Метод сопоставления относительных величин

Для проведения идентификации портрета Франциска Скорины использовался метод сопоставления относительных величин [3]. Исследования проводились совместно со специалистом кафедры криминалистики Академии МВД Республики Беларусь.

При производстве криминалистической портретной экспертизы могут быть использованы количественные признаки.

Для выделения количественных признаков необходимо:

- проставить основные антропометрические точки на изготовленных репродукциях;
- провести оси координат;
- измерить линейные размеры частей лица;
- вычислить относительные размеры;
- оценить ошибки результатов измерении.

Для измерения проекций частей лица необходимо определить точное место антропометрических точек в соответствии с их описанием. В любом случае для измерений выбираются отчетливые, достоверно выявляемые на фотоснимках границы элементов лица, такие участки контуров, где образующие их линии сходятся в одной хорошо просматриваемой точке и резко изменяют направление, например углы глаз, рта. Крайние точки овальных элементов внешности (например, ушной раковины, кончика и крыльев носа и др.) определяются с помощью касательных линий, проведенных вертикально или горизонтально по отношению к контуру исследуемого элемента внешности. Высвеченные, затемненные или закрытые частями лица точки на изображении не отмечаются. Проставленные точки по размеру должны быть как можно меньше.

Для измерения проекций тех или иных частей лица на репродукции снимка проводят оси координат.

При измерениях изображений элементов внешности следует соблюдать ряд требований. Измеряться должны расстояния между отчетливо просматриваемыми точками, чтобы исключалась возможность получения разных результатов при многократных измерениях.

Расстояния между антропометрическими точками измеряются только по горизонтали (оси x и z) или по вертикали (ось y) с точностью ±0,5 мм. Линейные размеры по оси абсцисс характеризуют ширину лица, по оси аппликат – их протяженность в глубину или выступание вперед, а по оси ординат – их высоту. При обозначении вертикальных и горизонтальных проекций лица указываются соответствующие оси координат (x,y,*z*), в скобках обозначаются индексы антропометрических точек, ограничивающих исследуемую часть лица. Например, y(1-2) означает проекцию высоты лба (от верхнелобной точки до верхненосовой точки) на вертикальную ось ординат (ось y).

Чтобы исключить влияние возможных различий масштаба изображения на сравниваемых снимках, в качестве количественных признаков выделяются не абсолютные, а относительные размеры элементов лица. Для этого размер одной (меньшей) части лица делится на размер другой (большей) части лица.

Относительные размеры определяются с точностью 0,001.

Методом сопоставления относительных величин подтверждается совпадение внешности лиц, изображенных на Рис. 2, так как при сопоставлении относительных величин установлена их разница менее чем на 0,05, что не превышает пределы, допустимые для пропорции одного и того же лица, учитывая неудовлетворительное качество изображений (см. таблицу-разработку №1).

Рис.2 Фрагменты автопортрета Ф. Скорины и фрески «Афинская школа»

При сопоставлении относительных величин в качестве размерных характеристик использовались расстояния между следующими антропометрическими точками:

Таблица №1 - *Разработка метода сопоставления относительных величин.*

Наименование размерных арактеристик	*Размеры абсолютных величин, мм*		*Разница относительных величин*
	На черно-белом изображении	*На цветном изображении*	
$x(4-4_1)$	19	18	-
$x(6_1-6_2)$	27	25,5	-
$x(4-4_1)/ x(6_1-6_2)$	0,703	0,705	0,002
$x(11-11_1)$	34,5	34,5	-
$x(0-0_1)$	39	38	-
$x(11-11_1)/ x(0-0_1)$	0,884	0,907	0,02
$x(8_1-8_2)$	8,5	7	-
$x(4_2-4_3)$	37	35	-
$x(8_1-8_2)/ x(4_2-4_3)$	0,229	0,200	0,02
$x(Д_{21}-Д_2)$	28	28	-
$x(Д_3-Д_4)$	7	7,5	-
$x(Д_3-Д_4)/ x(Д_{21}-Д_2)$	0,250	0,267	0,01
y (2-14)	90	92	-
y (2-6)	50,5	47,5	-
y (2-6)/ y (2-14)	0,561	0,516	0,04
$y(5_2-10)$	58	57	-
y (8-10)	11	10	-
y (8-10)/ $y(5_2-10)$	0,189	0,175	0,01
y (6-14)	43	40,5	-
y (6-8)	11	10	-
y (6-8)/ y (6-14)	0,255	0,246	0,009
y (10-14)	20,5	22,5	-
$y(2-5_2)$	12	12,5	-
y (10-14)/ $y(2-5_2)$	0,585	0,555	0,03

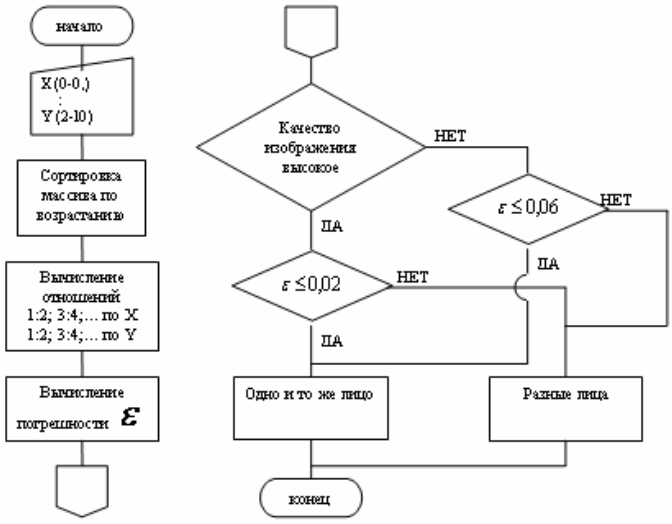


Рисунок 3

Заключение. В результате кодирования приведенного алгоритма создается программное обеспечение, позволяющее проводить портретную экспертизу методом сопоставления относительных величин с помощью компьютера и за короткое время.

Несомненно, важным и значительным является факт подтверждения выдвинутой гипотезы о том, что на известнейшей фреске рядом с великим художником изображен знаменитый просветитель белорусской земли Франциск Скорина!

Список литературы

1. Павловская А. И. Компьютерное моделирование в культурологи на примере Библии Скорины. Труды Международного Форума по проблемам науки, техники и образования, декабрь 5-9, 2005. М.: Акад. Наук о Земле,- 2005,- с.101-102.

2. Касабуцкий А. Н., Павловская А. И., Ревяко Г. М. Алгоритм фильтрации и криптография гравюр Библии Скорины. Труды Восьмой международной конференции «Распознавание образов и обработка информации», май 18-20, 2005.Мн.:PRIP'05,- 2005,-с. 177-179.

3. Зинин А. М., Кирсанова Л. З. Криминалистическая фотопортретная экспертиза. Под ред. В. А. Снеткова, З. И. Кирсанова. – М.:ВНИИ МВД СССР, 1991.

Об одном подходе к построению модели контекста

Парамонов А. И., e-mail: *anton@paramonov.info*
ст. преподаватель, Донецкий национальный университет,
Украина, Донецк, 83055, пр. Театральный, 13

Актуальность проблемы обработки цифровой информации в виде естественно-языковых (ЕЯ) текстов сегодня уже не вызывает сомнений. Информатизация захватывает все виды деятельности человека от обучения до сложных бизнес процессов. Соответственно, все большие объемы информации передаются через цифровые потоки. При решении задач автоматизации тех или иных процессов одним из краеугольных камней становится понимание ЕЯ информации. Под пониманием будем считать умение системы выделить составляющие текста и их отношения (контекст). Существуют подходы к решению данной задачи на основе хорошо разработанных моделей [1-3]. В работе предложен подход к построению модели контекста на основе нечеткой гибридной модели (ГМ) [4]. ГМ включает семантическую сеть для представления знаний об объектах, семантическую сеть для представления знаний о действиях, пропозициональную сеть для представления знаний о событиях. В указанных сетях представлены уникальные знания о прототипах объектов, действий и событий, соответственно. Узлы семантических сетей объектов и действий выступают в качестве элементов узлов пропозициональной сети.

Контекстом будем считать цепочку событий (подмножество узлов пропозициональной сети), субъектами которых выступают «главные действующие объекты». Под «главными действующими объектами» будем понимать множество субъектов (подмножество узлов семантической сети объектов), элементы которого наиболее часто представлены в обрабатываемом тексте.

Задачу выделения контекстных знаний необходимо рассматривать как обязательную составляющую процесса интерпретации текста. Знание основной цепочки событий и «действующих объектов» позволяет в большей мере корректно и адекватно реализовывать понимание и представление текста. Поэтому фрагмент текстовой информации интерпретируется с учетом контекстных знаний. Контекстные знания представляют собой активный подграф ГМ. Обработка фрагмента текста заключается в определении подграфа ГМ (контекста), который выделяется на основе нечеткой активности узлов. Нечеткая активность узла определена как нечеткое множество $\underset{\sim i}{J}$, представленное L-R функциями:

$$\underset{\sim i}{J} = \{x \mid \mu_{\underset{\sim i}{J}}(x)\}, x \in [0,1] \qquad (1)$$

Значение этого множества вычисляется для каждого узла как совокупность нескольких составляющих в зависимости от нескольких факторов. Конечное значение активности узла определяет его наличие в искомом нечетком подграфе ГМ.

В терминах ГМ все знания о входной текстовой информации будут представлены множеством Т:

$$T = \{S_i \mid \underset{\sim i}{J} \cup \varnothing \neq \varnothing\}, \quad (2)$$

где S_i - узел пропозициональной сети (3), $\underset{\sim i}{J}$ - активность узла, определеного в (1).

$$S = \{\underset{\sim}{P}_i^{o}, \underset{\sim}{P}_i^{c}, \underset{\sim}{P}_i^{д}\}, \quad (3)$$

где $\underset{\sim}{P}_i^{\circ}, \underset{\sim}{P}_i^{c}$ – узлы семантической сети объектов (объект и субъект), $\underset{\sim}{P}_i^{д}$ – узел семантической сети действий.

Введенное множество содержит все возможные (все известные системе) знания представленные во входном информационном блоке, в том числе и данные неявно присутствующие в тексте. Однако, по оценкам психологов, для качественного восприятия вновь поступающей и синтеза новой информации необходимо выделять контекстные знания.

Отличие контекстных знаний от множества всех знаний определяется логической законченностью и целостностью фрагмента.

Для выделения контекста разного уровня важности введено понятие α - уровень активности. Активный подграф гибридной модели будет содержать только те узлы, множество α - уровня которых не будет пусто [5].

$$[J_i]_\alpha = \{x \mid \mu_{\underset{\sim i}{J}}(x) \geq \alpha\}, \quad x \in [0,1]; \quad (4)$$

$$[J_i]_\alpha \cup \varnothing \neq \varnothing,$$

где $[J_i]_\alpha$ - множество α - уровня нечеткого множества активности.

Выделенный активный подграф описывает контекстные знания в терминах предлагаемой гибридной модели. Соответственно, чем больше значение α - уровня, тем сжатей получается представление текста и наоборот.

Для использования контекста в задаче интерпретации входного текста определение α-уровня осуществляется в зависимости от модели мира и устанавливается в момент настройки системы.

С учетом формулы (4) множество T запишется как $[J_i]_\alpha, \alpha = 0$;

Предложенная в работе модель контекста строится на основе пропозициональной сети ГМ. В расширенном виде предполагается возможность использования α-уровня и для сетей объектов. Это позволит использовать систему для выявления набора терминов, упоминаемых в тексте, с ограничением по частоте вхождений. Данная функциональность является своего рода аналогом механизма, используемого в системах поиска по ключу. Таким образом, использующие предложенную модель контекста интеллектуальные поисковые системы смогут легко изменять критерии поиска. Однако все вышесказанное вызывает некоторую коллизию в использовании одновременного ограничения и по пропозициям и по объектам. Отсечение объектов изменяет множество узлов событий в контексте и наоборот. Над решением этой задачи сейчас ведется работа.

Представленная модель контекста может быть использована для полноценной категоризации и аннотации текстов. А полученные результаты могут быть использованы для построения интеллектуальной поисковой системы.

Список литературы

1) Шенк Р. «Обработка концептуальной информации» / пер. с англ. Г.В.Сенина – М.: Энергия, 1980. – 360 с., ил.

2) Андерсон Дж. «Когнитивная психология. 5-е изд.» – СПб.: Питер, 2002. – 496с.: ил. – Серия «Мастера психологии»

3) Солсо Р. «Когнитивная психология» – СПб.: Питер, 2002. – 592с.: ил. – (Серия «Мастера психологии»).

4) Парамонов А. И., Каргин А. А. «Гибридная нечеткая модель обработки концептуальной информации» – Сборник трудов V международной конференции «ИАИ-2005» - Киев, с. 234-241

5) Заде Л. «Понятие лингвинистической переменной и его применение к принятию приближенных решений» / пер. с англ. Н.И. Ринго – Издательство «Мир», Москва, 1976.

Использование матриц сравнений в задаче поиска по эталону [1]

Перевалов Д. С.,
м.н.с., Институт Математики и Механики УрО РАН
620219, г. Екатеринбург,
ГСП-384, ул. С.Ковалевской, 16
e-mail: Denis.Perevalov@mail.ru

Многие задачи анализа изображений реальных сцен зачастую содержат в себе задачу поиска и локализации какого-либо объекта на изображении. Для ее решения обычно используется принцип сопоставления с эталоном [1]. В этом случае перед началом работы системе предоставляется одно или несколько эталонных изображений искомого объекта, по которому строится некоторый *критерия сопоставления*. С его помощью на входных изображениях ищутся объекты, изображения которых близки к эталонным.

В данной работе вводится один пиксельный критерий сопоставления на основе так называемых матриц сравнений. Для полутонового изображения $X = (X_1, ..., X_n)^T$, состоящего из n пикселов, $X_i \in [0, 1]$, *матрицей сравнений* будем называть матрицу $M = M(X)$ размером $n \times n$, задаваемую по правилу

$$M_{ij} = \begin{cases} 1 & \text{если } X_i \geq X_j \\ 0 & \text{в противном случае.} \end{cases}$$

Это матрица бинарного отношения "$\geq$" между яркостями пикселов X. Она является представлением изображения в порядковой шкале измерения [2]. В частности, это означает, что M инвариантна к изменениям яркости пикселов изображения с помощью произвольной строго возрастающей функции.

Матрица сравнений является удобным средством для работы со следующей моделью изображения объекта. Зададим эту модель как предикат $P(X)$, который строится по набору непересекающихся множеств пикселов $A_1, \dots A_m$ и двумя наборами индексов $(p_1, \dots, p_K)$, $(q_1, \dots, q_K)$, где $p_k,\ q_k \in \overline{1, m}$:

$$P(X) = \bigwedge_{k=1}^{K} (\forall i \in A_{p_k} \forall j \in A_{q_k} (X_i \geq X_j)).$$

Истинность предиката для изображения X означает, что для всех k область A_{p_k} на X ярче (не темнее) области A_{q_k}.

Утверждение 1. $P(X) = (\mathrm{dist}(X; P) = 0)$, где

$$\mathrm{dist}(X; P) \triangleq \sum_{k=1}^{K} |A_{p_k}| \cdot |A_{q_k}| - \sum_{k=1}^{K} \chi(A_{p_k})^T M(X) \chi(A_{q_k}),$$

$\chi(A)$– *характеристический вектор множества* A.

Таким образом, $\mathrm{dist}(X; P)$ можно считать критерием, позволяющим оценить степень отличия изображения от модели – чем это значение ближе к 0, тем отличие меньше.

Для практического применения данного критерия желательно, чтобы модель изображения объекта строилась автоматически на основании одного или нескольких эталонных изображений этого объекта. Один из возможных способов построения модели дает следующее утверждение.

Утверждение 2. *Пусть множество индексов, задающих предикат P, "транзитивно" в следующем смысле: если для некоторых i и j $q_i = p_j$, то найдется k такое, что $p_k = p_i$ и $q_k = q_j$. Пусть $S(P) = \{X \in [0, 1]^n : P(X)\}$ – множество, состоящее из изображений, удовлетворяющих предикату P, а $M(P)$ – матрица, полученная усреднением по всем матрицам сравнений изображений из $S(P)$:*

[1] Работа поддержана грантом РФФИ №06-01-229.

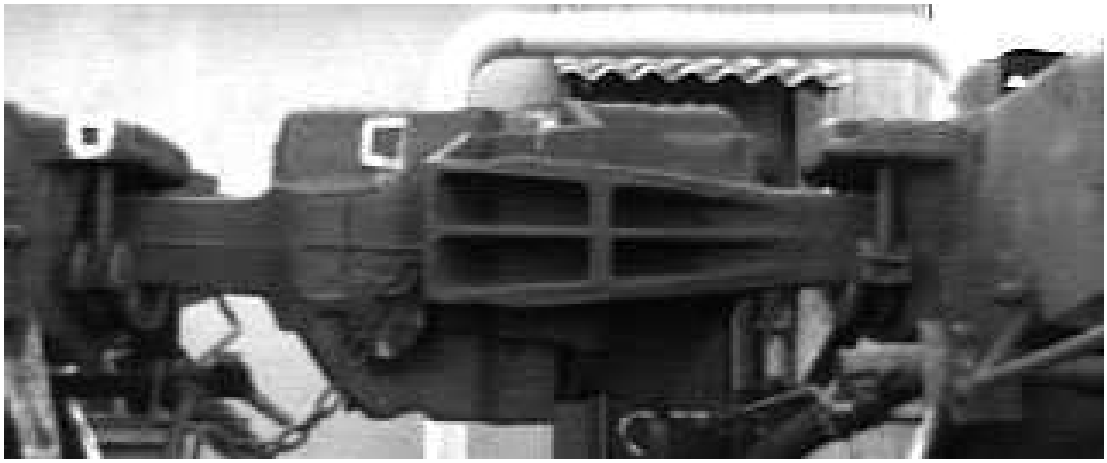


Рис. 1: Пример изображения узла автосцепки.

$$M(P) = \frac{1}{\int_{S(P)} dX} \int_{S(P)} M(X) dX.$$

Тогда для произвольных i и j равенство $M(P)_{ij} = 1$ выполняется в том и только том случае, когда $\exists k$ $(i \in A_{p_k} \wedge j \in A_{q_k})$.

Следовательно, если имеется набор эталонных изображений $T = (T_i, \ldots, T_N)$, то в качестве оценки матрицы $M(P)$ можно взять матрицу $\overline{M} = \frac{1}{N} \sum_{k=1}^{N} M(T_k)$ и выбрать множество пар индексов (i_k, j_k), $k = \overline{1, K}$, для которых $\overline{M}_{i,j}$ принимает наибольшие значения. Будем считать, что $A_{p_k} = \{i_k\}$, $A_{q_k} = \{j_k\}$, а построенный с их помощью предикат P есть оценка модели объекта, изображенного на эталонах. В данном случае выражение для dist(X; P) примет более простой вид: $\mathrm{dist}(X; P) = K - \sum_k M(X)_{i_k j_k}$.

Нужно заметить, что параметр K должен выбираться из практических соображений, в зависимости от требуемых в конкретной задаче качества и быстродействия алгоритма сопоставления с эталоном.

Работоспособность предложенного критерия была проверена путем сравнения с другими широкоизвестными критериями сопоставления в задаче обнаружения изображения автосцепки. На входном изображении требовалось обнаружить автосцепку двух сцепленных железнодорожных вагонов (рис. 1), либо сообщить, что ее на изображении нет. Съемка тестового видео проводилась на сортировочной горке станции Свердловск-сортировочный стационарно закрепленной видеокамерой, ось которой была направлена параллельно осям колесных пар вагонов.

Для тестирования использовалось 685 кадров с изображением 137 автосцепок, при этом каждая автосцепка была представлена пятью кадрами с разных ракурсов; а также 23211 кадров, на которых не было автосцепки. Был выбран набор из $N = 10$ эталонных изображений крестовины автосцепки размером $n = 216 = 24 \times 9$ пикселов.

Использовались следующие критерии сопоставления [1, 3]:

- $\mathrm{dist}_{L_2}(X; T)$ – расстояние от X до ближайшего эталона в метрике L_2;
- $r(X; T)$ – коэффициент корреляции изображения X с усредненным изображением эталонов;
- $\rho(X; T)$ – коэффициент корреляции Спирмена рангового вектора изображения X с усредненным вектором рангов эталонов;
- $\mathrm{dist}(X; P)$ – предложенный критерий с $K = n$.

На каждом тестовом изображении было проведено сканирование локальным окном, в каждой позиции окна было вычислено значение критерия. Экстремальное значение критерия для изображения (минимум для $\mathrm{dist}_{L_2}(X; T)$ и $\mathrm{dist}(X; P)$; максимум для $r(X; T)$ и $\rho(X; T)$) было записано в таблицы. На основании таблиц для всевозможных пороговых значений были вычислены частоты

Рис. 2: ROC-кривые.

верной детекции и ложной тревоги. Результаты, отражающие эти данные, представлены на рис. 2 в виде ROC-кривых (Receiver Operating Characteristic Curves) [4].

Чем ближе график такой кривой к верхней горизонтали $y = 1$, тем алгоритм точнее, а площадь под кривой есть численная мера точности. В данном случае площади оказались равными 0.3423, 0.8491, 0.9793 и 0.9894 для критериев dist_{L_2}, r, ρ и dist соответственно. Таким образом, предложенный критерий dist показал лучший по качеству результат, сопоставимый по качеству с коэффициентом корреляции Спирмена ρ. В то же время скорость вычисления dist превосходит скорость вычисления ρ примерно в $\log_2 n \approx 8$ раз. Это объясняется тем, что время их работы есть $O(n)$ и $O(n \log_2 n)$ операций соответственно.

Проведенный эксперимент является свидетельством в пользу эффективности использования аппарата матриц сравнений для построения моделей и алгоритмов в ряде задач обработки изображений.

Список литературы

[1]. Гонсалес Р., Вудс Р. *Цифровая обработка изображений.* : Пер. с англ. – М.: Техносфера, 2005. – 1072 с.

[2]. Рунион Р. *Справочник по непараметрической статистике.* : Пер. с англ. – М.: Финансы и статистика, 1982. – 198 с.

[3]. Баклицкий В.К., Бочкарев А.М., Мусьяков М.П. *Методы фильтрации сигналов в корреляционно-экстремальных системах навигации.* – М.: Радио и связь, 1986. – 216 с.

[4]. Courtney P., Thacker N. A. *Performance Characterisation in Computer Vision: The Role of Statistics in Testing and Design*– In "Imaging and Vision Systems: Theory, Assessment and Applications", NOVA Science Books, 2001.

Принятие решений методом анализа иерархий: как сократить экспертоемкость?

Плаксин Михаил Александрович
Доцент каф. математического обеспечения вычислительных систем
Пермского государственного университета
614008 г.Пермь ул.Ленина д.102 кв.71.
plaksin@psu.ru

Метод анализа иерархий (МАИ) – метод принятия стратегических решений, разрабатываемый в последние три десятилетия Т. Саати [1,2]. Он предназначен для выбора оптимального решения проблемы в том случае, когда это решение должно удовлетворять нескольким (противоречивым) критериям и отсутствуют объективные показатели для сопоставления альтернатив.

Суть МАИ заключается в двух моментах: в постепенности построения оценки и в использовании специальной «мягкой» (качественной) шкалы сравнений. Многошаговость метода заключается в том, что МАИ не пытается разом оценить приоритетность того или иного решения с точки зрения поставленной проблемы. Сначала попарно сравниваются между собой критерии. Эксперт должен ответить на вопрос: какой из каждой пары критериев важнее с точки зрения решаемой проблемы и насколько важнее. По полученной матрице парных сравнений (МПС) определяются сравнительные веса (приоритеты) критериев. На следующем шаге выполняется сопоставление альтернативных решений. Но сопоставляются они не «вообще», а относительно конкретного критерия. Эксперт должен ответить на вопрос: какое из двух решений предпочтительней с точки зрения конкретного критерия и насколько предпочтительней. По полученной МПС определяются локальные приоритеты (веса) решений относительно каждого из критериев. После этого на основе весов критериев и локальных весов решений относительно критериев вычисляются глобальны веса решений.

Описанная схема легко распространяется на многоуровневую иерархию. Так для задач прогнозирования стандартной является иерархия в 6-7 уровней (проблема, первичные факторы (экономические, технологические и пр.), акторы (действующие лица, влияющие на решение проблемы), цели акторов, политики (способы действия) акторов, контрастные сценарии развития событий, обобщенный сценарий).

В случае невозможности «жесткого» количественного сравнения применяется качественная шкала из значений типа «равная важность», «слабое превосходство», «сильное превосходство» и т.п.

Одним из главных недостатков метода является его чрезвычайно высокая «экспертоемкость». Она связана с двумя моментами: необходимостью большого числа парных сравнений и несогласованностью (противоречивостью) экспертных оценок.

В качестве иллюстрации потребности в большом числе парных сравнений рассмотрим вышеупомянутую иерархию для решения задач прогнозирования. Саати позволяет каждому элементу иерархии иметь до 15 потомков. Пусть их будет только 7. Тогда анализ иерархии потребует заполнить 113 МПС размером 7x7: одна матрица для сравнения факторов, 7 – для оценки акторов относительно каждого из факторов, 7 – для сопоставления целей каждого из семи акторов, 49 МПС – для сравнения политик акторов по отношению к их целям (у семи акторов целей 49). 49 МПС – для оценки контрастных сценариев относительно 49 политик, которые могут быть реализованы акторами.. При увеличении количества потомков до 10 количество МПС возрастает до 221, а размер матрицы – до 10x10.

Для снижения экспертоемкости метода были предложены следующие механизмы:

1) построение МПС, согласованных «по построению»;
2) уменьшение числа сравнений за счет исключения из иерархии части элементов;
3) ускорение поиска рассогласованностей в матрице парных сравнений, выбор элементов, согласование которых даст максимальный эффект.

Для построения МПС, согласованных «по построению», разработана специальная «таблица сложения сравнений» (ТСС).

Применение ТСС базируется на следующем факте. Рассмотрим тройку альтернатив a_i, a_j, a_k.с весами p_i, p_j, p_k. Элементы МПС, соответствующие сравнению этих альтернатив друг с другом, будут равны $\frac{p_i}{p_j}$, $\frac{p_i}{p_k}$, $\frac{p_j}{p_k}$. (в числителе – вес «элемента-строки», в знаменателе – вес «элемента-столбца»). Тогда после заполнения экспертом любых 2-х элементов МПС возникает желание третий вычислить автоматически. Такому автоматическому вычислению препятствует нечеткость шкалы сравнений. Традиционная таблица умножения в данном случае не годится. Например, если a_i имеет очень слабое превосходство над

a_j, a_j – среднее превосходство над a_k, то как оценить отношение a_i и a_k? Для ответов на подобные вопросы была разработана специальная таблица «сложения сравнений». Таблица составлена в терминах качественной шкалы и предлагает для каждой пары «складываемых сравнений» диапазон возможных вариантов (очень слабое превосходство + очень слабое превосходство = от очень слабого до среднего превосходства, слабое превосходство + средне превосходство = от средне-сильного до сильного превосходства и т.д.). Качество ТСС было проверено при автоматической генерации около 600 тыс. иерархий. Все полученные с ее помощью МПС имели хорошую степень согласованности (до 10%). (По Саати степень согласованности до 10% считается хорошей, до 20% – терпимой, свыше 20% – неприемлемой.)

При ручном заполнении МПС таблица сложения сравнений может использоваться для оперативного контроля действия эксперта. Начиная со 2-й строки, вводимые экспертом элементы МПС могут с помощью ТСС поверяться на соответствие ранее введенным значениям. Таким образом можно выявить несогласованность МПС непосредственно во время построения. При этом необходимо иметь в виду два момента. Во-первых, несоответствие «нижних» строк «верхним» может означать неточность как «внизу», так и «наверху». Нельзя считать «верхние» элементы «более правильными» только потому, что они были записаны ранее «нижних». Во-вторых, значения из ТСС являются не более, чем рекомендацией, которую эксперт волен принимать или не принимать.

Еще одним свойством МПС, которое хочется использовать для уменьшения нагрузки на эксперта при заполнении матрицы, является ее обратная симметричнсть (элементы, симметричные относительно главной диагонали, должны быть обратны: над диагональю – $\frac{p_i}{p_j}$, значит под диагональю – $\frac{p_j}{pi}$). Кажется естественным использовать эксперта для заполнения только одной половины матрицы, а вторую получить автоматически. Делать этого ни в коем случае нельзя! Как показывает опыт, при заполнении полной матрицы, называемые экспертом оценки далеко не всегда являются обратно симметричными. Более того, иногда оценка меняется не только количественно, но и качественно (при начальном сравнении $a_i > a_j$, при повторном – $a_j > a_i$). Поэтому обратную симметричность можно и должно использовать, но не для автоматизации заполнения МПС, а для контроля принимаемых экспертом решений.

Второй механизм – исключение из иерархии некоторых элементов – основывается на следующих рассуждениях. Качество вырабатываемых рекомендаций зависит (как минимум) от трех факторов: качества иерархии, квалификации эксперта и качества заполнения экспертом матриц парных сравнений. Увеличивая детальность иерархии, мы можем увеличить качество иерархии. Но при этом стремительно возрастает количество и размер МПС (пример был приведен выше). Соответственно, возрастает нагрузка на эксперта, их заполняющего. Для качественного выполнения работы требуется высококвалифицированный специалист. Время таких работников – дорого. Поэтому увеличение нагрузки ведет к сокращению внимания, уделяемого каждому конкретному вопросу, ведет к снижению качества заполнения матриц. В результате может оказаться, что увеличивая детализацию иерархии, мы не повышаем, а понижаем качество модели.

Возникает вопрос: существует ли способ определить, какая степень детализации иерархии является наилучшей? Ответ будет содержать в себе противоречие: и нет, и да. Нет – потому что это невозможно сделать статически, в момент построения иерархии. Да – потому что это оказывается можно сделать динамически во время анализа иерархии, поэтапного продвижения от ее корневой вершины к концевым.

В ходе построения иерархии мы не имеем еще никакой информации о значимости того или иного ее элемента. Поэтому в это время у нас нет никаких оснований принимать решение о том, следует ли включать тот или иной элемент в иерархию или без него можно обойтись. Поэтому чем более подробная иерархия будет построена, тем лучше. Иное дело – динамика. Процесс анализа иерархии заключается в том, что мы двигаемся сверху вниз, заполняем матрицы парных сравнений и вычисляем сначала локальные, а потом и глобальные веса элементов. При этом элементы естественным образом кластеризуются (например, все акторы, все цели одного актора, все политики одного актора). Появляется возможность в каждом кластере отделить весомые, значимые элементы от незначимых. И незначимые – отбросить. Например, пусть первоначально мы включили в иерархию 10 акторов. После вычисления их глобальных весов выяснилось, что решение проблемы на 80% зависит только от четырех акторов (вес каждого – 15-25%), а суммарный вклад остальных шести акторов – 20% (вес каждого 3-4%). Возникает вопрос: стоит ли продолжать анализ этих шести акторов, если от них почти ничего не зависит?

Такое динамическое прореживание иерархии позволяет значительно сократить трудоемкость анализа. Без него анализ крупных иерархий становится попросту невозможен.

Прореживание иерархии приводит к вопросу о корректности этого действия, о том, какое влияние оно окажет на конечный результат. Как скажется исключение того или иного элемента на весах контрастных сценариев, т.е. на тех рекомендациях, которые будут выработаны в результате применения МАИ? В приведенном нами примере картина была достаточно ясна, разрыв между лидерами и аутсайдерами достаточно велик как по их индивидуальным весам (15-20% против 3-4), так и по суммарным (80%

против 20). В практических задачах картина не всегда настолько контрастна. Отсюда – два вопроса, ответы на которые должны определять допустимость прореживания. Как скажется на конечном результате отбрасывание элемента с тем или иным весом? (Элементы какого веса еще можно отбросить, а какого – уже нельзя?) Как скажется на конечном результате отбрасывание группы элементов с тем или иным суммарным весом? (Каков может быть суммарный вес отбрасываемых элементов?)

К сожалению, точных ответов на эти вопросы пока не найдено. По полученным оценкам в случае отбрасывания акторов с суммарным весом в 10% вероятность смены лидера в списке контрастных сценариев составляет около 5%. При отбрасывании 20% эта вероятность возрастает до 10%. При отбрасывании 30% – до 15%. При отбрасывании 40% – до 20%.

Поскольку оценка влияния прореживания на конечный результат неточна, было предложено заменить прореживание более мягкой операцией – сверткой. Заключается она в том, что часть наименее значимых элементов кластера исключается из рассмотрения, но «не совсем». Вместо них в кластер включается новый элемент с именем «И другие» и с весом, равным суммарному весу исключенных элементов. Таким образом, исключенные элементы продолжают влиять на принятие решения. Но поскольку все они свернуты в один единственный элемент, их дальнейший анализ уже не требует трудозатрат. (Свернув акторов-аутсайдеров, мы избавляемся о необходимости анализировать их цели и политики.) В том случае, если элемент «Другие» должен появиться в МПС, во всех позициях строки и столбца матрицы, относящихся к этому элементу, ставится одно и то же «нейтральное» значение, не влияющее на веса остальных элементов (1/n, где n – количество сравниваемых альтернатив).

Как показывает опыт, замена прореживания на свертку всегда приводит к некоторому «сглаживанию» результатов. Разница между весами контрастных сценариев становится менее ощутимой. Примерно в 10% случаев прореживание и свертка приводят к существенно разным результатам: к тому, что у контрастных сценариев не просто меняются веса, а в списке контрастных сценариев происходит смена лидера.

Третий механизм сокращения экспертоемкости связан с устранением несогласованностей в МПС. Несогласованность выражается в нарушении транзитивности отношения превосходства (если A>B и B>C, то должно быть A>C). Нарушение это может носить характер качественный (A>B, B>C, C>A; это возможно, поскольку альтернативы сравниваются попарно), а может – количественный. Количественная несогласованность связана с тем, что мы не просто говорим о превосходстве той или иной альтернативы, но пытаемся оценить это превосходство по некоторой шкале (слабое/среднее/сильное). Отсюда возможна ситуация: A>>B, B>>C, A>C. Дело осложняется тем, что при сравнении необходимо оперировать качественными понятиями (сильный-слабый).

Причина несогласованности матрицы заключается в следующем. Вообще говоря, содержимое МПС определяется n числами – «истинными весами» альтернатив. В идеале матрица должна состоять из отношений этих n чисел. Но поскольку истинных весов мы не знаем, мы вынуждены выбрать для матрицы $(n^2-n)/2$ различных значений (размер матрицы – $n \times n$, но на главной диагонали стоят единицы, а левый нижний треугольник заполнен величинами, обратными по отношению к величинам из правого верхнего треугольника), которые будут служить некоторыми приближениями для отношений истинных весов. При построении матрицы парных сравнений обеспечивается ее обратная симметричность, но в матрице, построенной из отношений истинных весов, связи между элементами гораздо более многообразны, поскольку вес каждой альтернативы входит в (2n-1) элементов матрицы (все элементы одной строки и одного столбца).

В МАИ существует простая процедура, которая позволяет легко определить «степень согласованности» МПС. Даются рекомендации о том, какую согласованность считать хорошей, какую – приемлемой. К сожалению, в классических работах Т.Саати [1,2] ничего не говорится о том, как определить место несогласованности МПС и устранить эту несогласованность. Рекомендации по этому поводу можно найти в [3, 4].

Список литературы

1. Саати Т., Кернс К. Аналитическое планирование. Организация систем. М.: “Радио и связь”, 1991. - 224 с.

2. Саати Т. Принятие решений. Метод анализа иерархий. М.: “Радио и связь”, 1993. - 320 с.

3. Митягин А. В., Плаксин М. А. Некоторые усовершенствования метода анализа иерархий. //Математика программных систем: Межвуз. сб. научн. Трудов / Перм. ун-т. – Пермь, 2001. С.56-66.

4. Плаксин М. А. Некоторые трудности применения метода анализа иерархий и пути их преодоления. //Первая международная конференция «Системный анализ и информационные технологии» САИТ-2005 (12-16 сентября 2005 г., Переславль-Залесский, Россия): Труды конференции. В 2 т. Т.1. – М.: КомКнига, 2005. С.291-297

Онтологический анализ как средство интеллектуализации поисковых механизмов на рынке информационных услуг

Плескач В. Л.,
к.т.н., доцент кафедры экономической кибернетики КНУДТ,
заведующий отделения информационно-аналитического обеспечения и экономико-математического моделирования Научно-исследовательского финансового института при Министерстве финансов Украины,
02222, Киев, вул. Закревського 49/1, кв.198,
тел.(044) 515-08-0б, e-mail: v_pleskach@ukr.net
Рогушина Ю. В.,
к.ф.-м.н., с.н.с. Институт программных систем НАН Украины,
03150, Киев, вул. Федорова, 9, кв.20,
тел.(044) 268-06-28, e-mail: jjj_@ukr.net

Наиболее важными и перспективными инновациями в области Интернета сейчас считаются Web-сервисы и семантические приложения. Они призваны стать, с одной стороны, очередным большим шагом в развитии отрасли ИКТ, а с другой – составной частью электронного бизнеса, дистанционного образования и государственного электронного управления.

Сегодня ответственность за создание принципиально новых возможностей ИКТ впервые на протяжении всей истории развития отрасли полностью ложится на потребителей новых технологий, что означает переход от промышленной экономики к массово-персонифицированной экономике знаний. В своем развитии современные ИКТ как отрасль столкнулись с целым рядом неожиданных проблем и сложностей. Прежде всего, создание адекватного информационного наполнения (контента) оказалось делом значительно более трудным, чем в случае с предшествовавшими коммуникационными технологиями – радио и телевидением. Переход от традиционных приложений и Web-сервисов к интеллектуальным требует формирования интероперабельных баз знаний (БЗ) для различных предметных областей; именно в этой сфере необходимы новые методы обработки и анализа информации, обеспечивающие извлечение знаний пользователей.

База знаний содержит описание предметной области и представляет концептуальный уровень информации в системе, необходимый для интеграции источников данных и организации содержательного поиска в них.

Электронный бизнес оказался на практике значительно более сложным, трудоемким и к тому же требующим значительных инвестиций и интеллектуального капитала, чем это казалось в начале его развития. Радикальные преобразования в сферах банковских технологий, дистанционного образования, страхового дела, розничной торговли и медицинских услуг оказались трудоемким процессом, который, тем не менее, все наращивает темпы своего роста.

Рынок информационных услуг (электронный рынок) – это система экономических отношений в виртуальном пространстве между субъектами экономической деятельности в процессе торговли услугами и товарами на базе ИКТ. Перспективы развития электронного рынка связаны с его интеллектуализацией и совершенствованием его организационно-технологической инфраструктуры и развитием электронной экономической деятельности.

Современной тенденцией в создании интеллектуальных систем является реализация парадигмы обработки знаний. Важным компонентом интеллектуальной системы является база знаний. Актуальным в данном аспекте направлением инженерии знаний является онтологический анализ – метод, обеспечивающий построение онтологических систем и их использование для решения стоящих перед пользователем задач.

Понятие “онтология” заимствовано из философии, что означает учение о бытии, сущем как таковом в его наиболее общих характеристиках, независимо от частных проявлений и в отвлечении от логико-гносеологических и методологических вопросов, связанных с его познанием. В основе онтологического анализа лежит описание концептуально-логической модели предметной области в терминах сущностей, отношений между ними и преобразования сущностей, которое выполняется в процессе решения некоторой задачи [1]. Основное назначение онтологий – обеспечить общее понимание терминологии субъектами деятельности для того, чтобы их взаимодействие было корректным, эффективным и приводило к ожидаемым результатам. Чем больше различий (культурных, национальных, территориальных, интеллектуальных и др.) между субъектами процесса, тем тяжелее им прийти к соглашению без использования общей базы знаний.

В искусственном интеллекте под онтологией понимают БЗ специального типа, которые могут ”считываться” и пониматься, отчуждаться от разработчика и/или физически разделяться пользователями

[1]. Онтологии используются разработчиками Web-приложений в качестве лингвистической спецификации интеллектуальных информационных систем, призванной облегчить процесс передачи информации и обмена знаниями [2].

Онтологический анализ обычно начинается с составления словаря терминов, который используется при обсуждении и исследовании характеристик объектов и процессов, составляющих рассматриваемую систему, а также создания системы точных определений этих терминов. Кроме того, документируются основные логические взаимосвязи между соответствующими введенным терминам понятиями.

Результатом этого анализа является онтология системы как совокупность словаря терминов, точных их определений и взаимосвязей между ними. База знаний содержит описание предметной области и представляет концептуальный уровень информации в системе, необходимый для интеграции источников данных и организации содержательного поиска в них.

На электронном рынке основная задача – это поиск покупателем товара или услуги, наиболее соответствующего его потребности, а также продавца, способного предложить необходимый товар на взаимовыгодных условиях.

В определенной мере эта задача сводится к проблеме информационного поиска: по описанию потребности пользователя и описаниям товаров, представленным продавцами, надо осуществить их сопоставление и найти наиболее близкие (что аналогично выполнению информационных запросов к поисковым системам Интернета). При этом целесообразно учитывать не только явно представленные сведения, но и дополнительные знания как о продавце, так и о покупателе, а также об особенностях и свойствах товаров или услуг, которыми они хотят обменяться.

В будущем можно ожидать формирование “разумного” электронного рынка, имеющего общую глобальную онтологию, приемлемую для всех его участников – в частности, в рамках проекта Semantic Web. Все услуги, предлагаемые на таком рынке, также будут описаны на семантическом уровне. Но на сегодня это, к сожалению, еще очень отдаленная перспектива.

Более прагматичный подход связан с использованием локальных онтологий программными агентами покупателей и продавцов (что аналогично интеллектуальному поиску в Интернет). В дальнейшем под интеллектуальным поиском будем понимать поисковую процедуру, в которой для формирования результата используются знания о предметной области, интересующей пользователя, и средства их обработки, используемые в приложениях искусственного интеллекта.

Локальные онтологии значительно проще и компактнее глобальных, они отражают убеждения отдельных субъектов электронного рынка (как индивидуальных, так и корпоративных). Онтологическое представление знаний о субъектах экономической деятельности, которые входят в состав электронного рынка, позволяет сопоставить их информационные ресурсы и находить соответствия между предлагаемыми товарами и услугами не на формальном, а на семантическом уровне (например, описания могут быть представлены на различных языках, с использованием синонимии, обобщения или уточнения) [3].

На основе вышесказанного предлагаются следующие рекомендации субъектам электронного рынка:

- покупателям следует осуществить онтологический анализ своих потребностей, описать основные термины и отношения между ними (типа «класс»-«подкласс», «часть»-«целое»). Основой для этого может послужить какая-либо ранее созданная онтология соответствующей предметной области, а также автоматическая обработка текстовых документов, связанных с удовлетворением пользователем своих потребностей (инструкций, чеков, рекламных проспектов и т.п.) с целью выделения содержащейся в них терминологии – тезауруса покупателя T_s;

- продавцам следует составить перечень предлагаемых ими товаров, сведения о которых хранятся в их базе данных (интеграция различных баз данных возможна, если в онтологии существует понятие, соответствующее элементам структур различных источников данных), и сопроводить каждый из них описанием, а затем не только сгруппировать описанные товары, но и указать более сложные связи между ними (типа «используется совместно с», «применяется для обработки» и т.д.), а также связать используемые термины с какой-либо подходящей онтологией и в результате сформировать тезаурус продавца T_b.

При поиске соответствий между запросом покупателя и предложениями продавцов вначале находится множество товаров, формально соответствующее запросу покупателя, а затем осуществляется его упорядочение по степени близости тезаурусов покупателя и продавцов, т.е по соотношению пересечения тезаурусов к их объему. Будет найден “наиболее подходящий” продавец, тезаурус которого удовлетворяет условию (1):

$$T_{b_{best}} = \frac{\left|T_{b_{best}} \cap T_s\right|}{\left|T_{b_{best}}\right| + \left|T_s\right|} = \max_{i=1}^{n} \frac{\left|T_{b_i} \cap T_s\right|}{\left|T_{b_{it}}\right| + \left|T_s\right|} \qquad (1).$$

Выводы. Объединение развитых локальных онтологий субъектов электронного рынка в единую базу знаний позволит обеспечить его функционирование на новом интеллектуальном рынке и приведет к значительному увеличению его эффективности функционирования.

Список литературы

1. Т. А. Гаврилова, В. Ф. Хорошевский «Базы знаний интеллектуальных систем» — СПб: Питер, 2000. — 384с.:

2. Мошела Д. Бизнес-перспективы информационных технологий: как заказчик определяет контуры технологического роста. – М.:МПБ ”Деловая культура”. – 2004.– 252 с.

3. Плескач В. Л., Рогушина Ю. В. Информационное обеспечение субъеков экономической деятельности на рынке информационных услуг // Збірник наукових праць НАУ “Проблеми підвищення ефективності інфраструктури ”. – Випуск № 11. – 2005. – С. 230–244.

Проблемы преобразования программных моделей для моделирующих сред

Поленов М. Ю., *e-mail: polenov@dce.tsure.ru*
Таганрогский государственный радиотехнический университет, докторант,
г. Таганрог, пер. Некрасовский 44;

В настоящее время одной из наиболее значимых характеристик сред визуального моделирования сложных динамических систем является легкость их программирования, поскольку основная работа пользователя заключается именно в программировании и отладке моделей. В большинстве случаев эффективность среды во многом определяется тем, насколько быстро можно освоить ее язык моделирования и какой уровень подготовки пользователя требуется для этого.

Учитывая эту характеристику, разработчики современных пакетов моделирования стремятся максимально упростить процесс программирования моделей. Анализ пакетов визуального моделирования сложных систем [1] показывает, что практически все они основаны на использовании графических средств и языков описания проектов и моделей в виде структурных схем, графов связей, принципиальных физических схем, а также на использовании средств задания моделей в форме уравнений, записанных на конкретном языке программирования или моделирования. Как правило, каждый пакет имеет достаточно большую библиотеку моделей для различных дисциплин и поддерживает мультидисциплинарное моделирование.

Однако, как показывает практика, для каждой отдельной задачи моделирования сложной системы зачастую требуется создавать свой набор уникальных моделей или использовать в качестве компонентов системы созданные ранее модели. В этом случае важно, чтобы среда моделирования позволяла пользователю самостоятельно расширять библиотеки моделей как путем создания новых, так и путем подключения существующих внешних моделей, разработанных в сторонних средах и написанных на других языках моделирования.

Что касается средств описания моделей, то любой современный пакет предоставляет пользователю широкие возможности задания моделей в графическом виде, в символьном виде, в виде совокупности уравнений или в текстовом виде на одном из языков моделирования. Возможности же импорта и подключения внешних моделей в таких средах или не предусмотрены или ограничиваются одной-двумя внешними средами (языками моделирования). Так, например, широко известный пакет MSC.EASY5 (MSC.Software) дает возможность “встраивать” в расчётную схему модели, построенные в среде MSC.Adams (этого же производителя), в системе MATLAB/Simulink (MathWorks) и в пакете NI $MATRIX_X$ (National Instruments, Inc.). Не менее известный пакет VisSim (Visual Solution Inc.) позволяет организовать взаимодействие со средой MatLab и импортировать модели из подсистемы Simulink; еще одна среда Dymola (Dynasim AB), основанная на перспективном языке моделирования Modelica, также ограничивается интерфейсом с подсистемой Simulink.

Поскольку каждая современная среда характеризуется большим числом собственных моделей для поддерживаемых дисциплин, и перечни таких дисциплин для различных сред зачастую значительно пересекаются, то весьма актуальной становиться задача обеспечения более широких возможностей импорта и экспорта внешних программных моделей, обеспечивающих значительное расширение библиотек моделирующих сред.

Для решения поставленной задачи предлагается два подхода. Первый основан на сохранении формальной схемы представления моделей и метода ее расчета, но предполагает преобразование данной

модели с внешнего языка на язык конкретной среды моделирования и подключение полученной модели при помощи модулей связывания как внешней модели. Второй подход основан на преобразовании исходной модели к платформонезависимому параметрическому описанию и дальнейшему переходу к формальной схеме средств описания моделей в импортирующей среде. В этом случае исходная внешняя модель подключается к библиотеке моделей как собственная модель.

Оба подхода требуют разработки новых методов и средств трансляции программных моделей. Связано это с тем, что использование для подобных преобразований существующих средств трансляции, так называемых компиляторов компиляторов, таких как: Lex&Yacc, Flex&Bison, Grammatica, ProGrammar, AnaGram, Visual Parse++, Elkhound, Styx и других, не всегда эффективно и основным недостатком известных подходов и средств является то, что они не реализуют методов многоязыковой трансляции сложных программных моделей. Действительно, каждый транслятор в данных системах традиционно строится для единственного языка программирования, в лучшем случае с поддержкой его разновидностей, и оптимизируется под выходной язык (для большинства перечисленных систем выходные языки – C, C++ и Java). Более того, каждый такой транслятор обычно работает только с программами определенной аппаратной платформы. Отмеченные обстоятельства делают актуальной разработку средств многоязыковой трансляции, позволяющих работать с произвольным набором входных и выходных языков программирования и моделирования.

С целью решения этой задачи в Таганрогском радиотехническом университете были разработаны и исследованы методы построения трансляторов [2], способных воспроизводить процессы многоязыковой трансляции, и основанные на использовании для трансляции идей искусственного интеллекта, а именно, продукционных систем. Также были разработаны реализующие их инструментальные средства трансляции и конверсии программ с одних языков программирования и моделирования на другие языки, названные Мультитранслятором [3].

Мультитранслятор (МТ) основан на Backtrack анализе и представлении информации о грамматиках входных языков программирования или моделирования в виде отдельных файлов продукционных правил, названных грамматическими модулями. МТ при помощи предикатных частей продукций производит полный грамматический разбор транслируемых моделей и на основе заключительных частей продукционных правил генерирует программу модели на требуемом выходном языке. Таким образом, он представляет собой универсальную инструментальную среду для обеспечения трансляции исходных программ, написанных на многих входных языках моделирования, в программы на одном из выходных языков.

Разработанные средства многоязыковой трансляции позволили реализовать оба предложенных подхода к обеспечению импорта внешних моделей в моделирующих средах. Реализация подходов проводилась на примере среды виртуального моделирования сложных динамических систем Virtual Test Bed (VTB) [4]. Данная среда, как и большинство рассмотренных выше, поддерживает импорт моделей из подсистемы MATLAB/Simulink. Но в процессе работы с VTB потребовалось расширить перечень импортируемых языков моделирования такими языками, как ACSL, Modelica и специализированным языком структурного моделирования SML.

В рамках реализации первого подхода были разработаны грамматические модули Мультитранслятора для входных языков ACSL, Modelica и SML, генерирующие модели на языке C++ (ACSL2VC, Modelica2VC и SML2VC), поскольку данный язык является языком описания VTB-моделей.

Организация подключения генерируемой модели была основана на сохранении ее свойств согласно представлению на исходном языке. В объявляемом классе модели использовались два метода: первый для инициализации переменных модели и второй для реализации модели средствами C++, но с сохранением полностью логики ее работы соответствующей исходной программе. Также в данном классе использовалась специальная функция для организации интерфейса с модулем подключения (компоновщиком) модели и согласования шагов моделирования исходной модели и среды VTB.

При организации связи оттранслированных моделей с VTB использовались два основных метода подключения моделей [5]: натуральное связывание и сигнальное связывание. При натуральном связывании и моделировании в электротехническом домене, подключаемая модель, в зависимости от вида уравнений исходной модели, представлялась как источник тока, управляемый напряжением или как источник напряжения, управляемый током. При сигнальном связывании, подключаемая модель представлялась как источник напряжения, управляемый напряжением. Поскольку сгенерированная модель должна была подключаться к VTB в виде dll, то при генерации модели для каждого метода связывания использовался специальный компоновщик моделей, содержащий соответствующий шаблон.

Однако отличия в формальных схемах, используемых в различных средах и языках моделирования, не позволяют сделать процедуру преобразования моделей полностью автоматической и приводят к необходимости уточнения системных параметров транслируемых моделей, что, в свою очередь, требует соответствующей доводки оттранслированного текста модели.

Для решения этой проблемы был предложен и реализован специальный дополнительный интерактивный модуль [5], который на основе диалога с пользователем и полученных от него параметров завершает формирование модели. Данный модуль, названный Мастером генерации моделей,

активно взаимодействует, с одной стороны с грамматическим модулем входного языка, основываясь на особенностях соответствующего метода подключения, с другой стороны, с соответствующим шаблоном моделей. Работа Мастера завершает генерацию преобразованной Мультитранслятором модели, которая после компиляции и в виде dll подключается к среде моделирования, в данном случае к VTB, при помощи соответствующего Компоновщика моделей.

Поскольку современные моделирующие среды для описания моделей используют различные языковые средства и внутренние форматы, то в качестве единого формата целесообразно выбрать один из структурированных языков описания не зависящих от конкретной платформы. В последнее время в качестве такого языка все чаще используется язык XML, достаточно хорошо подходящий для параметрического описания моделей и их характеристик. Например, последняя версия среды VTB в качестве внутреннего формата для хранения моделей использует XML.

Второй подход к преобразованию и подключению внешних моделей также основан на использовании Мультитранслятора. В рамках реализации данного подхода в качестве единого формата представления моделей был выбран язык XML, а для трансляции моделей в формат XML-шаблона были разработаны соответствующие грамматические модули МТ для языков ACSL, Modelica и SML (ACSL2XML, Modelica2XML и SML2XML)[6]. Помимо Мультитранслятора для импорта моделей в среду VTB использовалась работающая с XML-описаниями система Paragon [7], являющаяся, по сути, расширенной базой данных моделей. В свою очередь, система Paragon при помощи пользовательских средства описания моделей самой среды VTB осуществляла их импортирование в эту среду.

При данном подходе грамматические модули МТ были ориентированы не на C++-код, а на язык XML-представлений. Но поскольку реализованные ранее модули в предикатных частях продукционных правил содержали полное описание грамматик выбранных языков, то это позволило значительно сократить время разработки новых грамматических модулей с необходимой корректировкой только заключительных, генерирующих частей продукционных правил. Однако, как и при первом подходе в процессе трансляции возникали ситуации требующие доопределения моделей, что также привело к необходимости использования Мастера генерации моделей. Для обеспечения универсальности Мастера и использования его при обоих подходах он был сформирован в виде dll-библиотеки, имеющей набор необходимых запросов, активизируемых при помощи функций вызовов из тела используемого грамматического модуля Мультитранслятора. В результате Мастер получил непосредственный доступ к генерируемому при трансляции коду модели.

Проведенные экспериментальные исследования сгенерированных МТ и импортированных в среду VTB внешних моделей показали корректность проводимых преобразований. Использование Мультитранслятора в качестве основы инструментальных средств импорта моделей позволит не только значительно расширить набор языков моделирования при создании новых моделей, но и использовать многочисленные библиотеки моделей, накопленные в уже существующих средах моделирования и реализовать возможность организации комплексного моделирования междисциплинарных моделей.

Список литературы

1. Ю. Б. Колесов, Ю. Б. Сениченков. Имитационное моделирование сложных динамических систем. [Электронный ресурс]. (http://www.exponenta.ru/soft/Others/mvs/mvs.asp)

2. В. Ф. Гузик, Ю. В. Чернухин, М. А. Самарин, О. А. Мунтян. Принципы построения универсального транслятора продукционного типа// Методы кибернетики и информационные технологии: Сборник научных трудов. Вып.3 – Саратов: ГосУНЦ «Колледж», 1999. – С.20-28.

3. Ю. В. Чернухин, М. Ю. Поленов, Р. В. Фадеев. Интерактивная среда мультиязыковой трансляции сложных программных моделей. //Анализ и моделирование развивающихся интеллектуальных систем: Межвуз. сборник науч. трудов. Вып.4 – Ростов н/Д: Изд-во СКНЦ ВШ, 2003. – С.10-20.

4. Gokdere L., Brice C., Dougal R. A Virtual Test Bed for Power Electronic Circuits and Electric Drive Systems // Proceedings of 7th IEEE Workshop on Computers in Power Electronics (COMPEL 2000). Blacksburg, VA, USA, 2000.

5. Ю. В.Чернухин, М. Ю. Поленов. Инструментальная подсистема многоязыковой трансляции виртуальных моделирующих систем// Тематический выпуск "Интеллектуальные САПР". Известия ТРТУ, № 3. – Таганрог: Изд-во ТРТУ, 2004. – С.115-120.

6. Чернухин Ю. В., Гузик В. Ф., Поленов М. Ю., Писаренко С. Н. Мультитрансляция сложных текстовых данных// Искусственный интеллект, №4, Донецк, Украина: Изд-во «Наука і освіта», 2005. С.302-310.

7. V. Chaudhary, M. Francis, X. Huang, H. A. Mantooth. PARAGON – A Mixed-Signal Behavioral Modeling Environment // Proceedings of IEEE International Conference on Communications, Circuits and Systems (ICCCAS 2002), Chengdu, China, 2002.

Исследование некоторых характеристик сетей массового обслуживания [1]

Пономарев Д. Ю.

Красноярский государственный технический университет
660074, Красноярск, ул. Киренского, 26
E-mail: kafaes@krasmail.ru

Моделирование информационных систем различного назначения очень часто сводится к исследованию различных параметров и характеристик систем и сетей массового обслуживания. Теория массового обслуживания предлагает достаточно много возможностей по оценке вероятностно-временных характеристик систем массового обслуживания различного вида, однако объединение систем в единую сеть массового обслуживания приводит к усложнению задачи анализа характеристик систем, особенно при исследовании сетей большой размерности. В данной работе предлагается метод оценки вероятностно-временных характеристик сетей массового обслуживания с использованием тензорной методологии, что позволяет обеспечить более простое решение данной задачи.

Исследованию систем массового обслуживания посвящено достаточно большое количество работ. Известны результаты, полученные для систем с различными параметрами и характеристиками: по входному потоку , по распределению длительности обслуживания, по количеству обслуживающих приборов, по размерности буфера, приоритетам и т.д. Однако, в связи с тем, что в реальности применение отдельных систем массового обслуживания, а тем более использование систем массового обслуживания, как моделей реальных информационных (измерительных, телекоммуникационных, управляющих) систем, сильно ограничено. Для моделирования большинства реальных комплексов обслуживания информационных потоков необходимо использовать набор систем массового обслуживания с определенной структурой связи между ними, т.е. в качестве модели требуется применение сетей массового обслуживания.

Сети массового обслуживания также исследуются достаточно давно, существует достаточно много классов сетей с известными результатами, однако при этом достаточно сложно применять полученный математический аппарат для инженерного анализа реальных информационных систем. В связи с этим, а также со все более усложняющимися технологиями обслуживания информационных потоков, увеличением масштабов сетей, усложнением процессов обеспечения качества обслуживания, актуальной становится задача обеспечения дальнейшего развития информационных сетей простым, удобным и понятным аппаратом исследования основных характеристик данных сетей.

В данной работе предлагается применить для оценки вероятностно - временных характеристик (ВВХ) сетей массового обслуживания математический аппарат преобразования систем координат, рассматривая сети массового обслуживания, как геометрические объекты, проекции которых в различных системах координат различны, но физические свойства самих объектов при этом не меняются. Основоположником тензорной методологии анализа систем является известный американский ученый и инженер Г. Крон, который впервые использовал тензорный анализ и топологию в приложении к теории электрических сетей ([1]). Дальнейшее развитие идеи тензорного анализа для информационных систем получили в работах Петрова А.Е, Арменского А.Е., Кузнецова О.Л., Петрова М.Н. ([2]) и др.

Для инженерного анализа ВВХ сетей массового обслуживания (СеМО) в данной работе предлагается использовать в качестве инвариантного уравнения известное выражение для определения коэффициента использования устройств (ρ), дающее связь между интенсивностью поступления вызовов (λ) и средним временем обслуживания:

$$\rho = \lambda t. \tag{1}$$

Однако, в отличие от методов в ([3,4,5]) позволяющих решать вопрос исследования сетей массового обслуживания, как моделей информационных сетей, с применением контурного метода, в данной работе предлагается использовать несколько другой подход, связанный с модификацией инвариантного уравнения (1):

$$\lambda = \rho\mu. \tag{2}$$

[1]Работа поддержана грантом Президента Российской Федерации для государственной поддержки молодых российских ученых МК-1232.2005.9.

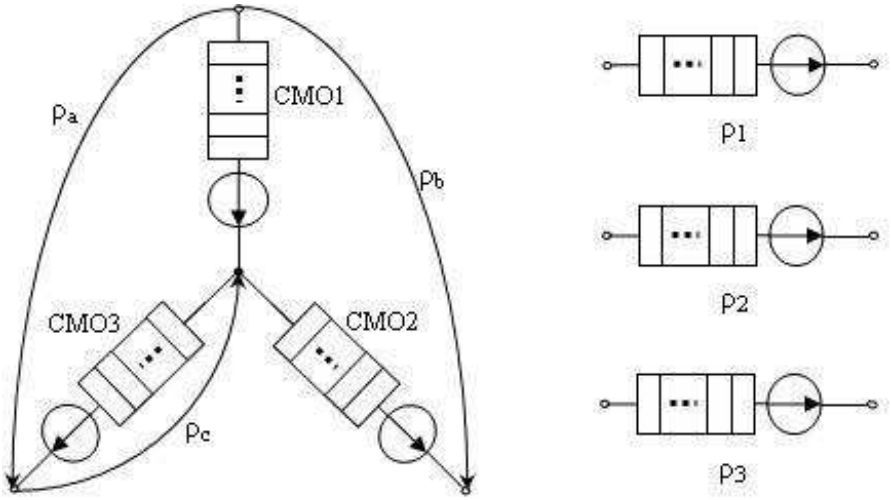


Рис. 1: Исследуемая сеть массового обслуживания (слева – исходная, справа – примитивная)

Определяя примитивную сеть, состоящую из такого же количества систем и описываемых инвариантным уравнением: $\bar{\lambda}' = \rho'\mu'$, находим матрицу перехода ($\bar{A}$) от одной проекции к другой: $\bar{\rho}' = \bar{A}\bar{\rho}$, и определяем составляющие матричного уравнения:

$$\bar{A}^T\bar{\lambda}' = \left(\bar{A}^T\bar{\mu}'\bar{A}\right)\bar{\rho}. \tag{3}$$

Рассмотрим использование данного подходя для СеМО, изображенной на рис.1. Для данной сети загрузка СМО1 будет определяться как: $\rho_1 = \rho_a + \rho_c$, для остальных аналогично: $\rho_2 = -\rho_a + \rho_b - \rho_c$; $\rho_3 = -\rho_c$. Следовательно, матрица перехода будет иметь вид:

$$A = \begin{Vmatrix} 1 & 0 & 1 \\ -1 & 1 & -1 \\ 0 & 0 & -1 \end{Vmatrix}$$

Запишем матричное уравнение (3) для рассматриваемой сети:

$$\begin{pmatrix} \mu_1+\mu_2 & -\mu_2 & \mu_1+\mu_2 \\ -\mu_2 & \mu_2 & -\mu_2 \\ \mu_1+\mu_2 & -\mu_2 & \mu_1+\mu_2+\mu_3 \end{pmatrix} \begin{pmatrix} \rho_a \\ \rho_b \\ \rho_c \end{pmatrix} = \begin{pmatrix} \lambda_1-\lambda_2 \\ \lambda_2 \\ \lambda_1-\lambda_2-\lambda_3 \end{pmatrix}$$

Решая данную систему относительно узловых загрузок ρ_a, ρ_b, ρ_c, при заданных интенсивностях обслуживания и поступления можно найти распределение загрузок в исследуемой сети. С другой стороны, в данной сети могут быть известны любые другие параметры, и возможно нахождение значений тех величин, которые наиболее интересны.

Кроме того, при объединении возможностей контурного ([3,4,6]) и представленного методов, появляется возможность решения задачи определения вероятностно-временных характеристик сетей так называемого ортогонального типа [1], что позволит обеспечить дальнейшее расширение класса сетей, для которых возможно применение тензорного подхода.

В результате можно отметить, что данный подход позволяет при неизвестных теоретических выражениях для некоторого класса СеМО оценивать их загрузку, а, следовательно, и анализировать качество обслуживания информационных потоков для сетей практически любой размерности. Кроме того, формализация процедуры определения распределения загрузки по узлам позволяет обеспечить простоту программной реализации данного метода ([5]).

Список литературы

1. Крон Г. Тензорный анализ сетей — М.: Сов. радио, 1978.

2. Петров М. Н. Вероятностно-временные характеристики в сетях и системах передачи интегральной информации —Красноярск: КГТУ, 1997.

3. Ponomarev D. U. Tenzors analysis for investigation next generation network // Proceedings of IEEE International Siberian Conference on Control and Communications (SIBCON-2005) Ц Tomsk: The Tomsk IEEE Chapter and Student Branch. Ц 2005. Ц P. 53-57.

4. Пономарев Д. Ю. О подходе к анализу сетей массового обслуживания с использованием тензорной методологии // Труды V Международной конференции кИдентификация систем и задачи управлениянь SICPRO 06. Ц M: Институт проблем управления им. В. А. Трапезникова РАН. Ц 2006. Ц С. 697-704.

5. Красницкий И. Г., Пономарев Д. Ю. Программно-вычислительный комплекс для анализа вероятностно-временных характеристик сетей интегрального обслуживания. Ц M: ВНТИЦ, 2006. Ц N50200600037.

6. Пономарев Д. Ю. Тензорная методология в телекоммуникациях // Системы управления и информационные технологии. Ц 2006. Ц 1.1(23). Ц С. 161-165.

Язык описания нейронных сетей NNML

Попов В. Ю.,
профессор, Кафедра Алгебры и Дискретной Математики,
e-mail: Vladimir.Popov@usu.ru
Баклановский М. В.,
старший преподаватель, Кафедра Вычислительной Математики,
e-mail: max@suncloud.ru
Морнев М. Л.,
Математико-Механический Факультет,
e-mail: max.mornev@gmail.com
Уральский Государственный Университет им. А. М. Горького,
г. Екатеринбург, ул. Ленина, 51

Для описания, обработки и хранения нейронных сетей необходим удобный способ их представления. С описательной точки зрения нейронную сеть можно рассматривать как ориентированный граф с помеченными вершинами и ребрами. Поэтому задача представления нейронных сетей включает в себя аналогичную задачу для графов. При этом желательно, чтобы инструмент представления был расширением уже имеющихся и хорошо зарекомендовавших себя общих способов представления данных, среди которых нужно в первую очередь отметить XML (общее описание XML см в. `http://www.w3.org/TR/REC-xml/`).

Общая идея предлагаемого нами способа представления нейронных сетей следующая. Исходный граф некоторым способом разбивается на поддеревья. Каждый из полученных подграфов размещается в отдельном слое. А для представления деревьев в слоях уже можно использовать готовые инструменты, такие, как XML.

Эту идею можно реализовать следующим образом. Будем рассматривать множество $\mathbb{T}(G)$ всех лесов, содержащихся в заданном орграфе G. Подмножество $F \subseteq \mathbb{T}(G)$ назовем расслоением, если $\bigcup_{T \in F} T = G$. Таким образом, расслоение есть набор лесов, полностью покрывающий исходный граф. Если нам дано расслоение как набор слоев, то для того, чтобы восстановить исходный граф по нему, каждой вершине в слое должен быть приписан идентификатор, уникальный в пределах слоя. Вершины с одинаковыми идентификаторами, лежащие в разных слоях, склеиваются в одну вершину исходного графа. Аналогично склеиваются и ребра.

Обычные способы представления графов, списки смежности и списки ребер, являются частными случаями расслоений. Список ребер есть расслоение, каждый слой которого содержит по одному ребру графа. Для реализации списков смежности расслоением необходимо выделить для каждой вершины графа отдельный слой, и поместить в этот слой саму вершину и все ребра, исходящие из нее.

Для хранения расслоения мы можем использовать XML-документ, структура которого определенной следующим образом. Корневой элемент называется `nn:system`. Он содержит подэлементы `nn:layer` и единственный (необязательный) подэлемент `nn:verts`. Элементы `nn:layer` хранят леса расслоения. Их потомками являютсяэлементы `nn:root`, задающие конкретыне деревья в слое. Элемент `nn:verts` содержит в подэлементах `nn:vert` дополнительную информацию о вершинах, например, тип нейрона. Элементы `nn:root` хранят деревья из элементов `nn:edge`, задающих ребра. Последние, в отличие от вершин, могут содержать дополнительную информацию о ребрах (например, вес соединения). При этом дублирования информации всегда можно избежать, удалив из расслоения кратные ребра.

Элементы `nn:root`, `nn:vert` и `nn:edge` имеют обязательный атрибут `id`. Для `nn:root` он задает идентификатор корня дерева. Для `nn:vert` атрибут `id` указывает на вершину, к которой относится дополнительная информация, содержащаяся в `nn:vert`. Для `nn:edge` этот атрибут задает идентификатор концевой вершины ребра. Начальная вершина ребра определяется в зависимости от того, в каком элементе оно содержится: если в `nn:root`, то начальной вершиной считается корень, в противном случае — концевая вершина родительского элемента `nn:edge`.

Элемент `nn:edge` имеет необязательный атрибут `dir`, указывающий направление ребра: от родителя к потомку или наоборот. В некоторых ситуациях использование этого атрибута позволяет существенно сократить количество слоев в представлении графа. Атрибут `dir` можно указывать и у элементов `nn:root`, `nn:layer`, `nn:system` — тогда он задает направление по умолчанию для всех ребер-потомков.

В качестве альтернативы рассмотренному XML-представлению можно использовать многослойный документ. Физически, он может быть представлен потоком символов с выделенным символом для разделения слоев. Внутри слоев деревья могут быть представлены тем же XML.

Строить расслоение предполагается при помощи жадного алгоритма. Возможно и построение с наложением некоторых условий минимальности. Для оценки эффективности представления графов расслоениями мы введем характеристики ξ и ζ, первая из которых есть минимальное число слоев в расслоении данного графа, а вторая — минимальное число вершин, с учетом кратностей. Величина ζ определяет размер представления графа расслоением, а ξ позволяет учесть накладные расходы на управление слоями. При этом, для модели в равной степени пригодны как ориентированные, так и обыкновенные графы.

Эффективность соответствующих построений характеризуют следующие два утверждения.

Теорема 1. $\xi(G) = \max_{v \in V} \mathrm{in}(v)$, *где* G *— бесконтурный орграф,* V *— множество его вершин,* $\mathrm{in}(v)$ *— количество ребер, для которых вершина* v *является концевой.*

Теорема 2. $m + 1 \leqslant \zeta(G) \leqslant 2(m + q) - n$, *где* G *— обыкновенный граф,* n *— число его вершин,* m *— число ребер,* q *— число компонент связности.*

Рассмотрим расслоения на конкретном примере сети. Ее топология описывается полным трехдольным графом вида $3 - 5 - 1$ (два входа и поляризатор), в котором удалили три ребра, идущих из первого слоя во второй. Для входов будем использовать обозначения `x`i, для нейронов скрытого слоя — `v`i и `u` для выхода. Ребра `x0-v3`, `x1-v1` и `x2-v4` в графе отсутствуют. Заметим, что, поскольку $\mathrm{in}(\mathtt{u}) = 5$, мы, согласно теореме 1, не смогли бы обойтись менее чем 5 слоями. Но, применив атрибут `dir` один раз, мы можем уменьшить количество слоев до 4.

```
<nn:system dir=backward>
  <nn:layer>
    <nn:root id=u>
      <nn:edge id=v1> <nn:edge id=x0 /> </nn:edge>
      <nn:edge id=v2> <nn:edge id=x1 /> </nn:edge>
      <nn:edge id=v3> <nn:edge id=x2 /> </nn:edge>
      <nn:edge id=v4 />
      <nn:edge id=v5 />
    </nn:root>
  </nn:layer>
```

```
  <nn:layer>
    <nn:root id=v2> <nn:edge id=x0 /> </nn:root>
    <nn:root id=v3> <nn:edge id=x1 /> </nn:root>
    <nn:root id=v5> <nn:edge id=x2 /> </nn:root>
  </nn:layer>
  <nn:layer>
    <nn:root id=v4> <nn:edge id=x0 /> </nn:root>
    <nn:root id=v5> <nn:edge id=x1 /> </nn:root>
    <nn:root id=v1> <nn:edge id=x2 /> </nn:root>
  </nn:layer>
  <nn:layer>
    <nn:root id=v5> <nn:edge id=x0 /> </nn:root>
    <nn:root id=v1> <nn:edge id=x1 /> </nn:root>
    <nn:root id=v2> <nn:edge id=x2 /> </nn:root>
  </nn:layer>
</nn:system>
```

Здесь используется 4 тега `nn:layer` с закрывающими, 10 тегов `nn:root` с закрывающими и 17 тегов `nn:edge` без закрывающих. Атрибут `id` используется 27 раз, атрибут `dir` — один раз; всего 45 тегов и 73 синтаксических элемента. Расслоение этой же сети, эквивалентное спискам смежности мы, ради краткости, не приводим. Кратные ребра в этом расслоении удалены. Расслоение содержит 9 тегов `nn:layer` с закрывающими, 17 тегов `nn:edge` без закрывающих и 9 тегов `nn:root` с закрывающими. При этом атрибут `id` используется 26 раз. В сумме получается 53 тега и 79 синтаксических элементов. Выигрыш составляет $17,7\%$ для тегов и $8,2\%$ для синтаксических элементов.

Перечисленные выше возможности не исчерпывают весь NNML. Полную спецификацию языка в виде XML-схемы и дополнительные примеры можно найти по адресу `ftp://cs.usu.edu.ru/langs/nnml/`.

Хорошо известно, что многие задачи не могут быть решены в рамках стандартной архитектуры сети [1, 2]. Поэтому, в языке описания нейронных сетей архитектурыне средства — шаблоны и им подобные конструкции — не являются бесполезными. Наряду с чисто описательными средствами, в NNML имеются вычислительные возможности. Их наличие обусловлено необходимостью решать задачи управления расслоениями (удаление маргинальных слоев, слияние слоев и т. п.), анализа расслоений (изоморфизм сетей), а также, необходимостью строить и поддерживать темпоральные структуры, используемые для накопления и исследования изменений в процессе обучения.

Список литературы.

[1] *M. L. Minsky, and S. A. Papert,* Perceptrons - Expanded edition, MIT Press, 1988.

[2] *J. Šíma and P. Orponen,* A computational taxonomy and survey of neural network models, IEEE Transactions on Neural Networks, vol. XX, no. Y, month 2001.

О логическом подходе к синтезу технологических процессов

Попов С. В.[1], Трифонова Е. Е.[2]
Институт прикладной математики им. М.В.Келдыша,
125047, Москва, Миусская площадь, д.4,
[1]С.н.с. s-v-popov@yandex.ru, [2]м.н.с., etrifonova@yandex.ru

Введение. Разработка экспертных систем, ориентированных на задачи с обширными множествами решений, особенно важна в областях, где выбор неадекватного решения влечет большие затраты. Например, выбор неправильного технологического процесса чреват существенными потерями как из-за неудовлетворительного качества конечного продукта, так и из-за высоких затрат на его реализацию. Оценка трудозатрат на разработку и анализ эффективности технологических процессов показали, что без привлечения информационных технологий не удается создать и оценить более одного конкурентоспособного технологического процесса. Осуществлять же не автоматизированный синтез технологических процессов затруднительно из-за большого разнообразия технологий получения конечного продукта. При этом необходимо выбирать из них наиболее эффективные в соответствии с многопараметрическими критериями.

Нами предлагается подход к решению задачи синтеза технологических процессов, основанный на логическом анализе предметной области и выделении наиболее характерных технологических этапов. При этом разработка технологии представляется как последовательность преобразований, где каждое последующее наследует результат предыдущего. В этом процессе выделены следующие этапы.

1) Логический синтез процесса, который позволяет получить схему переходов без акцентуации на имеющееся в распоряжении технологическое оборудование.

2) Уточнение логической схемы с учётом имеющегося в распоряжении оборудования, требований к качеству исходных, промежуточных и окончательных продуктов, времени выполнения отдельной технологической операции и т.п.

3) Окончательное представление технологического процесса в терминах понятных технологу с окончательно выбранными оборудованием и ресурсами (финансовыми, людскими, энергетическими и т.д.).

4) Размещение технологического оборудования на производстве с учетом всех требований технологических операций, транспортировки, техники безопасности, типа производства и т.д.

Ниже задача проектирования технологических процессов на основе логического подхода иллюстрируется разбором первых двух этапов. Третий этап – это обычное отображение формально-логического представления в формат, принятый на производстве. Размещение технологического оборудования на производственной площади это задача, решаемая с использованием так называемой пространственной логики. Таким образом, методика автоматизированного проектирования технологических процессов базируется, с одной стороны, на классификации оборудования, а с другой, – на методах решения переборных задач, разработанных авторами.

Синтезированный технологический процесс представляет собой информационную модель реального объекта (технологического процесса). Подобные информационные модели обладают тем преимуществом, что сопровождают весь жизненный цикл технологического процесса. Они не только позволяют оценивать те или иные целевые функции, но и дают возможность моделировать различные ситуации, в том числе и нештатные, планировать проведение регламентных работ и т.п.

Рассмотрим подробнее предлагаемый подход к решению задачи синтеза технологических процессов.

1. Содержательная постановка задачи. Пусть задан вектор σ_{inp} входных продуктов и ресурсов, которые необходимы и доступны для получения конечного продукта. Последний характеризуется определенными признаками, образующими вектор σ_{out}. Входными могут быть людские, финансовые и временные ресурсы, а также характеристики материалов, доступных для получения окончательного продукта. Выходной вектор описывает качество конечного продукта и определенные технологические параметры.

Для решения задачи синтеза технологического процесса требуется построить технологическую цепочку, так преобразующую входной вектор в выходной, что, во-первых, каждый ее шаг был бы обоснован (было в наличии соответствующие оборудование и ресурсы), и, во-вторых, вся технология удовлетворяла целевой функции (описывающей финансовые вложения, временные характеристики, конкурентоспособность окончательного продукта и т.д.)

Таким образом, технологический процесс можно представить как объект (формально представимый ор-графом), развивающийся во времени. Каждый переход между его состояниями представляет собой некоторое отображение входных параметров в выходные и реализуется определенным набором инструментов. Следовательно, разработка технологического процесса – это задача, с одной стороны,

синтеза функциональных и временных преобразований вектора σ_{inp} входных продуктов и ресурсов в выходной вектор σ_{out}, а с другой, описание того же процесса на языке оборудования (элементной базы), которое позволяет осуществлять такие преобразования.

2. Синтез технологических процессов. Синтез технологического процесса можно рассматривать как разновидность более общей задачи синтеза информационных моделей объектов управления. В ходе процесса моделирования необходимо достигнуть следующего:

– синтезировать информационную модель, отображающую весь технологический процесс, реализуемый объектом управления;

– исследовать заданную целевую функцию в зависимости от значений микропараметров модели, т.е. параметров базисных элементов модели (стоимость, надежность и т.п.), и от ее макропоказателей (общая производительность, качество получаемого продукта и т.п.);

– уточнить разработанную модель с целью оптимизации целевой функции.

Для построения информационной модели необходимо наличие следующих компонентов:

– спецификация моделируемого объекта, включающая логическое исчисление, которое описывает технологические преобразования объекта управления и спецификацию конкретных условий, в которых искомая технология реализуется;

– БД базисных технологических элементов с указанием их параметров и осуществляемых преобразований;

– перечень целевых функций, зависящих как от глобальных, так и от локальных параметров.

Логическое исчисление в обобщенных терминах должно описывать искомую технологию. Трудность реализации подобной задачи заключается в невозможности априорно задать всю необходимую информацию. Поэтому разработчик переходит к более детальной версии исчисления после разработки нескольких упрощенных вариантов. Возможные технологические переходы, описанные в логическом исчислении, представлены на языке временной логики. Каждая конструкция подобного языка представляет собой импликацию, которая трактуется следующим образом: ее посылка описывает условия, которым должен удовлетворять конкретный вход (мощность электрического источника, параметры входного продукта и т.п.), заключение – конкретные условия на выходные потоки (параметры, определяющие качество выходной продукции, производительность выходного потока и т.п.). Имя конструкции соответствует технологической единице (насос, труба и т.п.), обеспечивающей преобразование.

Все описания технологических переходов хранятся в БД оборудования, где помимо данных, касающихся параметров конкретного оборудования (напряжение, производительность и т.п.), хранятся условия его применения и получаемый при этом результат. Достоинство подобного представления временной модели состоит в том, что обеспечивается возможность синтеза большого числа моделей, исходя из входных и выходных параметров, а также целевой функции.

Метод синтеза временных моделей представляет собой процедуру нахождения неподвижных точек логических программ. Эта процедура основана на том, что при определённой интерпретации сама спецификация предметной области и начальных условий является программой для построения модели. Тогда неподвижная точка – это такая интерпретация исчисления, при которой ни одно из действий программы не выполнимо. В результате неподвижная точка есть искомая модель для заданной предметной области с конкретными начальными условиями. Перебор при нахождении неподвижных точек для формул, описывающих технологические процессы, не превосходят полинома от числа узлов графа переходов.

После получения окончательной временной модели, детально представляющей технологический процесс, осуществляется оценка значения целевой функции. Если предварительные оценки удовлетворительны, то наступает следующий этап синтеза – разработка конкретной технологии с заменой каждого перехода модели на последовательный технологический участок, состоящий из имеющихся в БД элементов.

Предлагаемый подход синтеза технологического процесса сверху вниз (т.е. от укрупненных переходов к их детальному представлению) предоставляет возможность совместить разработку информационной модели с наполнением БД необходимым технологическим оборудованием.

3. Временная логика для описания синтеза технологических процессов: логическая схема переходов. Не указывая всех деталей, приведем описание временной логики, которая служит для описания технологических переходов. Дополнительными связками по сравнению с обычными пропозициональными выступают следующие:

$@^n$ – (доступно) спустя n моментов времени; $\Diamond$ – (доступно) в некоторый момент времени;

$\Box$ – (доступно) во все моменты времени.

Каждое логическое выражение имеет вид продукции:

Оборудование, Ресурсы, Комплектующие → Результат.

Оборудование – это приборы, механизмы, приспособления и т.п., необходимые для выполнения технологической операции. *Ресурсы* – дополнительные условия, необходимые для нормальной работы

оборудования и выполнения данной операции (энергетические ресурсы, рабочая сила и ее квалификация и т.п.). *Комплектующие* – необходимые материалы для выполнения технологической операции .

Интерпретация каждого выражения такова: если все переменные слева истинны (что содержательно обозначает наличие соответствующих оборудования, ресурсов и комплектующих), то выполняется действие, в результате которого оказывается истинным выражение справа. На языке технологических процессов это обозначает получение нового продукта, который может быть поставлен потребителю или участвовать в последующем процессе.

На первом этапе логические формулы описывают лишь состояния технологического процесса без учета конкретных параметров, например, производительности станка, квалификации работника или стоимости всей операции. Получаемая модель представляет сбой ор-граф, вершины которого суть моменты времени с указанием кортежей логических переменных (здесь оборудования, ресурсов и комплектующих, необходимых для перехода в следующее состояние), принимающих истинное значение.

4. Второй этап логического синтеза технологических процессов: уточнение схемы переходов. Второй этап синтеза технологических переходов сводится к уточнению оборудования и параметров промежуточных состояний продукта. Каждому переходу временной диаграммы можно сопоставить одно из двух приведенных ниже логических выражений.

Начальное состояние (x), **Финальное состояние** (y) $\Rightarrow$
Оборудование ($f_1(x, y)$), **Ресурсы** ($f_2(x, y)$), **Процесс** ($f_3(x, y)$);
Начальное состояние(x), **Оборудование**(y), **Ресурсы**(z), **Процесс**(u) $\Rightarrow$
Финальное состояние ($f(x, y, z, u)$).

Первое из них по параметрам x продукта, поступающего на вход операции, и параметрам y продукта после обработки указывает необходимое оборудование, ресурсы и требуемую последовательность действий (здесь $f_3(x, y)$), которая приведет к искомому результату. Второе – по начальному состоянию, оборудованию, ресурсам и описанию процесса указывает состояние, которое может быть получено в после обработки.

Для одного перехода могут быть составлены несколько подобных логических выражений в зависимости от задаваемых параметров. Не представляет труда сделать подобный выбор автоматизированным, когда вмешательство человека происходит лишь на этапе контроля за правильностью описания переходов и восстановления недостающих параметров. Таким образом, на втором этапе задача синтеза также представляет собой построение логической модели, но уже для исчислений первого порядка.

Заключение.

Задача синтеза технологических процессов представляет интерес как с точки зрения математики – появляется новая область переборных задач и возникает необходимость в разработке адекватного логического инструмента, так и с практической точки зрения – появляется возможность решать новые задачи с большим экономическим эффектом. Предлагаемые в работе методы решения подобных задач могут стать основой для создания экспертных систем, предназначенных для синтеза технологических процессов в той или иной предметной области.

Список литературы

1. Попов С. В. Логические модели как неподвижные точки программ. Седьмая национальная конференция по искусственному интеллекту с международным участием. КИИ'2000. Труды конференции. М.: Изд-во Физико-математической литературы. 2000. С. 214-222.

2. Попов С. В. Логическое моделирование. Математические вопросы кибернетики. М.: ФИЗМАТЛИТ, 2001. С. 167-214.

Использование модифицированных генетических алгоритмов в задаче распределения ресурсов

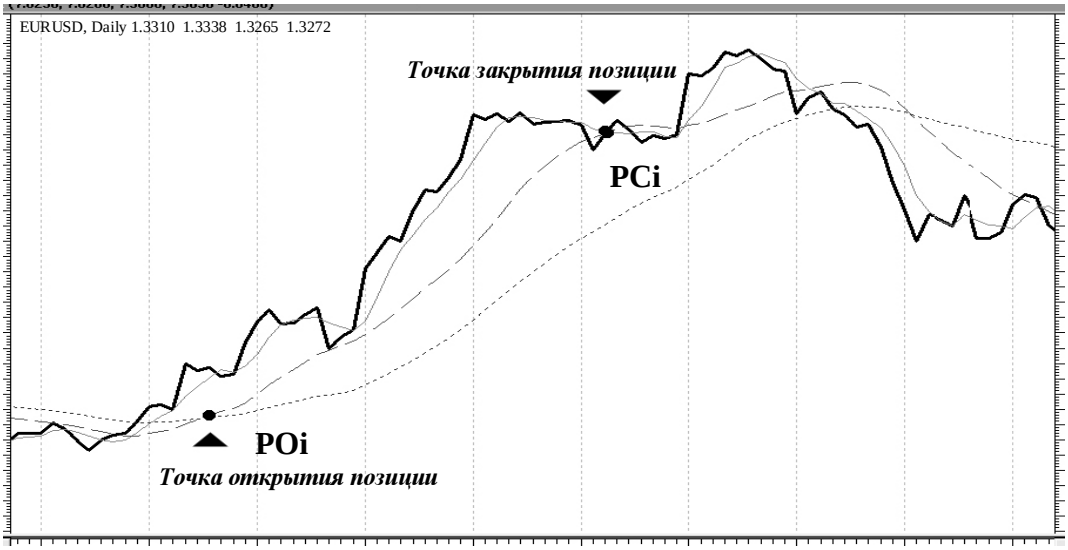
Портнов К. В., *e-mail: sk7@mail.ru*
Самарский Государственный Технический Университет, каф. «Информационные технологии», 443100, г.Самара, ул.Молодогвардейская 244,

В работе предлагается методика оптимизации портфеля методов прогнозирования при принятии решений по управлению ресурсами(в нашем случае капиталом) в условиях неопределенности. Рассмотрен подход оценки эффективности и риска методов прогнозирования при использовании на волатильных инструментах (рынке ценных бумаг).

Предложены подходы к оптимизации портфеля методов прогнозирования с использованием модифицированных генетических алгоритмов.

В настоящее время много сил уделяется разработке различных методов прогнозирования, основанных на обработке временных рядов, для принятия решений в условиях неопределенности. Данные методы применяются в различных отраслях промышленности и экономики при работе с инструментами, имеющими стохастический характер, и сильную динамику во времени. К ним можно отнести цены на сырье необходимое при производстве, какие либо качественные характеристики процессов, имеющие динамический характер. Эти методы также включают в себя набор некоторых правил, рекомендующих принятие решения или его отмену. Например, для большой группы методов, это такие события которые определяются пересечением индикаторов характеризующих направления или силу прогнозируемой величины, характеризующих например смену тенденций движения. Данные методы подробно рассмотрены в работе [1].

На рисунке 1 приведен пример сигнала PO_i на покупку (точку открытия позиции) и сигнала на продажу PC_i (точку закрытия позиции) для валютного курса EUR/USD, основанного на пересечении скользящих средних разных порядков.



Р и с. 1. График цены закрытия EUR/USD, с частотой дискретизации в 1сутки (сплошная толстая линия), с тремя скользящими средними, рассчитанных по ценам закрытия за 4 (сплошная тонкая линия), 9(крупный пунктир) и 18(мелкий пунктир) дней.

Подобным образом работают и бесчисленное множество других методов и их комбинаций.

Однако использование различных методик прогнозирования, не лишены недостатков, т.к. используются лишь в определенных условиях (*рыночных условиях*). В конечном счете, риск инвестора сводится к виду занимаемой рыночной позиции и ее размеру, показывающего насколько тот или иной вариант развития событий на оценивается управляющим (*инвестором*), как наиболее вероятный. Вид и размер этой ставки определяются на основе используемого алгоритма принятия решений, или, другими словами, используемого метода прогнозирования. Следовательно, общий риск потерь при активном управлении (*инвестировании*) напрямую зависит от эффективности используемого управляющим (*инвестором*) метода, принятия решения.

Действительно, в случае динамичной эффективности, рынка управляющий (*инвестор*) располагает некоторыми возможностями для оценки вероятных будущих рыночных движений в виде используемых им методов прогнозирования и теоретически может правильно оценивать моменты, как роста, так и падения всего рынка в целом. В этом случае потенциальный доход инвестора напрямую будет зависеть только от его аналитических навыков в виде потенциальной эффективности используемых в анализе алгоритмов принятия управленческих (*инвестиционных*) решений, при этом единственное, чем действительно рискует инвестор – вероятность того, что полученный прогноз не оправдается, в силу того, что при текущем уровне эффективности рынка используемый метод окажется неработоспособным.

Однако независимо от эффективности методов прогнозирования и принятие решений, всем им свойственна, динамика их эффективности во времени, что обуславливает поиск и формирование инструмента для обеспечения качественного подбора набора методов для обеспечения устойчивости принимаемого решения, в условиях неопределенности.

Данная методика должна предусматривать:

- построение модели отбора и оценивания портфеля методов активного управления ресурсами в условиях неопределенности и ее математическую формализацию;

- разработку структуры исходных данных для оптимизации портфеля методов стратегий активного управления;
- разработку структуры ограничений при оптимизации портфеля методов активного управления;
- разработку обобщенного критерия эффективности и риска методов активного управления;
- оптимизацию набора методов прогнозирования для обеспечения устойчивости принимаемого решения;

Выделив основную идею модели активного управления портфелем, перейдем к ее количественному описанию в виде математической интерпретации.

Для построения модели оптимального управления материальными ресурсами необходимы некоторые данные. Имеется некоторая дискретная историческая статистика (временной ряд) $\boldsymbol{c}$, динамики прогнозируемый величины $\boldsymbol{C}$ (цена биржевого инструмента), заданная с установленной частотой дискретизации (например 1 час, 1 день), на определенном временном интервале. Предполагаем, что в некоторый начальный момент времени $\boldsymbol{T_0}$, мы располагаем каким то объемом средств (капиталом) $\boldsymbol{K}$, имеется также некоторый набор (множество) методов прогнозирования Δ, используемых для принятия управленческого решения. Эффективность каждого $\boldsymbol{j}$-ого решения принятого с использованием выбранного $\boldsymbol{i}$-ого метода, может быть оценена на исторических данных прогнозируемой величины $\boldsymbol{C}$, на определенном интервале времени как:

$$\boldsymbol{r_{ij} = \psi_{ij} * \frac{(c_{t_к j} - c_{t_н j})}{c_{t_н j} * (t_к - t_н)}}, \tag{1.1}$$

а ожидаемая эффективность $\boldsymbol{i}$-ого метода определяется как:

$$\boldsymbol{R_i = \frac{1}{N_j} \sum_{j=1}^{N_j} r_{ij}}, \tag{1.2}$$

где: $c_{t_н j}$ - значение C в момент $t_н$, т.е. при принятии j-ого решения (на основании сигнала),

$c_{t_к j}$ - значение C в момент $t_к$, т.е. при отмене j-ого решения (на основании сигнала),

$(t_к - t_н)$ - временной интервал, количество периодов дискретизации между принятием решением и его отменой,

ψ_{ij} - принятое j-е решение, с использованием $\boldsymbol{i}$-ого метода: $\psi_j = \{-1;+1\}$, причем:

для решения о покупке (длинная позиция) $\psi = +1$,

для решения о продаже (короткая позиция) $\psi = -1$,

N_i - количество принятых решений при использовании $\boldsymbol{i}$-ого метода;

Указанная в выражении 1.2. эффективность, определяется эмпирическим путем оценки выбранных стратегий на исторических данных динамики выбранной переменной величины. Данную оценку применительно к инвестиционным инструментам обращающихся на организованных рынках, способны производить программные продукты такие как Equis MetaStock Professional 7.0 или Omega Research TradeStation 2000i.

Логика подхода заключается в том, что управляющий (*инвестор*) не будет склонен использовать заведомо неэффективный метод, соответствующий политике управления, соответственно поставим условие:

$$R_i > 0 \tag{1.3}$$

Таким образом, введем множество $\boldsymbol{M}$ - множество допустимым к использованию методов, при этом $\boldsymbol{M} \in \Delta$. Существует некоторое количество $\boldsymbol{m_i} \in \boldsymbol{M}$, причем объем $\boldsymbol{M}$ в долгосрочном периоде стремится к бесконечности, в связи с тем, что управляющий волен использовать новые или применять абсолютно любые действующие $\boldsymbol{m_i}$ и их бесконечные комбинации, в своем стремлении найти наилучший, оптимальный. Однако, следует заметить, что в момент $\boldsymbol{T_0}$ инвестор располагает лишь ограниченным количеством методов определенным множеством M.

Оценка всего множества доступных управляющему (*инвестору*) комбинаций портфелей методов $\boldsymbol{M}$ производится на основании показателей ожидаемой эффективности $\boldsymbol{R_i}$ и риска $\boldsymbol{V_m}$ связанных с применением каждого $\boldsymbol{m_i}$. Оценить эти показатели возможно только по фактическим историческим результатам применения $\boldsymbol{m_i}$ в виде динамики размера управляемого по ним $\boldsymbol{K}$. Соответственно, показателем эффективности набора методов $\boldsymbol{M}$ выступит коэффициент являющийся математическим ожиданием эффективности метода:

$$Mx_{R_i} = R_i \,, \tag{1.4.}$$

где R_i - эффективность j -ого решения, принятого с i -м методом,

N_i - количество принятых решений при использовании i -ого метода;

Дисперсия эффективности измеряется:

$$D_x(R_i) = \sum_{j=1}^{N_i} \frac{(r_{ij} - Mx_{R_i})^2}{N_i} \tag{1.5}$$

Следует отметить, что все методы, имеющие отрицательное математическое ожидание эффективности, согласно условию (1.3), должны сразу же исключаться управляющим из числа используемых в момент T_0, ведь рациональный управляющий не склонен использовать заведомо проигрышный вариант, тем не менее в дальнейшем управляющий может снова протестировать данный m_i на работоспособность, в связи с тем, что математическое ожидание эффективности, в условиях динамики прогнозируемой величины может иметь динамический характер.

Между тем, в качестве показателя, оценивающего риск m_i, как дисперсию использован быть не может в связи с тем, что в применении к динамике K стандартное отклонение будет показывать среднее значение как отрицательных, так и положительных отклонений размера K, в то время как реальным риском, принимаемым управляющим (инвестором) являются только отрицательные отклонения. Но и использовать среднюю величину отрицательных отклонений также будет некорректно, в виду того, что при невысоких усредненных показателях сохраняется вероятность понести значительные по величине разовые потери. Например, при среднем отрицательном отклонении K за год –10%, в начале периода могло произойти отклонение на –50%, после чего величина K не менялась до конца года, принеся управляющему (инвестору) потерю половины первоначальных средств K.

Таким образом, в качестве показателя риска целесообразно взять показатель Value-at-Risk (VaR). Данная методика подробно рассмотрена в работе [2], и широко распространена при нормировании банковских резервов центральными банками, инвестиционными фондами, рейтинговыми агентствами и т.п.

Данная величина характеризует максимальный ожидаемый размер снижения K, на определенном временном интервале T, при использовании стратегии m_i, с опредешленным уровнем доверия равным $1-\alpha$ т.е.:

$$P(|R_i| \leq VaR) = 1-\alpha \tag{1.6.}$$

Отметим, что R_i является случайной величиной и соответственно характеризуется некоторым вероятностным распределением $f(R)$. Значение VaR определяется из распределения приращений следующим образом:

$$1-\alpha = \int_{-\infty}^{-VaR} f(R)dR \tag{1.7.}$$

Таким образом, задача сводится к нахождению функции распределения случайной величины, которая находится методом исторического моделирования или методом Монте-Карло. В случае нормально распределенной R_i нахождение VaR, сводится к нахождению квантиля $z_{1-\alpha}$:

$$V_i = VaR_{(T,1-\alpha)} = z_{1-\alpha} * K * \sigma_{R_i} * \sqrt{T} \,, \tag{1.8}$$

Где σ_{R_i} - Среднеквадратической отклонение случайной величины R_i;

Учитывая динамический характер прогнозируемых величин и неоднородность допущений лежащих в основе концепций любой m_i, существует вероятность неустойчивости практических результатов применения каждого отдельного m_i в настоящий период времени. Таким образом, целесообразней будет пользоваться сразу несколькими m_i, или по-другому сформировать из m_i портфель M_p для обеспечения более устойчивых результатов. Совокупная эффективность управляемого портфелем M_p, капитала K, в этом случае будет определяться как:

$$E_p = \sum_{i-1}^{N_p} w_i R_i \qquad (1.9.)$$

где N_p - количество методов в портфеле M_p, w_i - доля i -ого метода при принятии решения такое что:

$$\sum_{i=1}^{N_p} w_i = 1 \qquad (1.10)$$

Теперь определим совокупный показатель риска M_p. Логично, что в указанных условиях, инвестор будет стремиться сформировать портфель абсолютно независимых друг от друга m_i. Ведь если в условиях данной рыночной конъюнктуры один m_1 покажет низкую эффективность, то и портфель схожих с ним по содержанию m_2 сделает то же самое. При этом необходимо ввести следующее существенное условие для доли каждого используемого метода в портфеле:

$$w_i > 0 \qquad (1.11)$$

Ведь нулевая доля m_i, будет означает то, что метод попросту не используется, а отрицательная – то, что должен использоваться наоборот, что по сути является уже абсолютно другим m_i, который должен быть рассмотрен отдельно на предмет включения в качестве положительной доли M_p.

С целью подбора независимых m_i, необходимо рассчитывать совокупные корреляционные связи кривых эффективности R_i управляемых по каждому конкретному известному m_i, с целью нахождения таких комбинаций, совокупный коэффициент корреляции которых будет стремиться к 0:

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \to 0 \qquad (1.12)$$

где σ_{xy} -ковариация эффективности двух выбранных методов;

Множество выбранных полностью независимых методов m_i, входящих в портфель, при обеспечении положительной ожидаемой эффективности метода позволит теоретически решить проблему устойчивости принимаемого решения в условиях неопределенности, позволяя при $N_p \to \infty$ активно управлять ресурсами K, независимо от риска падения эффективности отдельного метода.

Таким образом, существует возможность получить множество комбинированных активных стратегий управления M_p, разброс совокупных результатов которых характеризуется показателем дисперсии кривой K:

$$V_p^2 = VaR_p^2 = \sum_{i=1}^{N_p} w_i^2 V_i^2 + 2 \sum_{i=1}^{N_p - 1} \sum_{j=i+1}^{N_p} \rho_{ij} w_i V_i w_j V_j \qquad (1.13)$$

Учитывая выражение 1.12 получаем:

$$VaR_p^2 = V_p^2 = \sum_{i=1}^{N_p} w_i^2 V_i^2 \qquad (1.14)$$

Определив понятия эффективности и риска рассматриваемого портфеля методов прогнозирования и установив ограничения, аналогично модели Марковица, подробно рассмотренной в предыдущей главе, а также в работах [**2, 4**], перейдем к построению оптимизационной модели:

$$\begin{cases} E_p = \sum_{i-1}^{N_p} w_i R_i \to \max \\ V_p^2 = \sum_{i=1}^{N_p} w_i^2 V_i^2 + 2 \sum_{i=1}^{N_p - 1} \sum_{j=i+1}^{N_p} \rho_{ij} w_i V_i w_j V_j = const \\ w_i > 0 \\ \sum_{i=1}^{N_p} w_i = 1 \end{cases} \qquad (1.15)$$

Поставленная в выражении (1.15) задача, при подборе оптимальных методов прогнозирования будет решаться многократно, в связи с чем, в качестве метода оптимизации был выбран генетический алгоритм, который удобен в реализации в качестве программного модуля.

В работе [5] произведена формализация описания объектов манипулирования использованных в генетических алгоритмах, поэтому, прежде чем построить поисковый алгоритм нужно определиться с символьной моделью задачи, которая включает в себя:

1. пространство потенциальных решений O - **множество особей**,
2. пространство представлений S – **множество наборов хромосом**,
3. функцию кодирования e и декодирования e^{-1},
4. функцию цели $f: O \to R$ – **функцию оценки приспособленности особи**, такая, что $\forall o_1, o_2 \in O$, если $f(o_1) > f(o_2)$, то o_1 считается решением лучше, чем o_2, R - множество вещественных чисел.
5. функцию оценки представлений $\varphi: S \to R$ – **функцию оценки приспособленности хромосомы**.
6. система ограничений.

Как было сказано ранее генетический алгоритм оперирует особями, которые представляют собой потенциальное решение поставленной нами задачи. В нашем случае решение задачи это вектор весов:

$$W_j = \{w_1, w_2..w_i..w_{N_p}\} \tag{1.16}$$

где w_i - вес i-ого метода в j-ом портфеле (решении) при принятии решения, соответствующий требованиям (1.10) и (1.11) описанными ранее, N_p - количество методов в портфеле.

При реализации алгоритма удобней пользоваться набором хромосом, представленных в двоичном виде вектора:

$$s_j = \{x_1, x_2..x_i..x_{Np}\} \tag{1.17}$$

где x_i - вес i-ого метода в j-ом портфеле (решении) при принятии решения представленный в двоичном виде.

Генетический алгоритм за один шаг производит обработку некоторой популяции особей. Популяция $G(t)$ представляет собой конечный набор векторов (особей):

$$G(t) = \{s_1^t, s_2^t..s_{RP}^t\} \tag{1.18}$$

где, RP - целое число, определяющее размер популяции, а t - номер поколения.

Суть генетического алгоритма заключается в следующем. Пусть на шаге t имеется популяция $G(t)$, состоящая из RP-строк. Для популяции вводится понятие средней ценности популяции $F_{cp}(G(t))$:

$$F_{cp}(G(t)) = \frac{1}{RP}\sum_{j=1}^{RP} f(e^{-1}(s_j^t)) \tag{1.19}$$

Генетический алгоритм переходит от популяции $G(t)$ к популяции $G(t+1)$ таким образом, чтобы средняя численность составляющих ее строк увеличивалась, причем количество новых строк в популяции равно $SO * RP$, где SO - коэффициент новизны (степень обновления популяции). Если $SO < 1$, то популяция будет неперекрывающейся, т.е. в новой популяции сохранятся некоторые строки из старой, а если $SO = 1$, то она будет не перекрывающейся, т.е. подвергнется полному обновлению

Основная особенность генетического алгоритма состоит в том, что анализируется не одно решение, а некоторое подмножество квазиоптимальных решений, называемых хромосомами и состоящих из генов. Это подмножество носит название «популяция». В данной реализации начальная популяция формируется случайным образом. Для рассматриваемой задачи набор хромосом s_j, j-ой особи описывает процентную вес i-ого метода в портфеле, при принятии решении состоящего из N_p - методов, поэтому имеет дискретную структуру. С учетом поставленной задачи (1.15) для каждой j-ой особи, из набора хромосом (портфеля методов) вычисляется целевая функция $\varphi(s_j)$ называемая fitness-функцией:

$$\varphi(s_j) = \sum_{i=1}^{N_p} R_i x_i \text{ ,} \tag{1.20}$$

где x_i - вес, а R_i -ожидаемая эффективность i -ого метода в портфеле при принятии решения.

Такая функция получает на вход набор двоичных хромосом s и возвращает число, показывающее, приспособленность каждой особи. В данном случае значение целевой функции характеризует собой эффективность методов в портфеле. А j -ая хромосома в наборе хромосом особи представляет собой двоичное представление веса i -ого метода в решении (портфеле). Ген в хромосоме представлен 0 и 1. Любое решение, закодированное в наборе хромосом, должно удовлетворять имеющимся ограничениям. Поэтому все вновь созданные особи проверяются на допустимость применения в качестве решения.

Ограничения, описанные в (1.15) реализуются в виде оператора оптимизации целенаправленно подбирающего оптимальную точку кроссовера, формирования новых особей, с учетом ограничений и максимизации функции приспособленности. Данный алгоритм можно назвать гибридным т.к. в нем целенаправленно производится модификация хромосом.

Таким образом, для гибридного генетического алгоритма выделяется четыре основных этапов: 1) формирование начальной популяции 2) селекция родительских пар 3) оптимизация выбора точки кроссовера и формирование новых хромосом 4) мутация 6) редукция.

Общий принцип работы генетического алгоритма соответствует алгоритму, разработанному ранее и описанному в работе [5]. Алгоритм показал свою эффективность при подборе порядков скользящих средних в методе прогнозирования основанного на пересечении трех разно-порядковых простых скользящих средних.

Список литературы

1. Ле Бо Ч., Лукас Д. В. Компьютерный анализ фьючерсных рынков: Пер. с англ. - М.: Издательский Дом "АЛЬПИНА", 1998- 304с.

2. Финансовый менеджмент. – М.: Перспектива, 1999.-268с

3. Вороновский Г. К. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности – Х:ОСНОВА, 1997. – 112с.

4. Markovitz H. M. Portfolio selection. Journal of Finance. – 1952. vol.7, №1.

5. Портнов К.В. Генетические алгоритмы и поиск эффективных порядков индикаторов в Биржевой торговой стратегии на основе пересечения трех скользящих средних. – Вестник СамГТУ – 2005 - № с. 72– 76.

Концепция конструирования и реализации диагностической системы в клинической иммунологии на основе экспертных знаний

Примаков Д. А., *primada@velnet.ru*
к.т.н., доцент каф. Высшей математики,
Московской финансово–промышленной академии,
Можайское шоссе,4, корп. 1, кв. 404,

Спецификой задачи принятия решения в медицине является слабая формализация систем принятия решений, большой удельный вес неточных знаний и наличие качественно измеряемых показателей. В связи с ограниченностью возможности применения традиционных методов оптимизации и машинного моделирования возникает необходимость в новых методах, с помощью которых можно анализировать качественную информацию на ЭВМ и принимать решения на плохо формализуемых этапах. Основой такого подхода может служить математическая модель принятия решений квалифицированным экспертом, которая базируется на его профессиональных знаниях, как о закономерностях предметной области, так и о методах принятия решений на основе этих знаний. На этом принципе основаны экспертные системы (ЭС).

Медицинские знания – это накопившийся за долгое время опыт человечества в области диагностики заболеваний и возможных способов их лечения. Этот опыт сконцентрирован и в то же время рассеян во многих тысячах медицинских книг, научных статьях, справочниках и практических руководствах.

Современный поток медицинской информации, определяющий, в конечном счете, знания врача, крайне велик и удваивается по некоторым оценкам каждые три года.

Как оперировать таким объёмом информации? Как каждый раз получить доступ к той её части, которая необходима в данном конкретном случае? По-видимому, современные ЭВМ с их способностью хранить и обрабатывать огромные массивы информации могут оказать тут некоторую помощь в систематизации информации и облегчении доступа к ней. Однако если эта деятельность направлена лишь на создание огромного компьютерного «Медицинского справочника», то большой практической пользы она не принесёт. Мало обладать знаниями, нужно уметь ими рационально пользоваться. Ведь опытный врач при осмотре пациента очень быстро, отбрасывая всё ненужное, использует ту часть своих знаний, которая необходима в данном конкретном случае. Другая причина, которая обуславливает целесообразность использования ЭВМ в медицине, связана с тем фактом, что резкое увеличение количества информации и углубление знаний по конкретным направлениям медицины приводят к появлению узкоспециализированных врачей. Они способны оказать помощь больному, в основном лишь по своей специальности, и то только в том случае, если специалист и больной «нашли» друг друга.

Как избавить больного от ненужного и изнурительного поиска путём «полного перебора» специалистов, на который часто уходят годы, годы, в течение которых болезнь прогрессирует?

Действительно, трудно найти врача, который был бы высококвалифицированным специалистом одновременно во всех нужных в данном случае областях медицинского знания. В то же время если бы мы сумели поместить в ЭВМ знания всех лучших специалистов, необходимые для постановки диагноза данного заболевания, то мы получили бы программу, соответствующую консилиуму лучших специалистов в данной области. Это стремление дать с помощью ЭВМ знания лучших специалистов всем врачам также послужило серьёзным стимулом к развитию диагностических медицинских систем.

Сегодня создаются средства, которые позволяют хранить и накапливать в ЭВМ наши знания о понятиях и явлениях окружающего мира и использовать их для решения задач качественного характера. Именно такие задачи наиболее характерны для медицины, где большая часть знаний врачей выражается вербально. Появление компьютеров и программного обеспечения, ориентированных на обработку нечисловой информации, также стимулировало применение ЭВМ в медицине.

Данная статья посвящена обсуждению концепции разработки ЭС. «МетаЭксп» в практической медицине, а в частности, в клинической иммунологии.

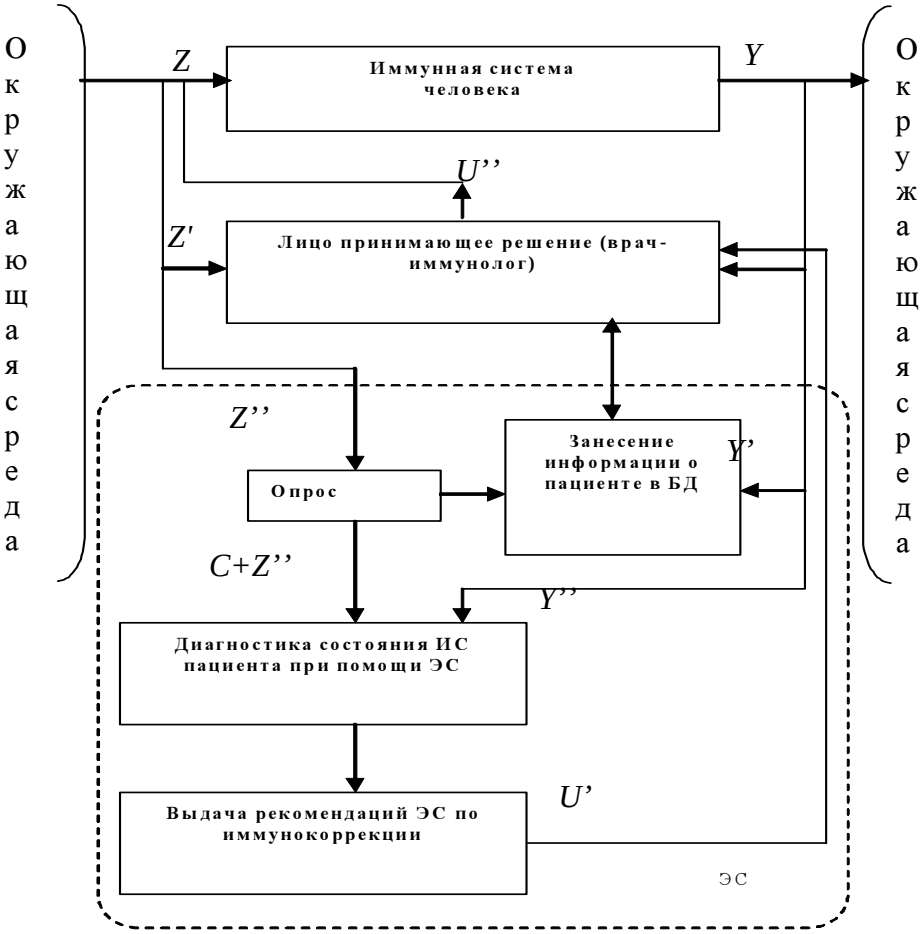


Рисунок 1 – Структурная схема системы принятия решений

Иммунная система (ИС) человека - это сложная многокомпонентная система, обеспечивающая контроль и защиту организма человека, функционирующего в стохастической среде. Предположим, что на вход ИС поступают возмущения *Z*, которые имеют вид n-мерного пространства R_z^n. Воздействия, которые наблюдаются системой управления, обозначим как $Z'' \in R_z^n$, а воздействия, которые наблюдаются лицом, принимающим решение (ЛПР), обозначим как $Z' \in R_z^n$ (см. рисунок 1). Размерность Z'' и Z' зависит от количества измеряемых переменных, которые характеризуют состояние окружающей среды. Пусть R_z^m – m-мерное пространство, тогда $Y'' \in R_y^m$ и $Y' \in R_y^m$, где Y'' и Y' - представляют собой конечные наборы выходных переменных *Y*, которые описывают реакцию ИС на внешние воздействия *Z*, и которые наблюдаются системой управления и ЛПР, соответственно. Рассмотрим ИС как объект управления, Под управлением будем понимать процесс изменения (стимуляция, угнетения, замещения) свойств ИС в сторону обновления ёё гомеостаза под влиянием иммунокорригирующих препаратов, то есть путем изменения параметров ИС в условиях воздействия *Z* для достижения некоторого стабильно–адаптированного состояния.

Пусть $U = \langle U', U'', U''' \rangle$ - вектор управления, компонентами которого являются: U' - рекомендуемый для ЛПР вариант управляющего воздействия системы управления; U'' - управляющее воздействие, которое подается на ИС непосредственно со стороны ЛПР. Известно, что ИС это система, которая регулирует свои свойства, поэтому существует внутреннее управление ИС U''', которое учитывает воздействие различных гормонов и медиаторов, вырабатывающиеся самим организмом и регулирующие интенсивность процессов иммунного ответа и функционального восстановления. Задача ЛПР состоит в оценке состояния среды и объекта управления, формулировке цели управления и принятии решения о выдаче управляющих воздействий U''.

Выработка рекомендаций U' осуществляется с помощью экспертной системы на основании эвристического анализа полученной информации о состоянии здоровья пациента в целом. Общая информация о состоянии здоровья пациента представляется в виде набора $C = \{c_1, c_2, ..., c_l\}$ качественных (клинико-анамнестических) показателей. Информация *C* добывается во время компьютеризированного анкетного опроса пациента и характеризует: характер и стадию основного заболевания, характер заболеваний, которые сопровождают основное заболевание пациента, перенесенные в течение жизни инфекционные заболевания, наследственную отягощенность пациента, условия проживания и работы и т.д.

Таким образом, задача выработки управляющего воздействия U' сводится к определению наилучшего скооперированного управления, полученного в результате синтеза управлений от ЭС, основанных на базе знаний, и от врача–иммунолога, основанных на врачебном опыте.

Разработанный в среде программирования Borland Delphi программный комплекс «МетаЭксп», используется как система поддержки принятия решений (ПР) в иммунологических исследованиях и состоит из двух функциональных частей:

– Оболочки ЭС;
– Редактора базы знаний.

ЭС служит для проведения эвристического анализа клинико-анамнестических признаков с целью оценки состояния иммунной системы человека и выработки рекомендаций по иммунокоррекции и восстановлению функций ИС. Логический вывод заключений осуществляется путем обратного вывода на динамической сети продукционных правил с элементами нечеткой логики.

Разработанная ЭС принадлежит к классу продукционных систем, которые определяются тройкой <БД, БЗ, МВ>, где БД – база данных, БЗ – база знаний, МВ – машина вывода.

БЗ, которая используется в процессе вывода, хранится отдельно от ЭС, и в общем случае, может описывать практически любую предметную область. Редактор БЗ, входящий в созданный программный комплекс, предназначен для создания новых и изменения существующих БЗ.

Для формата хранения знаний в БЗ был выбран язык XML, обладающий рядом достоинств:

– XML позволяет разработчику создать собственную разметку структуры хранимой информации;
– Разбор XML хорошо стандартизирован и реализован большим количеством производителей ПО, что позволяет извлечь информацию из XML-документов практически повсеместно;
– В стандарт XML включена поддержка кодовых страниц Unicode, что упрощает создание многоязычных документов;
– Приложения могут использовать XML-парсеры для проверки структуры документов, а при использовании схем – и типов данных. Это может значительно упростить разбор строго структурированных документов, снимая с программиста задачу проверки правильности документа;

– XML – это текстовый формат, то есть читаемый, легко документируемый и, иногда, более простой в отладке при условии, что сегодня отладочные средства для ряда XML-технологий пребывают в зачаточном состоянии;

– Для работы с XML создано множество средств на самых разных платформах, что делает использование XML более простым по сравнению с бинарными форматами при обмене сложными информационными потоками;

– XML-документы могут использовать значительную часть инфраструктуры, созданной для HTML, включая протокол HTTP и браузеры.

Теоретически XML может использоваться в любых приложениях, которым нужна структурированная информация – от сложных систем с гигантскими объемами передаваемой информации до обычных «однокомпьютерных» программ, использующих этот язык для описания или хранения различной информации.

Традиционно хранение БЗ осуществлялось в базах данных, и выбор в пользу XML в данной работе был сделан в силу следующих преимуществ XML над БД:

– иерархическое (объектно-ориентированное) представление данных;

– текстовое представление информации.

Структура XML файла состоит из двух частей: описания объектов и описания правил, использующих определенные объекты.

Описание объектов состоит из последовательного определения свойств каждого объекта. Для каждого объекта приводиться список предопределенных значений. Также возможно задание зависимостей между объектами.

Описание каждого правила содержит посылку и заключение. Посылка состоит из имени сравниваемого объекта, отношения и значения, с которым происходит сравнение. Максимальное количество посылок теоретически неограниченно, однако необходимо наличие как минимум одной посылки. Заключение состоит из имени определяемого объекта, устанавливаемого значения и степени доверия к данному заключению. Максимальное количество заключений теоретически неограниченно, однако необходимо наличие как минимум одного заключения.

Создание и редактирование БЗ необходимо проводить только с использованием редактора БЗ, иначе работа с БЗ будет невозможной.

Редактор позволяет поэтапно провести эволюционный процесс создания действительно полной и непротиворечивой БЗ. Продемонстрируем по шагам процесс создания новой БЗ.

Предназначение данной части созданного программного комплекса «МетаЭксп» – проведение экспертизы в указанной предметной области в соответствии с выбранной БЗ, созданной ранее в редакторе. При реализации проекта создана БЗ врача-иммунолога по определению иммунодефицитных и иммунопатологических состояний пациента и разработана оболочка ЭС, которая может быть использована во врачебной практике иммунолога-клинициста.

Оптимизация поэтапного распределения средств в слабо структурированных задачах

Пучков А. Ю., *кандидат технических наук, доцент.*
Окунев Б. В., *кандидат технических наук, доцент.*
Государственное образовательное учреждение высшего профессионального образования «Московский энергетический институт (ТУ)» филиал в г.Смоленске, кафедра Менеджмента
г. Смоленск 214000, Энергетический проезд, д.1, (48122) 354991.
E-mail: feu@sci.smolensk.ru (указать для Пучкова А.Ю.)

В работе предлагается модификация метода динамического программирования для условий нечеткого задания исходной информации о функциях дохода и остатка. Представлен пример решения типовой задачи распределения денежных средств с использованием рассматриваемого подхода.

Данная модификации метода динамического программирования может найти применение в областях, связанных с решением задач многошаговой оптимизации в условиях нечеткости исходной информации.

Ключевые слова: динамическое программирование, оптимизация многошаговых операций, нечеткая логика, принятие решений.

Принятие управленческих решений в экономике часто сопряжено с проблемой нечеткости и неполноты исходной информации о предметной области. В тоже время менеджеру высшего или среднего звена необходимо и в этих условиях делать взвешенные решения. При такой постановке задачи целесообразно применять методы искусственного интеллекта, в частности нечеткую логику, в сочетании с известными математическими алгоритмами.

Одним из таких алгоритмов является динамическое планирование, известное также как динамическое программирование, давно и успешно используется для решения оптимизационных задач в технике, экономике и других областях человеческой деятельности. Динамическое программирование представляет собой специальный математический аппарат оптимизации решений, специально приспособленный к многошаговым (или многоэтапным) операциям.

В экономических приложениях, например, задачах распределения ресурсов, с этапами часто ассоциируют временные интервалы - месяц, квартал, год и так далее. Рассматривая, далее, именно этот частный вид задач, отметим, что в качестве исходной информации задаются функции дохода от вложенных средств на каждом шаге и их остатка в конце этого шага. Обычно доход и остаток носят вид функциональных зависимостей от количества средств в начале этапа и эти зависимости считаются известными и четко определенными. Но отличительной чертой подавляющего большинства экономических объектов являются их сложность, неполнота и нечеткость информации об объекте, наличие не только количественных, но и качественных характеристик. Эти обстоятельства приводят к тому, что получение функциональных зависимостей для упомянутых дохода и остатка является не простой задачей и требуют применение специальных методов, в частности, статистических. Использование статистических методов, в свою очередь, сталкивается с проблемой нехватки данных в силу ограниченности числа исследуемых экономических объектов.

Более приемлемым подходом для описания дохода и остатка может служить применение экспертных оценок, которые выражаются в лингвистической форме, например, доход и остаток будет характеризоваться как «большой», «средний», «малый», «очень малый» и так далее. Ставится задача – каким образом учесть эти качественные характеристики при решении задач динамического программирования?

Предлагаемый подход решения вышеописанной задачи базируется на теории нечеткой логики (fuzzy logic), позволяющей принимать решения в условиях неполной и размытой исходной информации об объекте [1]. Порядок решения задачи динамического программирования в данном случае остается классическим [2], разница заключается лишь в форме задания функциональных зависимостей дохода и остатка.

Проиллюстрируем предлагаемый подход на примере решения типовой задачи распределения средств, использовав в качестве среды реализации алгоритма решения пакет Fuzzy Logic Toolbox системы MatLAB.

Постановка задачи формулируется следующим образом. Пусть имеется система из двух предприятий и начальное количество средств K_0=900 денежных единиц (ден. ед.), которые надо распределить в течении 3 лет между предприятием I и предприятием II. Средства x, вложенные в предприятие I приносят доход F1(x), средства y, вложенные в предприятие II, приносят доход F2(y). Поступление средств к началу каждого очередного года на первом и втором предприятии составляют G1(x) и G2(y) соответственно. Здесь термин «остаток» заменен на более общий «поступление», отражая тот факт, что к концу каждого года средства образуются, в зависимости от их количества в начале года, различным путями, не конкретизируя какими именно.

По истечении одного периода, поступившие от предприятий средства суммируются и заново перераспределяются между предприятиями. В производство вкладываются все поступившие средства, то есть y = K - x. В отличие от классической постановки задачи зависимости F1(K), F2(K), G1(K), G2(K) представляют собой базы знаний, состоящие из правил (П), разработанных экспертами.

Универсальное множество денежных средств сформируем с учетом экспертной оценки, например, в 2000 ден. ед. за рассматриваемый период. Обозначим нечеткие множества A1 – «средств мало», A2 – «средств средне», A3 – «средств много», B1 – «доход малый», B2 – «доход средний», B3 – «доход большой», B4 – «доход очень большой», C1 –«поступление мало», C2 –«поступление среднее», C3 –« поступление большое», C4 – « поступление очень большое». Данные нечеткие множества определены на универсальном множестве денежных средств с помощью соответствующих функций принадлежности. Для примера, на рисунке 1 представлен вид функций принадлежности выходной лингвистической переменной «доход». Следует отметить, что нечеткое множество B1 - «доход малый» (на рисунке 1 обозначено «malo») имеет симметричную относительно нуля функцию принадлежности. Если этого не предусмотреть и оставить только положительную область значений средств, то при использовании некоторых алгоритмов нечеткого вывода, например, Mamdani, возникает ситуация, когда при нулевых средствах получается ненулевой доход, чего быть не должно.

Обозначим z – доход, s – поступление и зададим базы знаний в следующем виде: F1(x): П1: Если x есть A1, то z есть B1;

П2: Если x есть A2, то z есть B2;

П3: Если x есть A3, то z есть B3;
F2(y): П1: Если y есть A1, то z есть B2;
П2: Если y есть A2, то z есть B3;
П3: Если y есть A3, то z есть B4;
G1(x):): П1: Если x есть A1, s есть C1;
П2: Если x есть A2, то s есть C2;
П3: Если x есть A3, то s есть C3;
G2(y): П1: Если y есть A1, то s есть C2;
П2: Если y есть A2, то s есть C3;
П3: Если y есть A3, то s есть C4;

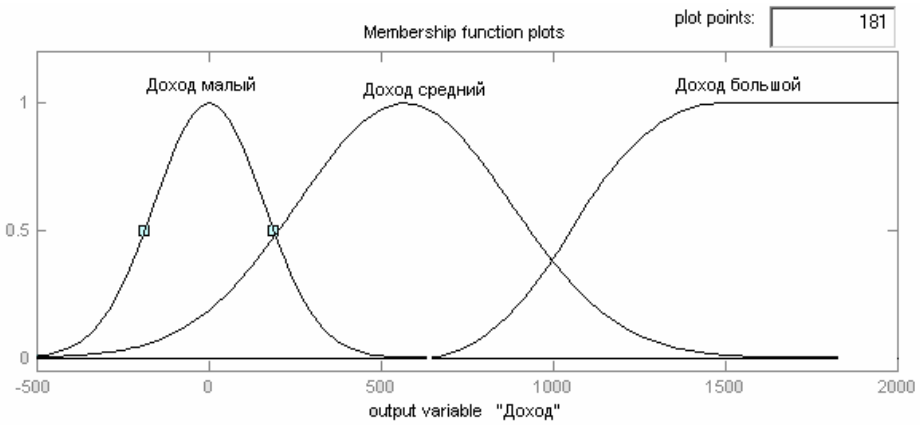


Рисунок 1 – функции принадлежности лингвистической переменной «Доход»

Отметим, что наборы лингвистических переменных, правил, вид функций принадлежности могут варьироваться в зависимости от сложности задачи и поставленных целей.

Управлением на i-м шаге будут количества средств x_i , y_i вкладываемые в соответствующие предприятия. Управление всем распределением средств U состоит из совокупности шаговых управлений $U = (U_1 ; U_2 ; U_3)$. Требуется найти оптимальное управление, при котором суммарный доход W приносимый обоими предприятиями за три года был бы максимальным.

Доход на i-м шаге определяется выражением $w_i = F1(x_i) + F2(K- x_i)$, где K – количество средств в начале i-го года.

Под влиянием управления x_i (вложение x_i средств в предприятие I и вложение $y_i = K- x_i$ средств в предприятие II) система перейдет из состояния K в состояние $K` = G1(x_i) +G2(K- x_i)$.

Основное функциональное уравнение принимает вид:

$$W_i(K) = \max\{ w_i + W_{i+1}(K`)\}, \qquad (1)$$

где максимум (max) определяется в диапазоне $0 \le x_i \le K$, $W_{i+1}(K`)$ – условное оптимальное значение дохода на всех последующих шагах.

Условное оптимальное управление на i-м шаге будет то, при котором достигается максимум (1).

Решение задачи, как обычно, начинаем с определения условного оптимального дохода на последнем шаге (начало третьего года):

$$W_3(K) = \max\{ F1(x_3)+F2(K- x_3)\}.$$

Оптимизация на последнем шаге не рассматривает последующие шаги, поэтому значение W_4 , требуемое в соответствии с (1), в данном случае равно нулю.

Для нахождения максимума W_3 (K) следует просчитать значение дохода W_3 при различном количестве средств и при различном распределении их между двумя предприятиями. Проводя промежуточные нечеткие выводы в среде MatLAB, получаем значения, представленные в таблице 1.

Из таблицы 1 находим максимальные значения дохода для последнего шага при различной величине K и соответствующие условные оптимальные управления (максимальные значения дохода и соответствующие управления подчеркнуты). Далее следует построить интерполирующий полином.

Вместо полинома может использоваться и другая зависимость, например нечеткая аппроксимация.

Переходим к планированию предпоследнего года. В соответствии с (1) условный оптимальный доход на этом шаге (начало второго года):

$$W_2(K) = \max\{ F1(x_2)+F2(K- x_2)+ WP_3(G1(x_2)+G2(K- x_2))\}.$$

Результаты расчетов сведем в таблицу 2. Отметим, что на предпоследнем шаге уже необходимо учитывать условное оптимальное управление на последнем шаге W_3.

При определении поступления средств к концу года может оказаться так, что полученное количество средств не представлено в таблице 1. В этом случае следует просчитывать таблицы с более мелким шагом.

Таблица 1

K = 450				
x_3	450	300	150	0
y_3	0	150	300	450
W_3	620	1148	1132	672
K = 900				
x_3	900	600	300	0
y_3	0	300	600	900
W_3	853	1410	1392	1100
K = 1350				
x_3	1350	900	450	0
y_3	0	450	900	1350
W_3	1500	1525	1730	1670
K = 1800				
x_3	1800	1200	600	0
y_3	0	600	1200	1800
W_3	1610	2132	2360	1680

В данном иллюстративном примере используется наиболее простая аппроксимация полиномом нулевой степени.

Далее из таблицы 2 находим максимальные значения дохода для предпоследнего шага при различной величине К и соответствующие условные оптимальные управления и находим полином $WP_2(K)$.

Таблица 2

K = 450				
x_2	450	300	150	0
y_2	0	150	300	450
W_2	1768	2558	2542	1820
K = 900				
x_2	900	600	300	0
y_2	0	300	600	900
W_2	2001	2820	3122	2510
K = 1350				
x_2	1350	900	450	0
y_2	0	450	900	1350
W_2	3230	3255	3460	1670
K = 1800				
x_2	1800	1200	600	0
y_2	0	600	1200	1800
W_2	3970	4492	4720	4040

Переходим к планированию первого года. На начало первого года известно количество средств К=900, поэтому здесь выбора нет: $W_1(900)= F1(x_1)+F2(900 - x_1)+ WP_2(G1(x_1)+G2(900- x_1))$.

Составляем таблицу 3. Из нее находим оптимальное управление на первом году: x_1 = 600 ден. ед., y_1 = 300 ден. ед.

Таблица 3

K = 900				
x_2	900	600	300	0
y_2	0	300	600	900
W_2	3213	3770	3701	3460

После окончания процесса построения условных оптимальных управлений проводим вторую стадию оптимизации, переходя от первого шага к последнему.

Зная x_1 и y_1 находим поступление средств к концу первого года G1(600)+G2(300)=487+521=1008 ден. ед.

Так как использовался полином нулевого порядка, то оптимальное управление на втором году можно найти прямо из таблицы 2: x_2 = 300 ден. ед., y_2 = 600 ден. ед. Учитывая это количество средств, к концу второго года поступление средств составит G1(300)+G2(600)=305+822=1127 ден. ед.

Из таблицы 1 находим оптимальное управление на третьем году: x_3 = 600 ден. ед., y_3 = 300 ден. ед.

Завершая процесс решения, составим план распределения средств по предприятиям, представленный в таблице 4.

Таблица 4

	1 год	2 год	3 год
Предприятие I	600	300	600
Предприятие I I	300	600	300

Прибыль за три года составит:
Wmax=F1(600)+F2(300)+F1(300)+F2(600)+F1(600)+F2(300)=770+640+510+882+770+640 = 4212 ден. ед.

Проведенное решение примера показывает достаточную громоздкость вычислений, поэтому целесообразно разработать соответствующее программное обеспечение.

Рассмотренная модификация метода динамического программирования может найти применение в различных областях, связанных с решением задач многошаговой оптимизации в условиях нечеткости исходной информации.

Список литературы

1. Андрейчиков А. В., Андрейчикова О. Н. Анализ, синтез, планирование решений в экономике. - М.: Финансы и статистика, 2000.

2. Конюховский П. В. Математические методы исследования операций в экономике – СПб.: Изд-во «Питер», 2000.

Проектирование экспертных систем технической поддержки на основе онтологий

Рабчевский Е. А.,
E-mail: evgeny@ranat.ru
Архипов Е. С.,
E-mail: saibro@mail.ru
Пермский государственный университет
кафедра компьютерных систем и телекоммуникаций
Россия 614990 г.Пермь, ул.Букирева, 15

Выявлены понятия, необходимые для осуществления технической поддержки. Создана онтология, предоставляющая соответствующие термины. Реализовано web приложение, использующее технологии Semantic Web, - машина логического вывода с интерфейсом экспертной системы, работающая с указанными понятиями. Работа приложения демонстрируется на технической поддержке компьютера и автомобиля.

Введение

Техническую поддержку многих систем экономически более выгодно оказывать удаленно, в автоматическом режиме, то есть использовать формализованные знания эксперта.. Знания, обеспечивающие техническую поддержку для различных предметных областей, имеют схожую структуру. Для их представления рационально использовать онтологии, и другие технологии Semantic Web.

Пользователь всякой системы технической поддержки, как правило, не знает всех причинно-следственных связей используемой им технической системы, и имеет дело лишь с «симптомами». Задача экспертной системы технической поддержки – по неполной информацию о системе, однозначно определить её состояние и выдать пользователю рекомендации по переводу системы в целевое состояние. Причем интеллектуальность системы зависит от того насколько быстро система «поставит диагноз».

При создании базы знаний экспертной системы технической поддержки предлагается разделить понятия, которыми оперирует экспертная система при принятии решения: состояние поддерживаемой

системы, факторы, влияющие на него и т.п.; и термины конкретной предметной области, которые можно отнести к этим понятиям. Например, «автомобиль» находится в состоянии «не заводится», на это может влиять то, что «аккумулятор» находится в состоянии «разряжен».

Такой подход позволит повторно использовать одну и туже машину логического вывода (уже с интерфейсом) в качестве экспертной системы для поддержки различных предметных областей. Для этого необходимо «разметить» конкретную предметную область в терминах предложенного словаря.

База знаний

В качестве базы знаний экспертной системы предлагается использовать OWL онтологию – компонент технологии Semantic Web. Принцип разделения понятий машины вывода и терминов конкретной предметной области разделяет базу знаний на онтологию логики принятия решений, в которой определяются понятия, с которыми оперирует машина логического вывода, и онтологию предметной области – формальное описание предметной области в терминах онтологии логики принятия решений.

Далее прописными буквами обозначаются ключевые элементы онтологии логики принятия решений.

Пользователь имеет дело с ЭЛЕМЕНТАМИ СИСТЕМЫ или её ПОДСИСТЕМ. Каждый ЭЛЕМЕНТ имеет своё СОСТОЯНИЕ. На состояние элемента влияет набор ФАКТОРОВ. Фактор имеет ряд разрешенных ЗНАЧЕНИЙ. Причем один фактор может влиять на несколько элементов, а на значение самого фактора могут влиять другие факторы. Состояние всякого элемента однозначно определяется значениями всех влияющих на него. В рамках одного элемента каждый фактор имеет ПРИОРИТЕТ. Для реализации отношения элемент-фактор-приоритет предлагается расширить триплетную модель данных OWL. Фактору определено КАК_ОПРЕДЕЛИТЬ_ИЗМЕНИТЬ.

Ниже представлена ER-диаграмма онтологии логики принятия решений, разработанной с помощью редактора онтологий Protégé 3.1.

Уровень логического вывода

Состояние системы определяется состоянием её элементов, пользователь указывает элементы, состояние которых его не устраивает, и определяет целевое состояние. Экспертная система определяет набор всех факторов, способных перевести систему в целевое состояние.

Если пользователь уже определил значения некоторых факторов, то ввиду того, что значение одного фактора может определяться значением другого, из набора всех факторов, способных перевести систему в целевое состояние, возможно, потребуется устранить факторы, запрещенные другими. Таким образом, образуется иерархия факторов.

Интеллектуальная задача системы состоит в том чтобы, определить фактор, изменив значение которого пользователь переведет систему в целевое состояние, при том что пользователю будет задано минимум вопросов (предложено определить значения минимального количества факторов).

Экспертная система выбирает фактор по следующим критериям:

- в иерархии факторов выбирается верхний (влияющий на наибольшее количество других факторов);
- фактор, влияющий на наибольшее количество элементов, состояние которых не устраивает пользователя;
- фактор с наибольшим приоритетом.

Пользователь определяет или изменяет значение предложенного фактора, и если система не переходит в целевое состояние, то по указанным критериям выбирается следующий фактор, и пользователю предлагается определить или изменить его значение.

Изначально полагается, что поддерживаемая система не обладает «памятью», то есть порядок изменения значений факторов не имеет значения.

Предполагается, что в онтологии предметной области указаны все факторы, способные перевести систему в целевое состояние. Это означает, что рано или поздно пользователю будет предложен фактор, изменение значение которого, переведет систему в целевое состояние.

В следствие, кросс-платформености Semantic Web приложений, предлагается использовать платформу Java2. Механизм логического вывода реализуется с помощью Jena Semantic Web Framework - библиотеки Java классов для работы с RDF и OWL онтологиями.

В коде приложения следует разделять классы, обеспечивающие загрузку онтологии в память, классы, которые проводят с онтологией указанные процедуры логического вывода, и классы непосредственно связанные с интерфейсом пользователя. Это позволит развивать систему модульно.

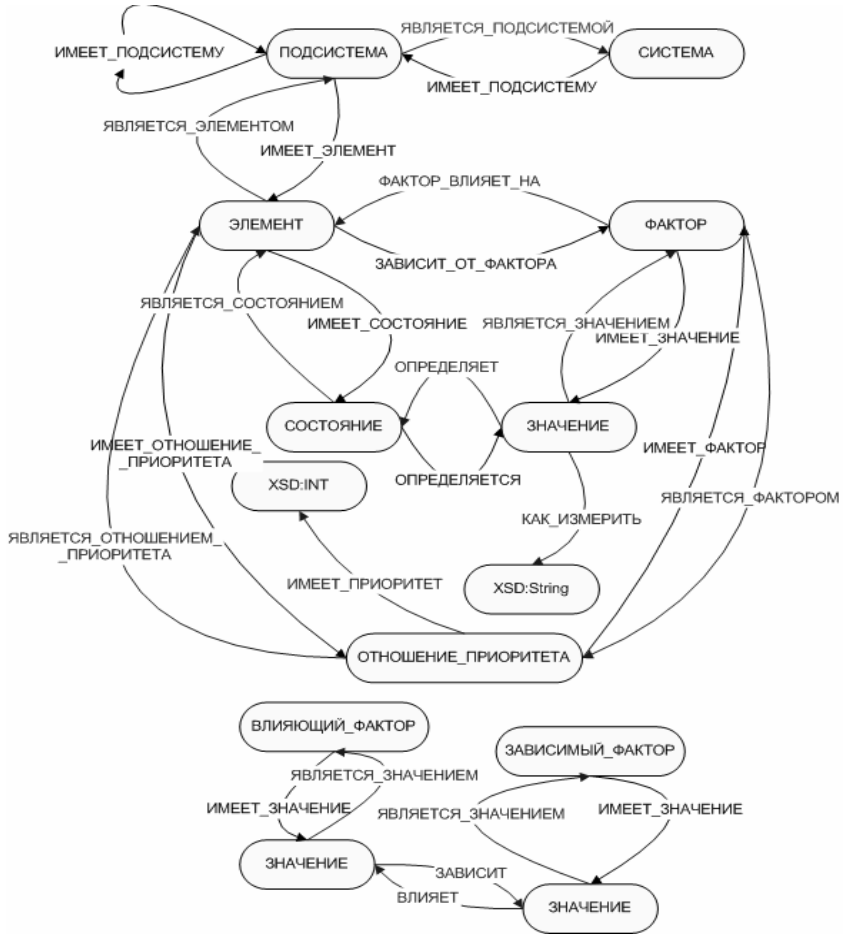
В частности предложенная онтология лишь определяет основные понятия работы эксперта, при этом простота OWL онтологий позволяет ввести, например понятие, обеспечивающее самообучение, лишь на уровне онтологии. Для реализации нескольких моделей вывода, например, еще нечеткого, предлагается ввести RDF-словарь самой экспертной системы с профилем логического вывода.

Интерфейс

Для реализации интерфейса предлагается использовать технологии JSP страниц и сервлетов, компонентов спецификации J2EE, которая позволяет разрабатывать копоративные веб приложения. Таким образом, экспертная система представляет собой веб приложение.

Классы интерфейса пользователя позволяют визуализировать элементы онтологии и осуществлять диалог с пользователем.

Важным моментом в успехе внедрения экспертной системы является осуществление через тот же веб интерфейс таких функций, как администрирование, модификация онтологии предметной области.



Заключение

Предложенные онтология логики принятия решений и машина логического являются одним из немногих российских приложений, использующих технологии Semantic Web. Используя данный подход, созданы онтологии для двух предметных областей: компьютер, автомобиль. Более подробно с соответствующими приложениями можно познакомиться на сайте http://semanticweb.dev.juga.ru.

Авторы благодарят М. А. Марценюка за обсуждение работы.

Список литературы

1. Tim Barners Lee, Semantic Web, SCIENTIFIC AMERICAN 2001.

2. OWL Web Ontology Language Use Cases and Requirements http://www.w3.org/TR/2004/REC-webont-req-20040210/

3. OWL Web Ontology Language Guide http://www.w3.org/TR/2004/REC-owl-guide-20040210/

4. Prot.eg.e home page http://protege.stanford.edu/

5. The Prot.eg.e OWL Plugin: for Semantic Web Applications Holger Knublauch, Ray W. Fergerson, Natalya F. Noy and Mark A. Musen Stanford Medical Informatics, Stanford School of Medicine

6. Ontology-Driven Software Development in the Context of the Semantic Web: An Example Scenario with Prot.eg.e/OWL

7. Defining N-ary Relations on the Semantic Web http://www.w3.org/TR/2006/NOTE-swbp-n-aryRelations-20060412/

8. Jena Semantic Web Framework home page http://jena.sourceforge.net/

9. http://semweb.krasu.ru

10. http://www.ksl.stanford.edu/people/dlm/webont/wineAgent/

Применение методов теории распознавания для анализа петроглифов Карелии[1]

Рогов А. А.,
профессор каф. Математического моделирования систем управления
Петрозаводского государственного университета,
185910, Карелия, г. Петрозаводск, пр. Ленина, д. 33,
rogov@psu.karelia.ru

Петроглифы Карелии представляют собой уникальный археологический памятник 3-5 века до н.э. При знакомстве с ними возникает масса вопросов: Что означали выбитые фигуры? Зачем они создавались? В какой последовательности заполнялись наскальные полотна? Существует ли какая-то взаимосвязь между разновременными фигурами? Возникает множество догадок, предположений, умозаключений. Понять беломорские петроглифы несколько легче по причине распознаваемости основной массы фигур (лодки, звери, люди). Онежские петроглифы оказываются более сложными для понимания. В них много необычных, очень оригинальных тем, сюжетов и деталей. Не удивительно, что они привлекают внимание уфологов, исследователей аномальных явлений (летающие тарелки, пришельцы из космоса), криптозоологии (длинношеие лебеди - динозавры). Разошлись и мнения первооткрывателей [1].

Применение современных информационных технологий и математических методов, в первую очередь теории распознавания, позволяет ответить на ряд вопросов, которые возникают у исследователей. Для получения ответов на эти вопросы необходимо решить следующие задачи:

1. Классификации изображений;
2. Визуализация исторического материала;
3. Поиск инварианта петроглифа, связанного с анализом его поверхности.

Для решения этих задач в первую очередь необходимо создание электронной коллекции изображений петроглифов и разработка поисковой системы по ней. Несмотря на активное создание электронных коллекций графических документов (графических баз данных), в настоящее время не существует универсальных методов поиска информации в этих базах данных. Все существующие методы поиска можно разделить на две группы:

- поиск по признакам;
- поиск по сходству изображений.

К первой, самой многочисленной группе относится поиск по названию или размерам изображения в поисковых системах в Интернете. При поиске в музейных коллекциях часто выделяют, как название конкретных предметов, так и классы к которым они относятся, а также время и место создания. Заметим, что поиск в этих системах осуществляется стандартными методами поиска информации в базах данных, так как каждому графическому изображению соответствует свой файл с текстовой информацией, по которой и осуществляется поиск. При характеризации графического изображения по признакам возникает задача не только выбора признаков, но и их анализа на зависимость. Как правило, для этого применяются статистические методы. В качестве примера можно назвать выделенные признаки сравнения петроглифов лосей и оленей и проверка их на независимость, описанные в работе [2]. В этой же работе рассматривается задача кластеризации охарактеризованных изображений.

Вторая группа методов поиска отличается сложностью алгоритмов и низкой универсальностью. Лучше отработаны методы распознавания текста при сканировании, т. к. каждая буква представляет

[1] Проводимые исследования поддержаны грантом РГНФ № 05-01-12118в (руководитель Н.В. Лобанова).

собой графический символ и задачу распознавания текста можно трактовать, как поиск введенного символа в графической базе данных, состоящей из изображений букв алфавита, цифр и специальных символов. Существуют специальные программы, связанные с поиском и конструированием документов в ГИС и САПР, поиском объектов на картах и т.д.

Рассмотрим особенности поиска по схожести изображений в электронных коллекциях петроглифов, которые представляют собой набор цветных фотографий. Определенную сложность поиска создает то, что у некоторых изображений фактически отсутствует его часть. Наряду с этим, требование, что при поиске необходимо только совпадение контура изображения, позволяет упростить поиск, а значит, изображение петроглифа можно рассматривать, как бинарное (скале соответствует белый цвет, а петроглифу – черный). При организации поиска требуется решение следующих задач:

1. перевод цветных фотографий в черно-белые (сегментация изображений);
2. выделение на фотографии из группы петроглифов отдельного изображения;
3. разработка методов непосредственного поиска искомых изображений.

Для решения первой задачи были опробованы эталонный метод, градиентный метод и методы фрактального анализа [3, 4]. Вторая задача решается легко решается методами вычислительной геометрии [5]. При разработке методов непосредственного поиска искомых изображений нужно учитывать следующие требования:

- возможные изменения размеров искомого объекта;
- повороты искомого объекта;
- наличие изображений в зеркальном виде;
- возможность поиска объекта, который может быть частью искомого;
- возможность поиска объекта, часть которого может быть искомым.

Второе требование объясняется тем, что у петроглифа часто невозможно определить верх и низ. Данные требования накладывает специфические условия на разрабатываемые алгоритмы, делая неосуществимым появление универсального алгоритма. Возможны следующие подходы:

1. двухуровневый поиск;
2. с использованием нейронной сети;
3. используя структурные методы распознавания.

Двухуровневый поиск осуществляется следующим образом. Вначале применяется быстрый, но неточный способ отбора похожих изображений, сужающих зону поиска. Затем используется точный, но медленный метод для окончательного отбора изображений. Для быстрого первичного отбора можно использовать следующий алгоритм. Вокруг петроглифа описывается окружность минимального радиуса, затем с шагом в 10 градусов проводятся радиусы, на которых определяются отрезки, соединяющие дугу окружности и границу петроглифа. Длина каждого отрезка делится на длину радиуса. Последовательность полученных чисел называется кодом петроглифа. Изображения считаются близкими, если существует циклическая перестановка кода петроглифа, такая что расстояния между кодами петроглифов меньше заданного числа. Для окончательного отбора была применена нейронная сеть Хемминга и сеть адаптивного резонанса [6].

Использование структурного метода поиска петроглифа возможно при применении следующего алгоритма:

1. выделение скелетона изображения;
2. кусочно-линейная аппроксимация скелетона с заданным параметром близости;
3. перевод кусочно-линейной аппроксимации в последовательность символов (код рисунка);
4. анализ кодов рисунков на совпадение кодов или их фрагментов с учетом правил их преобразования.

При определении инварианта петроглифа, связанного с анализом его поверхности, были использованы методы фрактального анализа, а именно метод мультифрактальной параметризации структур. Петроглиф выбивался при помощи ударов более крепкой каменной породы, играющей роль инструмента художника, о скалу. Бугристость его поверхности, как правило, выше бугристости скалы, хотя может наблюдаться и обратная картина. Поэтому поверхность петроглифа может рассматриваться как самоподобная поверхность. Проведенные исследования показали перспективность данного направления.

Список литературы

1. Петроглифы Карелии: http://smalt.karelia.ru/~petroglyphs/.

2. Зубкова Н. В., Рогова К. А. Применение методов многомерного анализа для сравнения петроглифов лосей и оленей. // Научные труды молодых исследователей программы «Шаг в будущее» Том 5. М: Изд-во НТА АПФН, 2002, стр. 59 – 61.

3. Айвазян С. А., Мхитарян В. С. Прикладная статистика и основы эконометрики. М.: «Юнити». 1998 г. – 1022 с.

4. Грузман И. С., Киричук В. С., Косых В. П., Перетягин Г. И., Спектор А.А. Цифровая обработка изображений в информационных системах. Учеб. Пособие. Новосибирск: Изд-во НГТУ, 2002 г. – 352 с.

5. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение / Пер. с англ. – М.: Мир, 1989 г. – 478 с.

6. Осовский С. Нейронные сети для обработки информации / Пер. с польского. – М.: Финансы и статистика, 2002 г. – 344 с.

Информационно – поисковая система «Петроглифы Карелии»

Рогова К. А.,
математический факультет Петрозаводского государственного университета
185026, Россия, Карелия, г. Петрозаводск, Комсомольский просп., д.15, кв. 275
E-mail: ksushar@mail.ru

На гранитных склонах островков низовья реки Выг и мысов восточного берега Онежского озера высечено множество разнообразных изображений: птиц, лесных и морских зверей, лодок, людей. Это – петроглифы Карелии – ценнейший памятник первобытной эпохи, получивший мировую известность [1,2]. Наскальные изображения постоянной подвергаются природным воздействиям: осадки, втер, температурные колебания, зарастание мхом и т.д. Некоторые петроглифы могут найти только специалисты. В последние годы к перечисленным воздействиям добавились и неблагоприятные вмешательства людей, стремящихся увековечить свое пребывание рядом с историческим памятником. С целью сохранения уникального наследия, создания возможностей для его изучения и разработки новых информационных методов изучения петроглифов была разработана информационно-аналитическая система «Петроглифы Карелии». Ее отличительной особенностью является широкое использование математических методов в изучении петроглифов. Полученные результаты и разработанные информационные методы анализа петроглифов позволят использовать их при анализе петроглифов других регионов России и во всем мире. Информационно-аналитическая система «Петроглифы Карелии» рассчитана в первую очередь на исследователей петроглифов, а также будет интересна любому человеку, интересующемуся петроглифами. Число пользователей не ограничено. Данное исследование поддержано грантом РГНФ № 05-01-12118в (руководитель Н.В. Лобанова).

Информационно-аналитическая система «Петроглифы Карелии» состоит из 3 блоков: базы данных петроглифов Карелии, модуля онлайнового доступа к базе данных петроглифов и модуля поиска по базе данных изображений петроглифов.

База данных петроглифов Карелии

База данных является наиболее важным блоком всей информационно – аналитической системы. Она необходима специалистам (историкам, краеведам) для хранения и пополнения информации о наскальных изображениях. Поскольку к настоящему времени число открытых петроглифов превышает 2000, то хранение информации о них на бумажных носителях затрудняет проведение исследований. В электронной базе данных становится более удобным ее поиск и хранение, появляется возможность легкого добавления и изменения. Данная информация объединяет в себе как количественный, так и качественный анализ петроглифов.

Петроглифы Карелии разнообразны. Они включают в себя изображения лосей, оленей, лодок, птиц, людей, ведических знаков, медведей и рыб, которые описываются по следующим признакам: местоположения петроглифа, его код, поворот, размер и т.д. Каждое из местонахождений петроглифов можно охарактеризовать по следующим признакам: количество изображений, дата открытия, имя открывателя. Таким образом, база данных петроглифов Карелии представляет собой структурированную информацию о месторождениях петроглифов, их изображениях и характеристиках по выше перечисленным признакам. Базу данных можно пополнять и изменять.

Модуль онлайнового доступа к базе данных

Доступ к базе данных через Интернет реализован в виде сайта (раздела «Каталог»). Одна из характеристик петроглифа – его местонахождение. Зная название места, где находится петроглиф, используя раздел «Каталог», можно найти его изображение, а так же детально исследовать остальные петроглифы и получить информацию о каждом изображении. Поиск происходит путем перемещения с помощью гиперссылок по картам. В настоящий момент на сайте находится более 200 фотографий групп петроглифов с их описаниями, общее количество представленных петроглифов превышает 1000 фигур.

При создании каталога петроглифов были созданы электронные варианты схем нахождения петроглифов.

Кроме раздела «Каталог» сайт «Петроглифы Карелии» включает в себя графитные копии и фотографии месторождений, представлен материал о карельских исследователях.

При создании сайта были выполнены следующие работы: разработана структура сайта и его цветовое решение, большая часть материалов переведена в электронный вид. Подготовлены информационные материалы о петроглифах, содержащие следующие разделы: информация, исследователи, публикации, каталог, новости, авторы, фото, видео, как добраться, математические методы, вопрос – ответ. Таким образом, сайт способствует популяризации, охране и разумному использованию в культурном туризме уникальных памятников древности. Описанный сайт можно посмотреть по адресу: http://smalt.karelia.ru/~petroglyphs.

Модуль поиска по базе данных

Модуль поиска по базе данных представляет собой информационно-поисковую систему «PIRS». Она является программным продуктом, выполненным в системе программирования Delphi. Данная система состоит из нескольких отдельных блоков, соединенных вместе. Основными из них являются: поисковая система по признакам и поисковая система по изображениям. Она предназначена в первую очередь специалистам, занимающимся изучением петроглифов, однако может быть полезной любому любознательному человеку, желающему представить себе, что такое петроглифы.

Блок поисковой системы по признакам разработан на примере петроглифов лосей и оленей Беломорья. Для этих петроглифов разработана система из 16 признаков. Примерами признаков являются: изгиб спины, изгиб задних/передних ног, тип корпуса, тип головы и т.д. После разработки данной системы все признаки были проанализированы на статистическую зависимость методами прикладного статистического анализа (методы определения зависимости между категоризированными признаками, проверка статистических гипотез на однородность и т.д.). Все выбранные петроглифы (лоси и олени Беломорья) были охарактеризованы по выделенной системе признаков и разбиты на группы по степени схожести с использованием метода корреляционных плеяд и иерархического кластерного анализа [3,4,5,6].

Таким образом, если человеку интересно найти петроглифы с определенными признаками, можно воспользоваться данным блоком. Пользователю предлагается выбрать значения признаков. После выбора интересующих признаков, предоставляется выбор точности поиска, т.е. количество совпадений признаков. Это либо полное совпадение всех значений признаков, либо не совпадение одного, двух или трех признаков. Несовпадение одного признака будет означать, что найдутся все петроглифы, у которых все признаки, кроме одного будут совпадать. После выбора всех условий поиска программа ищет петроглифы из базы данных, соответствующие выделенным критериям. В итоге пользователю выводятся на экран номера найденных петроглифов, причем они располагаются по местам их нахождения. При нажатии на номер петроглифа показывается его изображение и информация о нем.

Блок поисковой системы по изображениям предназначен для поиска похожих изображений к данному. На вход подается исследуемое изображение, а на выходе должно появиться наиболее похожее к исходному из базы данных. В большинстве случаях это изображение будет единственным, но в редких случаях их будет несколько. Для поиска используется сеть адаптивного резонанса [7]. Пользователю предоставляется доступ к информации о кодовом номере, месторасположении, характерных признаках найденного петроглифа и петроглифах, близких к нему по ранее описанным признакам.

Заключение

Созданная система «Петроглифы Карелии» позволяет:

- Совершать виртуальные путешествия неограниченному числу людей, знакомя их с археологическими памятниками, имеющими мировое признание;
- Изучать изображения как отдельных петроглифов, так и групп;
- Осуществлять поиск интересующих петроглифов разными способами;
- Сравнивать новые изображения с уже имеющимися в базе данных.

Возможно расширение возможностей поиска изображений в информационно-поисковой системе «PIRS». В настоящее время ведется разработка новых методов поиска для сайта «Петроглифы Карелии».

Список литературы

1. Савватеев Ю. А. Залавруга. Ленинград: «Наука». 1970 г.
2. Савватеев Ю. А. Каменная летопись Карелии. Петрозаводск: «Карелия». 1990 г.
3. Айвазян С. А., Мхитарян В. С. Прикладная статистика и основы эконометрики. М.: «Юнити». 1998 г.

4. Гайдышев И. Анализ и обработка данных. Специальный справочник. СПб.: «Питер». 2001 г.

5. Дюк В., Самойленко А. Data Mining, Учебный курс. СПб.: «Питер». 2001г.

6. Зубкова Н. В., Рогова К. А. Применение методов многомерного анализа для сравнения петроглифов лосей и оленей. // Научные труды молодых исследователей программы «Шаг в будущее» Том 5. М: Изд-во НТА АПФН, 2002, стр. 59 – 61.

7. Грузман И. С., Киричук В. С., Косых В. П., Перетягин Г. И., Спектор А. А. Цифровая обработка изображений в информационных системах. Учеб. Пособие. Новосибирск: Изд-во НГТУ, 2002 г.

Об одном варианте дедуктивного построения базы знаний

Рухая Х. М.,
зав. отд. мат. логики и методологии ИПМ ТГУ. Грузия, Тбилиси, ул. Университетская 2, ИПМ ТГУ.
e-mail: rukhaia@viam.hepi.edu.ge
Тибуа Л. М.,
СНС ИПМ ТГУ. Грузия, Тбилиси, ул. Университетская 2, ИПМ ТГУ.
email: tibua@viam.hepi.edu.ge

Одной из существенных особенностей современных математических теории (как формальных, так и содержательных) является ограничение и сведение, в некотором разумном смысле, к минимуму алфавита теории. Благодаря этому класс предложении теории становится обозримым, точно определяется понятие доказательства, четко ставится ряд математических проблем и облегчается их решение.

Хотя ограничение алфавита теории теоретически имеет ряд важных существенных преимуществ, тем не менее оно вызывает ряд существенных затруднений. Именно, если бы мы пользовались только символами ограниченного алфавита теории мы были бы вынуждены писать весьма длинные формы (формулы и термы). Практически невазможо как письменное представление таких форм, так и восприятие их содержания на основе их письменного представления. В часности, современный математический язык существенно отходит от обычного (математического) языка и прямое его использование становится невозможным. Для преодоления этих трудностей, вводят сокращающие символы и пользуются сокращенными формами. Этим ограниченный язык современной математической теории заменяется языком расширенной сокращающими символами теории, причем последний, оставаясь современным математическим языком, очень близок к обычному математическому языку.

Совершенно идентичное положение имеется в случае искуственных языков. Здесь в роли современной математической теории выступает машинный язык ЭВМ, а в роли расширенной, сокращающими символами, теории выступают языки программирования. Каждый язык программирования получается введением новых операторов, которые естественно можно назвать сокращающими символами. Язык программирования, противовес машинного языка, близок к обычному языку.

При составлении программ на языке программирования программист исполь зует общие законы связывающие язык программирования и машинный язык ЭВМ; т.е.общие законы о свойствах операторов (сокращающих символов) языка программирования и о свойствах предложений языка программирования.Но такие общие законы пока не доказаны -пока их использоваиие опирается на интуицию. Это вызывает необходимость отладки программ.При этом отладка программ осуществляется не безупречным образом. Отладки программ станут излишними,если упомянутые общие законы будут доказаны.Но для такого доказательства необходимо отыскать математическое понятие сокращающего символа(т.е.оператора языка программирования). Иначе говоря,необходимо создать общую теорию языков программирования.

Аналогичное положение было и в математике до 1977 г. Первая попытка создания рациональной теории сокращающих символов на основе введения математического понятия сокращающего символа (вместо интуитивного понятия сокращающего символа) предпринята Ш.С. Пхакадзе в монографии "Некоторые вопросы теории обозначений". математическое понятие сокращающего символа вводится на основе рационального ограничения правил введения сокращающих символов. В упомянутой монографии доказываются важные общие свойства сокращающих символов и сокращённых форм,если только сокращающие символы вводятся с помощью упомянутого ограниченного семейства правил введения сокращающих символов.При этом указанное семейство правил так богато,что с помощью этих

правил можно ввести почти все сокращающие символы, которые применяются в математических теориях.

Упомянутая теория сокращающих символов даёт возможность существенно повысить логическую строгость математических текстов. А это является необходимым условием для создания автоматических устройств,ориентированных на обработку математических текстов.Создание таких автоматических устройств является частью проблемы искусственного интелекта.

Само собой напрашивается создание общей теории операторов (теории сокращающих символов) для языков программирования с помощью тех же методов, которые применяются для создания теории обозначений.Создание такой теории будет означать создание такой общей теории языков программирования, в которой будут доказуемыми общие законы, устанавливающие связь между машинными языками и языками программирования.

Нами был построен один вариант утвердительной программировании, в котором введены понятия: "символ уровня-n" , "форма уровня-n" , "факт уровня-n". В базе знаний содержатся все факты уровня- 0, а также те производные факты, уровень которых больше нуля. База знаний разбита по слоям относительно уровня факта.Соответственно разработана эффективная стратегия достижения цели.

Список литературы

1. Ч. Чень, Р. Ли. Математическая логика и автоматическое доказательство теорем. М. 1983

2. Kh. Rukhaia, L. Tibua. One Version Of Programming According To Logic Tsr Language.Bulletin of TICNI,vol.№4,Tbilisi,2000.

3. Ш. С. Пхакадзе. Некоторые вопросы теории обозначений. Тбилиси,1977.

Интеллектуальная система прогнозирования асимптотической устойчивости в среднем квадратическом стохастической системы управления

Самигулина Г. А.,
к.т.н., доцент, с.н.с.,
Институт проблем информатики и управления МОН РК,
Казахстан, г. Алматы, 050010, ул.Пушкина 125, ком.102,
E-mail:galinasamigulina@mail.ru

В связи с тем, что современные системы управления характеризуются большой сложностью, неопределенностью параметров и нелинейным характером протекающих процессов актуальны неклассические методы анализа и синтеза данных систем. В настоящее время довольно продвинутым в области интеллектуального управления является применение Искусственных Иммунных Систем (AIS), построенных на принципах обработки информации молекулами белков [1].

Большинство сложных объектов управления функционируют в стохастической среде. Для исследования динамических свойств выделенного класса систем управления используется метод квазирасщепления [2]. Данный метод позволяет с помощью алгебраических проекторов декомпозировать сложную систему управления на взаимосвязанные подсистемы меньшей размерности, но эквивалентные по динамическим свойствам исходной системе.

Рассмотрим управляемый объект, на входе которого действуют возмущения типа «белого шума» и математическая модель которого представлена в векторно-матричной форме Ланжевена следующим образом:

$$\dot{X}(t) = A(t)X(t) + B(t)U(t) + D(X)Z(t), t \in I(t_0), \quad (1)$$

где множество $I(t_0) \subset [0,\infty)$, t - текущее время; $X(t) = (X_1(t),\ldots,X_n(t))^T \in R^n$ - вектор состояний управляемого объекта; T - символ транспонирования; $A(t) = [a_{ij}(t)] \in R^{nxn}$ - функциональная матрица; $B(t) = (b_1(t),\ldots,b_n(t))^T \in R^n$ - функциональный вектор, соответственно, удовлетворяющие соотношениям:

$$A^- \le A(t) \le A^+,\ B^- \le B(t) \le B^+. \quad (2)$$

Неравенства (2) понимаются в покомпонентном смысле, где A^+, A^-, B^+, B^- - известные константные матрицы и векторы соответствующих размерностей; $U(t) \in R^1$ - скалярное управление; Z(t) $\in R^n$ - аддитивное векторное возмущение типа белого шума, которое действует на входе объекта управления: $Z(t) = (Z_1(t),\ldots,Z_n(t))^T$.

$D(X) \in R^{nxn}$ - матрица шумов, зависящая от переменных состояний объекта управления.

Управление U(t) выбирается такое, чтобы обеспечить желаемую динамику в замкнутой системе:
$U(t) = U(X(t),t).$ (3)

Желаемая динамика замкнутой системы управления задается в виде:

$$\sigma(t) = C^T X = 0, \quad (4)$$

где $C \in R^n - Const, C^T = [C_1,\ldots,C_n], C_n = 1.$ Соотношения (1-4) определяют математическую модель, далее называемую S- системой [3].

Задача исследования формулируется следующим образом: разработать интеллектуальную технологию исследования асимптотической устойчивости в среднем квадратическом стохастической системы управления (S-системы) на основе метода квазирасщепления и биологического подхода Искусственных Иммунных Систем с целью прогнозирования поведения сложной системы и оперативного управления текущей ситуацией в реальном масштабе времени.

Недостатком математических моделей в форме Ланжевена является наличие возмущения типа белого шума, что ведет к разрывности траекторий движения. Более удобной формой представления математических моделей являются стохастические квазирасщепленные дифференциальные уравнения типа Ито, в которых зависимые возмущения заменяются независимыми винеровскими процессами и корректируются векторы сноса. После данной процедуры имеем две подсистемы: $\overline{S_1}$ и $\overline{S_2}$ записанных относительно функций $x'(t)$ и $\sigma(t)$ в виде стохастических дифференциальных уравнений типа Ито. При этом сохраняется кинематическое подобие исходной S – системе [3].

Для исследования асимптотической устойчивости в среднем квадратическом квазирасщепленных стохастических систем управления на основе перспективного биологического подхода AIS необходимо осуществить следующую классификацию решений. Пусть управление $U(t)$ будет таким [2], что для момента времени $t_1 \ge t_0$ для подсистем $\overline{S_1}$ и $\overline{S_2}$ решения $x'(t),\ \sigma(t)$ удовлетворяют следующему неравенству:

$$\|\sigma(t)\| \le \delta\|x'(t)\| + \eta, \text{ где } \delta,\ \eta \text{ - Const., } \delta \ge 0,\ \eta \ge 0. \quad (5)$$

В данном случае асимптотика решений S - системы определяется асимптотикой решений $\overline{S_1}$ - подсистемы.

В случае, когда выполняется обратное неравенство, то есть:

$$\|\sigma(t)\| > \delta\|x'(t)\| + \eta, \quad (6)$$

где $t \in I(t_1)$, то асимптотика решений определяется подсистемой $\overline{S_2}$.

В пространстве состояний выделим множества конусного типа следующего вида:

$G_{\delta,\eta} = \{x \in R^n : \|\sigma(x)\| \le \delta\|x'\| + \eta\}$, пусть η=0. (7)

$\overline{G}_{\delta,\eta} = \{x \in R^n : \|\sigma(x)\| > \delta\|x'\| + \eta\}$, пусть η=0. (8)

Таким образом, заданные множества порождают в пространстве R^n два класса решений: $\{x(t)\}_1$, $\{x(t)\}_2$, $t \in I(t_0)$. Первый класс решений относится к области $G_{\delta,\eta}$ - подсистема $\overline{S_1}$.

Второй класс решений относится к области $\overline{G_{\delta,\eta}}$ - подсистема $\overline{S_2}$. Исследование динамических свойств исходной S - системы сводится к исследованию либо подсистемы $\overline{S_1}$, либо подсистемы $\overline{S_2}$.

Так как при квазирасщеплении [2] считается, что пространство R^n представляется прямой суммой подпространств L_1 и L_2: $L_1 \subseteq R^n$, $L_2 \subseteq R^n$, то область $G^1_{\delta,\eta}$ будет дополнением множества $G_{\delta,\eta}$ до подпространства L_1, а область $\overline{G^2_{\delta,\eta}}$ будет дополнением множества $\overline{G_{\delta,\eta}}$ до подпространства L_2. Область $G^1_{\delta,\eta}$ будет относиться к третьему классу решений, а область $\overline{G^2_{\delta,\eta}}$ к четвертому классу.

Согласно идеологии AIS формируются эталонные матрицы управления $U_1, U_2, \overline{U_1}, \overline{U_2}$ для каждого из 4 классов. Классы выделяются экспертами в зависимости от областей функционирования и определяются особенностями процессов, параметрами квазирасщепленных подсистем $\overline{S_1}$, $\overline{S_2}$ и различных факторов, влияющих на систему. После сингулярного разложения, получаем правые и левые сингулярные вектора $\{x_1, y_1\}$, $\{x_2, y_2\}$ и т.д. эталонных матриц. Формируются также матрицы управления, рассматриваемые как образы: $u_1, \ldots\ldots u_n$. Энергии связи W_1, W_2, W_3, W_4 между формальными пептидами для предложенных 4 классов могут быть представлены в следующем виде:

$$W_1 = -x_1^T u y_1, \quad W_2 = -x_2^T u y_2, \; W_3 = -x_3^T u y_3, \; W_4 = -x_4^T u y_4, \tag{9}$$

где u - рассматриваемый образ, T – символ транспонирования.

Известно, что нативная (функциональная) укладка белковой цепи соответствует минимуму энергии связи, поэтому минимальное значение энергии связи определяет класс k, к которому принадлежит данный образ:

$$W_k = \min\{W_1, W_2, W_3, W_4\}. \tag{10}$$

Ниже приведен разработанный алгоритм 1, который состоит из 12 шагов.

Алгоритм 1.

Шаг 1. Получение выражений операторов проектирования P_1 и P_2 для стохастической системы управления (S - системы) в форме дифференциальных уравнений Ланжевена.

Шаг 2. Получение квазирасщепленных подсистем S_1' и S_2' на основе операторов проектирования P_1 и P_2 в форме Ланжевена.

Шаг 3. Получение стохастических квазирасщепленных подсистем $\overline{S_1}$ и $\overline{S_2}$ в форме дифференциальных уравнений типа Ито [3].

Шаг 4. Классификация областей решений квазирасщепленных подсистем $\overline{S_1}$ и $\overline{S_2}$.

Шаг 5. Нормировка входных информативных признаков [5].

Шаг 6. Выделение информативных признаков на основе методов факторного анализа [4]. Создание оптимальной структуры искусственной иммунной сети по весовым коэффициентам информативных признаков [5].

Шаг 7. Создание с помощью экспертов временных рядов, состоящих из информативных признаков характеризующих каждый класс, которые рассматриваются как антигены. Для улучшения специфичности узнавания сворачивание временных рядов в матрицы управления, являющиеся эталонами для каждого класса. Сингулярное разложение данных эталонных матриц управления и определение правых и левых сингулярных векторов.

Шаг 8. Процедуры обучения AIS с учителем или без учителя. Оценка погрешностей обучения.

Шаг 9. Создание матриц управления - образов по временным рядам информативных признаков. Матрицы образов рассматриваются как антитела.

Шаг 10. Определение минимальной энергии связи между формальными пептидами (антителами и антигенами) и решение задачи распознавания образов.

Шаг 11. Оценка энергетических ошибок распознавания образов AIS на основе свойств гомологичных белков [6].

Шаг 12. Определение асимптотической устойчивости в среднем квадратическом стохастической системы управления (S-системы) по квазирасщепленным подсистемам $\overline{S_1}$ или $\overline{S_2}$, прогноз поведения сложной стохастической системы и оперативная корректировка управления в реальном масштабе времени.

Несомненным достоинством данного алгоритма является Шаг 11. Обработка многомерной совокупности данных на основе технологии AIS неизбежно ведет к увеличению энергетических погрешностей, которые зависят от ряда факторов, и существенно влияют на достоверность прогноза. Определение нативной укладки цепи, соответствующей минимуму энергии затруднено из-за различных погрешностей, возникающих за счет неполноты данных, корреляции данных и ошибок измерения.

Устранение погрешностей энергетических оценок осуществляется на основе свойств гомологичных белков. Нативная укладка цепи может быть найдена в процессе сворачивания цепи, где потенциал каждого взаимодействия усреднен по гомологам [6]. Ниже приведен Алгоритм 2. оценки энергетических погрешностей AIS.

Алгоритм 2.

Шаг 1. Усреднение потенциалов по гомологам: $< E_i^* >_G = \frac{\sum_i^G E_i^*}{G}$, где символ <…> означает усреднение по гомологам, G - число гомологичных пептидов.

Шаг 2. Определение вычисленной энергии нативной (функциональной) структуры по усредненным гомологам: $< E_i^* >_G \approx E_N$.

Шаг 3. Определение Z - факторов. Значение Z - фактора определяется средним числом стандартных отклонений между энергией нативной структуры и энергией случайно выбранной укладки цепи: $Z = \frac{E_N - < E >}{< (E - < E >)^2 >^{\frac{1}{2}}}$, где $E_N \approx < E_i^* >_G$, $< E >$ - среднее число стандартных отклонений по гомологам, E - энергия случайно выбранной укладки цепи.

Шаг 4. Распознавание нативной структуры белков по гомологам и определение достоверности прогноза на основе AIS в зависимости от величин Z - фактора. Рассчет коэффициентов риска прогнозирования.

Использование биологического подхода AIS для исследования асимптотической устойчивости стохастических квазирасщепленных систем управления позволяет оперативно оценить достоверность прогноза на основе гомологичных белков. Нативная структура белковой цепи, соответствующая минимуму энергии связи, является для каждого класса определенной и позволяет определить принадлежность гомологов к какому-либо классу решений. Особенно это свойство ценно для образов, которые находятся на границах классов. Данная способность AIS существенно уменьшает погрешности энергетических оценок, повышает достоверность прогноза поведения интеллектуальной системы.

Список литературы

[1]. Tarakanov A. O. Formal peptide as a basic of agent of immune networks: from natural prototype to mathematical theory and applications. Proceeding of the I Int. workshop of central and Eastern Europe on Multi-Agent Systems.-1999.;

[2]. Емельянов С. В., Коровин С. К. Дискретные бинарные системы автоматического управления. //Итоги науки и техники. Техн. кибернетика, -М.: ВИНИТИ, -1984, -N17, -С. 70-160.

[3]. Самигулина Г. А. Автоматизированное построение бинарных стохастических систем управления на основе метода квазирасщепления. -Алматы: ИПИУ МОН РК.-2001.- 67с.

[4]. К. Иберла. Факторный анализ. -М.: Статистика. -1980.

[5]. Самигулина Г. А., Чебейко С. В. Выделение информативных признаков штаммов чумного микроба для прогнозирования эпизоотий чумы. //Проблемы нелинейного анализа в инженерных

системах. Казанский государственный технический университет им. А. Т. Туполева. -Казань. -2004.-вып.1(20).-том.10.-С.83-95.

[6]. Samigulina G. A., Chebeiko S. V. Technology of elimination errors the energy estimations of Artificial Immune Systems of the forecasting plague. Proceedings of the 6 international conference on Computational Intelligence and Natural Computation. - Cary, North Carolina, -USA. -2003. -P.P. 1693-1696.

Моделирование избирательной кампании методами нечетких игр

Самонов В. Е.
Доцент кафедры теоретической физики
Ставропольского государственного университета,
355019, г. Ставрополь, ул. Пушкина, 1,
e-mail: vit-samonov@yandex.ru

Как известно, теоретико-игровые модели весьма успешно применяются при анализе экономических и социально-политических процессов [1], субъектами которых могут быть финансово-промышленные группы, властные элиты, политические партии, отдельные государства и т.д. Однако в большинстве случаев при построении такой модели возникает проблема количественной оценки выигрыша сторон в конкретной игровой ситуации. Указанная проблема частично может быть разрешена благодаря переходу к нечетким игровым моделям, теоретическая основа которых разрабатывается в последнее десятилетие (см., например, [2]).

Поставим задачу разработки нечеткой теоретико-игровой модели типичного социально-политического процесса – избирательной кампании. Выбор объекта моделирования объясняется, во-первых, наличием большого числа соответствующих политологических исследований, позволяющих провести апробацию модели, а, во-вторых, самостоятельной практической ценностью разрабатываемой модели. В настоящей работе обобщаются результаты, полученные автором ранее [3].

Пусть на электоральном поле имеется M кандидатов на выборную должность, множество которых обозначим через $C = \{c_1, c_2, ..., c_M\}$. Будем предполагать, что между кандидатами отсутствуют какие-либо договоренности и коалиции, а целью каждого из них является только победа в выборах. Учитывая особенности избирательного законодательства, в ряде случаев следует ввести фиктивного $M+1$-го «кандидата против всех». Также на данном этапе моделирования будем предполагать неограниченность ресурсов кандидатов.

В общем случае каждый из M кандидатов может воспользоваться одной из стратегий множества $S = L \times T \times O$, являющегося декартовым произведением множеств $L = \{L_1, L_2, L_3\}$, $T = \{T_1, T_2, T_3, T_4\}$ и $O = \{O_1, O_2, O_3, O_4\}$, соответственно характеризующих общее направление избирательной кампании, ее активность во времени и позиционируемый образ кандидата. Согласно [4], общая линия кампании может основываться на рекламе позитивных сторон и качеств кандидата (L_1), защите его слабых и уязвимых сторон (L_2) и критике конкурентов (L_3). Активность проведения кампании во времени подразделяется на «стратегию рывка» (T_1), «стратегию большого события» (T_2), «стратегию быстрого финала» (T_3) и «крейсерскую стратегию» (T_4). И, наконец, кандидат может позиционировать себя как «спаситель Отечества» (O_1), «отец нации» (O_2), «обаятельный лидер» (O_3) или «свой парень» (O_4).

Как известно, агитационные мероприятия могут проводиться с различной частотой и иметь разную степень психологического воздействия на избирателя. Для учета этого факта введем нечеткую переменную интенсивности агитационных мероприятий I, которая может принимать одно из четырех значений: нулевая интенсивность I_0 (полный отказ от проведения агитационных мероприятий), низкая I_1, средняя I_2 и высокая I_3 интенсивности. Заметим, что величину интенсивности следует относить не к отдельному агитационному мероприятию, а к их совокупности или всей кампании в целом.

Оценка интенсивности агитационных мероприятий в конкретных избирательных кампаниях является предметом отдельного исследования. Аналогичным образом можно оценивать интенсивность произвольных рекламных акций.

Для учета различной интенсивности агитационного воздействия и возможности использования кандидатами смешанных стратегий заменим в строящейся модели множество стратегических линий избирательной кампании L на множество упорядоченных кортежей $\tilde{L} = \left\{\left\langle L_1, I^{(1)}\right\rangle, \left\langle L_2, I^{(2)}\right\rangle, \left\langle L_3, I^{(3)}\right\rangle\right\}$. Здесь, как и ранее, $L_i \in L$ характеризует одну из стратегических линий кампании, а $I^{(i)} \in I = \{I_0, I_1, I_2, I_3\}$ соответствует интенсивности агитационных мероприятий, реализующих данную стратегическую линию. Вообще говоря, аналогичное обобщение может быть выполнено и для множеств T и O. Однако оно малозначимо в практическом смысле и приведет к неоправданному усложнению модели, доведя общее число стратегий более чем до 4 млн.

Таким образом, множество возможных стратегий избирательной кампании в рамках строящейся модели имеет вид $\tilde{S} = \tilde{L} \times T \times O$. Можно показать, что общее число стратегий равно $card\left(\tilde{S}\right) = 1008$. В реальности количество используемых стратегий будет значительно меньше по причине отбрасывания заведомо проигрышных, а также ввиду ограниченности ресурсов, не рассматриваемой в настоящем исследовании.

Следует отметить, что простой анализ предвыборных стратегий кандидатов даже с учетом интенсивности мероприятий по их реализации не позволяет корректно предсказать итоги выборов. Иллюстрацией сказанного служит простейший пример. Допустим, что все кандидаты используют в своей кампании одну и ту же стратегию. Очевидно, в этом случае исход выборов для них заведомо не будет одинаковым.

По этой причине в разрабатываемой модели также необходимо учесть электоральные предпочтения избирателей, определяемые их социально-демографическим составом и политической ориентацией. Поскольку состав электората неоднороден, целесообразно разбить совокупность всех избирателей на кластеры. Проблема кластеризации избирателей в соответствии с их электоральными предпочтениями также является предметом отдельного исследования.

Допустим, эта задача решена и выделено K типологических групп избирателей, образующих множество $V = \{v_1, v_2, ..., v_K\}$. Численность каждой группы соответственно равна N_1, N_2, …, N_K. Общий итог выборов (выигрыш кандидата) равен сумме выигрышей в каждой из указанных групп. Причем, оптимальные стратегии кандидата в различных группах могут различаться.

Заметим, что стратегии кандидата в различных типологических группах избирателей не могут быть абсолютно независимыми. Это связано, во-первых, с ограниченностью ресурсов в реальной ситуации, а, во-вторых, с общедоступностью информационных каналов для всех групп избирателей. Последнее в особенности относится к электронным СМИ и наружной рекламе. По этой причине взаимодействие кандидатов целесообразно рассмотреть как игру на «множестве K игровых полей».

Ввиду отсутствия строгой зависимости между избранными стратегиями кандидатов и исходом игры, выигрыши сторон в каждой типологической группе следует выразить нечеткими числами. Общий выигрыш, соответственно, равен сумме выигрышей на каждом из «игровых полей». При этом степень нечеткости общего результата, согласно [5], будет возрастать.

В результате нами построена нечеткая теоретико-игровая модель «на множестве игровых полей». Причем, нечеткими являются как стратегии сторон (например, интенсивность агитационного воздействия), так и их выигрыши. Оценка выигрышей сторон и поиск оптимальных стратегий возможны после анализа большого числа реальных выборов и обработки результатов соответствующих политологических исследований.

Далее приведем некоторые соображения по упрощению построенной модели и выбору оптимальных стратегий игроков.

Несмотря на то, что кандидаты взаимодействуют с K группами избирателей (количество которых может достигать десятков), число игровых ситуаций и, соответственно, оптимальных стратегий на всех «игровых полях» значительно меньше. В подтверждение сказанного рассмотрим возможные отношения на множествах кандидатов C и групп избирателей V.

1. Отношение «*распределенности*» $\Omega_1 \subset V \times C$:

$$\begin{gathered}\forall v_i \in V\ \exists c_j \in C : (v_i, c_j) \in \Omega_1, \quad \forall c_j \in C\ \exists v_i \in V : (v_i, c_j) \in \Omega_1, \\ \forall v_i \in V (v_i, c_j) \in \Omega_1 \wedge (v_i, c_k) \in \Omega_1 \Rightarrow c_j = c_k, \\ \forall c_k \in C (v_i, c_k) \in \Omega_1 \wedge (v_j, c_k) \in \Omega_1 \Rightarrow v_i = v_j.\end{gathered} \qquad (1)$$

В этом случае электоральные предпочтения каждого из кластеров избирателей распределены между всеми кандидатами: каждый из кандидатов опирается на свою адресную группу, а каждая группа избирателей имеет «своего» кандидата. Число кандидатов равно числу избирательных групп ($M = K$).

2. Отношение «*протеста*» $\Omega_2 \subset V \times C$:

$$\forall v_i \in V(v_i, c_j) \in \Omega_2 \wedge (v_i, c_k) \in \Omega_2 \Rightarrow c_j = c_k, \quad \exists v_i \in V : \forall c_j \in C\,(v_i, c_j) \notin \Omega_2 . \qquad (2)$$

В рассматриваемом отношении интересы каждой избирательной группы выражает не более одного кандидата. При этом существуют группы избирателей, не поддерживающие ни одного из кандидатов. Представители этих групп при отсутствии надлежащих агитационных мероприятий могут либо вообще проигнорировать выборы, либо проголосовать против всех. Очевидно, в данном случае количество кандидатов меньше числа кластеров электоральных предпочтений ($M < K$).

3. Отношение «*частичной конкуренции*» $\Omega_3 \subset V \times C$:

$$\forall v_i \in V\ \exists c_j \in C : (v_i, c_j) \in \Omega_3, \quad \exists v_i : (v_i, c_j) \in \Omega_3 \wedge (v_i, c_k) \in \Omega_3, c_j \neq c_k . \qquad (3)$$

В этом случае каждая группа избирателей имеет «своего» кандидата, причем некоторые элементы множества V являются адресными группами нескольких кандидатов (например, электорат «СПС» и «Яблока» на парламентских выборах 2003 года). Число кандидатов в рассматриваемом случае превышает число избирательных групп ($M > K$).

4. «*Смешанное*» отношение $\Omega_4 \subset V \times C$:

$$\exists v_i \in V : \forall c_j \in C\,(v_i, c_j) \notin \Omega_4, \quad \exists v_i : (v_i, c_j) \in \Omega_4 \wedge (v_i, c_k) \in \Omega_4, c_j \neq c_k . \qquad (4)$$

Отношение Ω_4 допускает как наличие конкуренции между кандидатами внутри одной адресной группы, так и наличие избирательных групп, не поддерживающих ни одного из кандидатов. При этом возможно любое соотношение чисел M и K .

В соответствии с рассмотренными отношениями на множествах C и V , все K типологических групп избирателей применительно к конкретному кандидату можно разделить на четыре класса.

1. *Адресная группа кандидата в отсутствие конкуренции* Q_1 – класс избирателей поддерживающих исключительно данного кандидата. Необходимым условием существования этого класса является наличие одного из отношений Ω_1 или Ω_4. Отношение Ω_1 также является достаточным условием.

2. *Адресная группа кандидата при наличии конкуренции* Q_2 – класс избирателей, поддерживающих помимо данного кандидата одного или нескольких его конкурентов, также опирающихся на эту группу. Необходимыми условиями существования такого класса является справедливость одного из отношений Ω_3 или Ω_4.

3. *Адресная группа соперников (одного или нескольких)* Q_3. Достаточными условиями существования этого класса являются отношения Ω_1 и Ω_3.

4. *Группа «протеста»* Q_4 – класс избирателей, не поддерживающих ни одного из кандидатов. Необходимым условием существования такого класса является справедливость одного из отношений Ω_2 или Ω_4. Отношение Ω_2 также является достаточным условием существования Q_4.

Учет приведенной классификации типологических групп избирателей позволяет существенно уменьшить количество стратегий кандидата на соответствующем «игровом поле».

Так, в классе Q_1 доминирующими направлениями кампании являются формирование положительного имиджа кандидата и защита от критики конкурентов; в классе Q_3 преобладает критика конкурентов и формирование положительного имиджа кандидатов; в классе Q_4 – реклама позитивных сторон кандидата. Остальные стратегические линии являются менее важными и могут быть опущены в процессе «привязки» модели к конкретным условиям.

Наиболее сложной и ресурсоемкой является избирательная кампания в классе Q_2, требующая использования всех трех главных стратегических линий кампании L . Это связано с наиболее жесткой конкуренцией между кандидатами за право доминирования у «своих» избирателей. В таких случаях

результативность кампании определяется не только ее стратегией, но и тактикой проведения (см., например, [6]).

Разработка модели, включающей тактические приемы избирательной гонки, является предметом дальнейших исследований. Очевидно, новая модель будет более адекватно учитывать реакцию избирателей на действия сторон и позволит заблаговременно скорректировать ход кампании. В основе моделирования планируется использовать математический аппарат дифференциальных игр.

Список литературы

1. Малафеев О. А., Муравьев А. И. Математические модели конфликтных ситуаций и их разрешение. Том 2. – СПб: Изд-во СПбГУ, 2000. – 294 с.

2. Molina E., Tejada J. The equalizer and the lexicographical solutions for cooperative fuzzy games: characterization and properties. // Fuzzy Sets and Systems. - 2002. - Vol. 125, No. 3 - P. 369-387.

3. Самонов В. Е. Теоретико-игровой подход в моделировании предвыборной агитации. // Математические методы в технике и технологиях – ММТТ-19: Сб. трудов XIX Международ. науч. конф. В 10 т. Т 7. Секция 7 / под общ. ред. В. С. Балакирева. – Воронеж: ВГТА, 2006. – С. 177-179.

4. Общая и прикладная политология. / Под ред. Жукова В. И. и Краснова Б. И. – М.: МГСУ, 1997.

5. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ-Петербург, 2003. – 736 с: ил.

6. Фаер С. А. Приемы стратегии и тактики предвыборной борьбы. – СПб: «Стольный град», 1998. – 136 с.

Применение факторного анализа для систем биометрической идентификации

Семенов А. В.,
e-mail: semen11@psu.ru
Чугайнов Е. С.,
e-mail: yeugenius@mail.ru
Пермский государственный университет
кафедра компьютерных систем и телекоммуникаций
Россия 614990 г.Пермь, ул.Букирева, 15

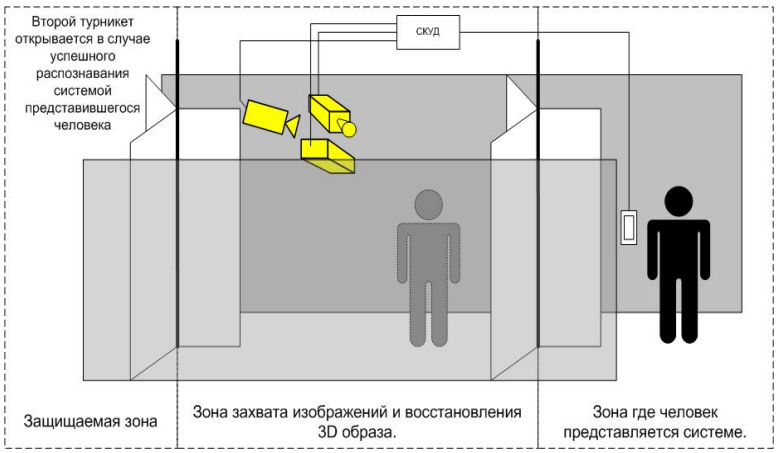
В данной работе описана постановка задачи использования факторного анализа для 3D модели человеческого лица с целью получения набора отличительных факторов, взаимосвязей между ними и выявления латентных признаков для систем биометрической идентификации.

Введение

Для систем биометрической идентификации, а главное, для систем контроля доступа, ускорение процесса идентификации и поиска образца на сегодняшний день одна из важнейших задач. Так, например, для системы контроля доступа большого предприятия на несколько десятков тысяч пользователей поиск соответствия в базе данных образца, который представлен системе может достигать десятков секунд. Для проходных крупных предприятий это не приемлемо, т.к. чревато ежедневными очередями людей. Проблему поиска полученной биометрической информации в базе данных (БД) можно решить организацией сравнения с определенными записями в БД закрепленными за конкретным пользователем. Пользователь системы представляется при помощи бесконтактной радиочастотной метки (это может быть карточка, активная метка, либо метка, вживленная в кожу человека). Задача системы - найти запись в БД и по биометрическим данным убедиться в соответствии.

Для случая распознавания лиц в потоке людей (метро, торговые центры, вокзалы), где система не получает информацию о личностях, осуществляющих проход, необходимо применять алгоритмы позволяющие быстро осуществлять захват изображений, получение биометрических данных и

максимально быстрый поиск в БД. Для такой задачи предлагается использовать факторный анализ (для выявления признаков) и кластерный анализ (для поиска по признакам), применяя их к 3D образам.



Для проходных крупных предприятий идеальной является статистическая идентификация пользователей по особенностям геометрии лица [4]

Рис.1 Проходная

Использование 3D образов обусловлено тем, что можно получить больше значимых факторов, нежели в плоском изображении, тем самым расширяя количество хранимых в памяти образов и уменьшая ошибки.

Факторный анализ на 3D образах также может позволить производить дальнейший поиск значимых компонент среди эмоциональных состояний человека и скрытым связям между ними, тем самым получая психологическое состояние человека, выявляя лица, проводящих диверсию, совершивших кражу или собирающиеся совершить какое либо преступление.

Обзор существующих работ

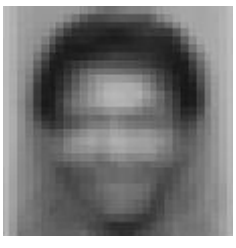
Моделирование трехмерных человеческих лиц на сегодняшний день имеет уже довольно широкое распространение. К такому выводу можно прийти по большому числу публикаций различных методов, например [6][7]. Из прочитанных статей наиболее близкими являются статьи, описывающие изменяемую модель для синтеза трехмерных образов человеческих лиц[1]-[2][5]. Основная идея в данных работах – это получение 3D образа лица из средней модели лица. Специально разработанный алгоритм выделяет важные факторы из двухмерного изображения и добавляет их в среднюю 3D модель, путем ее изменения. В случае использования изображений с разных ракурсов можно добиться качества лазерного сканирования.

Факторный анализ примененный к 2D изображениям

В работе реализован алгоритм, использующий факторный и кластерный анализ для распознавания 2D изображений и оптимизации хранения и поиска факторов в базе данных. Работа проводилась на внутреннем языке программного продукта MATLAB. В память были загружены обработанные изображения лиц – они представлялись в виде массива интенсивности, размером 30*30 – то есть в каждой ячейке находится интенсивность определённого пикселя. Изображения лиц предварительно были отцентрированы и приведены к единому масштабу. Далее, каждое изображение лица было переведено в вектор, размерностью k = m*n, где m и n - ширина и высота, то есть от двумерного массива интенсивностей мы перешли к одномерному [x1,x2… xk]. Таким образом, мы получили вместо переменных двумерных массивов переменные в виде одномерных векторов, к которым удобнее применять методы факторного анализа. Далее, учитывая среду программирования, где основным элементом является матрица, было решено составить из полученных векторов массив, чтобы получить возможность обрабатывать все вместе в цикле, а не по одной переменной. Затем необходимо было найти средний вектор – среднее лицо и затем построить новый массив векторов-разностей между текущим изображением и средним изображением. Средний вектор:

$$E = \frac{1}{M}\sum_{i=1}^{M} Y(i), \quad (1)$$

где Y(i) – вектора лиц, а M – их количество



Для того, чтобы перейти от хранения изображения в виде вектора к обычному двумерному виду, чтобы просмотреть результат, был написан алгоритм на примере среднего вектора – среднего лица, его можно увидеть на рисунке 2.

Затем был реализован алгоритм построения массива разностей между

исходным вектором изображения и средним вектором. Получив вектора разностей между средним и исходным, строим матрицу ковариации.

Из матрицы ковариации находим матрицу собственных векторов и диагональную матрицу собственных значений. Далее, для последующего выявления главных факторов, в чём и заключается смысл факторного анализа, необходимо упорядочить собственные вектора по собственным значениям, то есть от максимального к минимальному. Самый значащий вектор будет иметь наибольшее собственное значение. Для этого из диагональной матрицы собственных значений необходимо сделать вектор, а затем упорядочить. Получив собственные вектора для матрицы ковариации, мы можем получить собственные вектора, вычисленные для всего набора изображений лиц, которые называются собственными лицами (eigenfaces). Собственные лица имеют полезное свойство, заключающееся в том, что изображение, соответствующее каждому такому вектору имеет форму лица.

Таким образом, получив базис 30 значений, упорядоченный от наиболее важного к наименее важному можно проводить выборку по наиболее важным компонентам а остальные отбросить.

Далее, для поиска был применен кластерный анализ. Взяв за основу главный значащий фактор, мы разбиваем все изображения на группы, потом используем вторую по значимости компоненту, разбиение идет уже внутри группы, и так далее, до тех пор, пока не будет выявлено нужное нам изображение.

Экспериментальным путем устанавливается необходимое количество факторов для нахождения любого лица в базе данных. Остальные факторы можно отбросить, что значительно уменьшает базис и ускоряет поиск. Остальные компоненты кодируют мелкие отличия и шум. Разница между компонентами 30 изображений показана на рисунке 3. В каждом разбиении видны группы изображений, сходных по компонентам, которые уже различны во втором по значимости факторе.

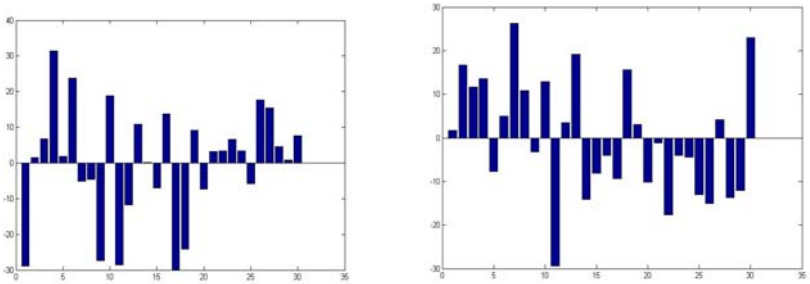


Рис.3 Разделение образов по главной и второй компоненте.

Ввод 3D образов.

Для получения 3D образов человеческих лиц можно использовать лазерный сканер или восстановление по нескольким изображениям с разных ракурсов. Первый вариант дорогой, второй более актуален на сегодняшний день, как по скорости, так и по цене. Набор изображений можно одновременно получать с нескольких видеокамер, производя детекцию лиц, локализацию, сопровождение и дальнейший захват, либо с одной видеокамеры, используя последовательность кадров движения головы человека. При использовании нескольких камер можно расположить их так, чтобы получить оптимальные виды человеческого лица для последующего восстановления 3D образа [3].

Цели и задачи. Факторный анализ для 3D случая

Основной целью на данный момент, является применение полученных результатов в двухмерном случае к 3D случаю, т.е. факторный и кластерный анализ.

Поверхность 3D образа можно представить в виде набора векторов хранящих пространственную информацию и информацию о цвете Поверхность это набор точек в пространстве. Представим ее в виде одномерного вектора S (сурфейс). Информацию о цвете так же в виде вектора T (текстура).

$$S=(X1,Y1,Z1,\ldots,Xn,Yn,Zn)\,, \qquad (2)$$

$$T=(R1,G1,B1,\ldots,Rn,Gn,Bn) \qquad (3)$$

Для получения среднего лица необходимо также как в двухмерном случае, из большого числа трехмерных образов найти средний вектор S и T.

$$S_{\text{mod}} = \sum_{i=1}^{m} a_i S_i\,, \qquad (4)$$

$$T_{\text{mod}} = \sum_{i=1}^{m} b_i T_i , \qquad (5)$$

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} b_i = 1 \qquad (6)$$

Соответственно разница между средним лицом и 3D образами:

$$\Delta S_i = S_i - \overline{S} , \qquad (7)$$

$$\Delta T_i = T_i - \overline{T} . \qquad (8)$$

Тем самым мы получили матрицы ковариации, из них мы находим матрицу собственных векторов и диагональную матрицу собственных значений. После необходимо произвести упорядочивание собственных векторов по собственным значениям. Тем самым мы получим упорядоченные компоненты, по которым необходимо производить поиск.

Заключение

В данной работе осуществлена постановка задачи применения факторного анализа к биометрической идентификации личности по 3D образу на основе алгоритмов и данных полученных на 2D изображениях. При использовании факторного анализа достигается высокая точность распознавания, дальнейшая скорость обработки и поиска, по ключевым компонентам, а также появляется возможность использовать психометрию для выявления скрытых отклонений в поведении человека.

Авторы благодарят М. А. Марценюка за обсуждение работы.

Список литературы

[1] J. Huang, B. Heisele, V. Blanz. Component-based Face Recognition with 3D Morphable Models.

[2] V. Blans, T. Vetter. A Morphable Model For The Synthesis Of 3D Faces.

[3] J. Lee, B. Moghaddam, H. Pfister, R. Machiraju. Finding Optimal Views for 3D Face Shape Modeling. IEEE Int'l Conf. on Automatic Face & Gesture Recognition (FG'04), April 2004.

[4] А. И. Иванов. Нейросетевые алгоритмы биометрической идентификации личности. М: «Радиотехника», 2004.-144 с.:ил.

[5] V. Blanz, T. Vetter. Face Recognition Based on Fitting a 3D Morphable Model. IEEE Transctions on pattern analysis and machine intelligence, VOL. 25, NO. 9 September 2003

[6] B. Moghaddam, J. Lee, H. Pfister, R. Machiraju. Model-Based 3D Face Capture with Shape-form-Silhouettes. ICCV'03, Nice, France.

[7] A. M. Bronstein, M. M. Bronstein, R. Kimmel, A. Spira. 3D Face Recognition without Facial Surface Reconstruction. CIS-2003-05

Генетические алгоритмы создания CoreWars-бойцов

Скробов А.
Уральский государственный университет,
620083, Екатеринбург, Тургенева, 4
E-mail: tyomitch@gmail.com

CoreWars – игра, в которой соревнуются друг с другом программы, написанные на языке ассемблера виртуальной машины "MARS" с достаточно необычным устройством [1]. Эти программы по своему

характеру схожи с компьютерными вирусами, и даже имеют общие с ними корни. Они используют все те же приёмы, что и биологические организмы: они могут двигаться и размножаться, общаться и нападать друг на друга, ставить западни, маскироваться, и даже чинить свои повреждения. Ветвь биологии, использующая CoreWars в качестве модели биологического мира, получила название “бинологии”, т.е. двоичной биологии. Однако CoreWars – не только объект бинологических исследований, но и классическая среда для состязаний программистов: соревнования CoreWars-программ регулярно проводятся на протяжении уже более 20 лет, и однажды (в 1988 г.) победителем международного турнира стал наш соотечественник, Е. Лилитко. Но несмотря на неугасающую популярность этой игры на Западе, в России она остаётся практически неизвестна: даже русский перевод её правил [2] мной был выполнен впервые.

Соревнования CoreWars-бойцов ведутся на “холмах”, где каждый боец получает очки по итогам сражения с другими бойцами на том же холме. После серии сражений каждого бойца со всеми остальными определяется его итоговый счёт; определённое число бойцов с наилучшим счётом остаются на холме, а остальные удаляются. При попарном соревновании бойцов, используемом обычно на холмах, каждому бойцу начисляется 3 очка за победу над противником и 1 очко за ничью. Различные холмы отличаются размером (числом бойцов, находящихся на холме одновременно) и используемыми правилами.

На основании сходства мира CoreWars с биологическим миром можно ожидать, что удастся эффективно использовать для генерации CoreWars-бойцов генетические алгоритмы, уже доказавшие свою эффективность при формировании реальных биологических организмов, – и действительно, с 1991 г. в соревнованиях участвуют бойцы, полученные при помощи генетических методов [3]. Можно использовать сгенерированных бойцов для соревнований, не анализируя их код – а можно попытаться разгадать стратегию, сформировавшуюся в них в процессе эволюции, и использовать эту стратегию в обычных, “рукотворных” бойцах. В 1999 г. генетически сгенерированный боец впервые выиграл соревнование с бойцами, созданными людьми; на данный момент бойцы, производимые такими программами – “генетическими генераторами” и обычно дорабатываемые вручную, выступают на равных с чисто “рукотворными”.

Всплеск исследований в области применимости генетических алгоритмов для создания CoreWars-бойцов приходится на начало 2000-х г.г., когда необходимые для этого вычислительные мощности стали более доступными. К настоящему времени в генетических генераторах опробовано множество разнообразных подходов: скрещивание, мутации, инверсии и различные их комбинации; различаются также подходы к реализации естественного отбора, к организации и инициализации популяций и т.д. Чаще всего генераторы работают на уровне “машинного кода” MARS, хотя есть и “оптимизаторы исходного кода”, подбирающие значения констант с использованием генетических методов. Работа включает ещё не публиковавшийся обзор конкретных стратегий, применявшихся в реально существующих генетических генераторах. Последний из обзоров такого рода относится к 2001 г. и не включает стратегии, выработанные за последние 5 лет [4]; кроме того, русскоязычных обзоров подобного рода не публиковалось вовсе.

Поскольку именно требования к вычислительной мощности являются сейчас фактором, сдерживающим развитие генетических генераторов, наиболее плодотворным оказывается создание параллельных и распределённых генераторов. Один такой распределённый генератор, работавший около трёх месяцев на сложно организованной сети из нескольких десятков обычных ПК, произвёл в 2003 г. [6] бойца, занявшего 16 место на холме, где до тех пор не было ни одного генетического бойца. Мной уже реализовано на C# ядро, эмулирующее MARS, и планируется создание на его основе собственного распределённого генератора. [7]

Будем считать генотипом бойца его программу; каждую её инструкцию, состоящую из кода операции и пары операндов с режимами адресации, – геном. Генофондом набора бойцов в этом случае будет совокупный набор их инструкций-генов. Такое введение генов возможно благодаря намеренной примитивности виртуальной машины MARS: 16 операций, 8 режимов адресации, арифметика по модулю 8000 и ограничение длины программы 100 инструкциями. Несмотря на это ограничение длины, реальные CoreWars-бойцы часто содержат внутри себя промежутки, не занятые выполнимым кодом и предназначенные для изоляции отдельных частей бойца: эти промежутки естественно уподобить межгенным промежуткам в реальных ДНК; но мы для однообразия будем считать инструкции-пустышки, образующие эти промежутки, генами – наравне с выполнимыми инструкциями. Такие гены, не соответствующие никакому выполняемому бойцом коду, будем считать “интронами”. В качестве первого шага моей работы я провёл исследование структуры одного CoreWars-бойца путём экспериментального выявления в его коде таких интронов. Оригинальность этого исследования состоит в применении генетических методов к существующему бойцу, а не “первобытному бульону” из случайных инструкций-генов, как у других авторов.

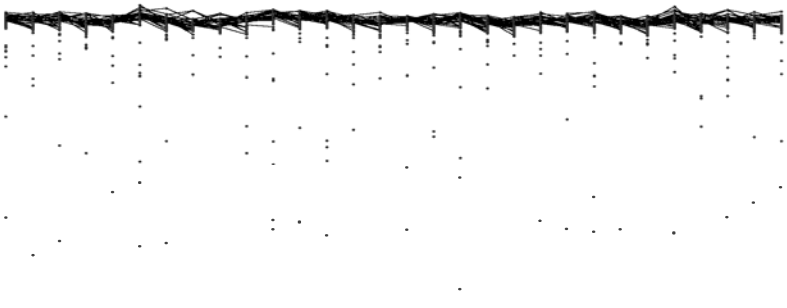
Эксперимент проходил следующим образом: я случайным образом менял в коде бойца значение одного операнда (“мутация”), и оценивал жизнеспособность получившегося бойца-мутанта, проводя его бой на холме с 50 эталонными бойцами, взятыми из специального комплекта бойцов с широким

спектром стратегий, и остальными мутантами того же “поколения”. Каждое поколение состояло из 50 бойцов: все бойцы первого поколения содержали ровно по одной случайной мутации, затем после каждого боя и следующей за ним оценки бойцов текущего поколения формировалось следующее поколение. Для этого 25 худших бойцов текущего поколения удалялись, исключая свои мутации из генофонда, а 25 лучших “размножались” – для каждого из них создавалось по копии с новой добавленной мутацией. Таким образом, бойцы всех поколений, кроме первого, содержали неравное число мутаций: у “более молодых” бойцов их было больше, у “более старых” – меньше. Попутно в этом эксперименте проверялась “мутационная устойчивость” бойца, т.е. сохранение им жизнеспособности после внесения в его код мутаций.

Мой эксперимент занял около 5 суток на компьютере с процессором Celeron 1700; за это время прошло 30 циклов-поколений. Самый “старый” из бойцов, выживших к концу эксперимента, прожил 8 поколений. Число накопившихся в каждом бойце мутаций – менее десятка; значит, бойцы, накапливавшие большое число мутаций, удалялись. Интересно, что две конкретных мутации есть во всех без исключения бойцах последнего поколения: очевидно, эти мутации добавились в генофонд на одном из первых поколений эксперимента, и затем по стечению обстоятельств все бойцы без этих двух мутаций были вытеснены.

Далее я проанализировал количество бойцов, содержавших мутации в каждой конкретной инструкции; если таких бойцов много, это значит, что такая мутация закрепилась за достаточное число поколений до прекращения эксперимента, и стало быть, прошла “проверку временем” как не нарушающая жизнеспособность. Большинство таких “безопасных мутаций” приходятся на инструкции-пустышки, хотя крайне интересно, что некоторые осмысленные инструкции также подвергались мутации без видимого ухудшения боевых качеств программы. Однако, далеко не все бывшие в исходной программе инструкции-пустышки распознались как “точки безопасных мутаций” (интроны): это можно объяснить недостаточным объёмом проведённого эксперимента – а именно, не во всех инструкциях успело произойти хотя бы по одной мутации за весь ход эксперимента. Всего в генофонде 30 поколения (образованном 25 бойцами) есть мутации в 32 инструкциях (из 100), в т.ч. в 15 пустышках (из 47). Можно предположить, что эксперимент втрое большей длительности позволил бы получить поколение, генофонд которого включает мутации во всех интронах.

Оценивая жизнеспособность бойцов-мутантов, я обнаружил, что большинство их (более ⅔) не отличаются от исходного бойца по своему “фенотипу”, т.е. поведению; в таблице итогов боя они все оказываются рядом, с незначительно отличающимся счётом: 138–147 очков, при 143 очках у начального бойца. Очевидно, в этих бойцах мутация попала либо в инструкцию-пустышку, либо в инструкцию, ценность которой не проверяется использованным мной набором эталонных бойцов. Оставшиеся бойцы (менее ⅓) получали с мутацией “критическое повреждение”, делающее их нежизнеспособными; такие бойцы оказывались внизу итоговой таблицы с очень маленьким счётом: 100 и менее очков. Иногда мутация не попадала ни в одну из этих двух категорий, и “калечила” бойца так, что он, оставаясь жизнеспособным, терял в своих боевых качествах и оказывался ближе к низу таблицы с промежуточным счётом: 119–134 очков. Такие “калеки” накапливались с течением поколений, и поэтому положение мутантов в итоговой таблице становилось всё менее и менее кучным. При этом ни один боец за весь ход эксперимента не превосходил по жизнеспособности исходного бойца сколько-нибудь существенно. На графике представлены очки всех 750 бойцов, полученных в ходе эксперимента: вертикальный столбик точек – одно поколение, точка – боец, наклонная линия – переход бойца в следующее поколение.



Объяснить полученные результаты можно тем, что сильные бойцы, поскольку они должны при получении повреждений во время боя оставаться жизнеспособными, содержат внутри себя достаточно средств для нейтрализации точечных изменений, будь они вызваны действиями противника во время боя или мутацией, – поэтому эффект мутаций в большинстве случаев не изменял поведения бойца в целом. С другой стороны, существенное улучшение боевых качеств программы потребовало бы изменения её стратегии, а это едва ли возможно вызвать лишь изменениями операндов, оставляя сами операции неизменными. При этом допущение случайной модификации операций (скажем, JMP на ADD) привело

бы к резкому увеличению числа нежизнеспособных мутантов, а значит, потребовало бы намного большего числа поколений для достижения значимых результатов.

По-прежнему остаются нерешёнными некоторые вопросы: например, были ли бы результаты качественно другими, если бы мутациям подвергался более слабый боец, или невозможность улучшения существующего бойца такими методами не зависит от его изначальной жизнеспособности. В настоящее время работа над данным проектом продолжается.

Список литературы

1. Скробов А., Информация о CoreWars (2006) – http://cs.usu.edu.ru/corewars/intro.html

2. Mark Durham и др., Комментированный проект стандарта ICWS-94 (1995); перевод: Скробов А. (2006) – http://cs.usu.edu.ru/corewars/icws-94.html

3. John Perry, Core wars genetics: The evolution of predation (1991) – http://www.koth.org/info/evolving_warriors.html

4. David G. Andersen, The Garden: Evolving Warriors in Core Wars (2001) – http://www.angio.net/res/garden.html

5. Barkley Vowk, 100% Evolved warrior on 94-nop (2003) – nntp:f6c0533b.0308111226.522f7031@posting.google.com

6. Скробов А., реализация MARS на C# (2006) – ftp://cs.usu.edu.ru/corewars/

Применение мультифрактального анализа при изучении петроглифов Карелии

Спиридонов,
Петрозаводский Государственный Университет.
Республика Карелия, г. Петрозаводск, ул. Белорусская д.17 , кв.206
e-mail: spiridonov@psu.karelia.ru

Особенностью петроглифов Карелии является то, что они выбиты на скалах [4]. Глубина выбивки составляет 2 – 3 мм, и по этому большинство из них становятся видными только в косых лучах солнца на закате или на восходе, когда становится видны все неровности скалы. Изображение петроглифов, как правило, образуют сюжетные группы. Однако есть изображения наложенные друг на друга. При их истолковании возникает ряд вопросов, ответы на которые могут дать современные компьютерные технологии и методы распознавания.

В первую очередь это поиск инварианта петроглифа, связанного с анализом его поверхности.

Петроглиф выбивался при помощи соударения более крепкой каменной породы, играющей роль инструмента художника, о скалу. Поэтому бугристость его поверхности, как правило, выше бугристости скалы, хотя может наблюдаться и обратная картина.

Для поиска инварианта и разделения на фотографии изображение петроглифа и скалы был применен метод бинарной сегментации изображений базирующийся на методе мультифрактальной параметризации структур[2].

На сегодняшний день теория мультифракталов довольно популярна в таких областях науки как физика и прикладная математике т.к. данная теория предоставляет искусный математический аппарат для анализа и моделирования природных и искусственных объектов [1]. Модели, полученные при использовании теории мультифракталов более изящны и, часто более адекватно описывают объекты окружающего мира. Чего стоит один, ставший уже классическим, пример с листом папоротника[2].

Одна из основных проблем, с которой сталкиваются ученые в своем исследовании при применении данной теории к практике – это трудность расчета фрактальных характеристик. Чтобы это увидеть наглядно, рассмотрим определение такой характеристики мультифрактала как спектр фрактальных размерностей Рении [2].

[2] Данное исследование поддержано грантом РГНФ № 05-01-12118в (руководитель Н.В. Лобанова).

1) Рассмотрим фрактальный объект, занимающий некую область M размера L в евклидовом пространстве размерностью d.

2) Разобьем всю область M на кубические ячейки со стороной ε и объемов ε^d

3) Рассмотрим величину $p_i(\varepsilon)$ - меру распределения вещества для каждой i - ой ячейки фрактального объекта

4) Введем в рассмотрение обобщенную статистическую сумму $Z(q,\varepsilon)=\sum_{i=1}^{N(\varepsilon)} p_i^q(\varepsilon)$, где $N(\varepsilon)$ - количество не пустых ячеек, $-\infty < q < +\infty$.

5) Спектр фрактальных размерностей Рении D_q, характеризующих распределение мультифрактала в области M определяется с помощью соотношения $D_q = \frac{\tau(q)}{q-1}$.

6) Где функция $\tau(q) = \lim_{\varepsilon \to 0} \frac{\ln Z(q,\varepsilon)}{\ln \varepsilon}$

Таким образом, мультифрактал в общем случае характеризуется нелинейной функцией $\tau(q)$, определяющей поведение статистической суммы при $\varepsilon \to 0$.

Первый вариант метода мультифрактальной параметризации структур был разработан в лаборатории прочности металлических материалов ИМЕТ РАН в 1993 г.[3]. Данная методика позволяет рассчитывать основные фрактальные характеристики фракталов и мультифракталов.

При использовании данной методики был разработан и реализован алгоритм бинарной сегментации изображений петроглифов на поверхности скалы. Приведем его описание:

1) Изображение разбивается на фрагменты $n \times n$ пикселей

2) Для каждого фрагмента считается спектр фрактальных размерностей Рении.

В качестве меры бугристости для каждого фрагмента брался посчитанный на данном фрагменте спектр фрактальных размерностей Рении.

3) Для каждого графика спектра фрактальных размерностей Рении считается коэффициент убывания (был предложен коэффициент, являющийся суммой отношений приращений функции к приращениям ее аргументов для всех точек $q = 0,1,2,3,4,5,10,15,20,30,40$).

4) Все коэффициенты сортируются по возрастанию.

5) Самый минимальный коэффициент относится к 1-ому классу K_1, а самый максимальный к 2-ому классу K_2.

6) Затем, исходя из минимального и максимального значений, находится средний элемент и производится деление всей совокупности коэффициентов убывания на два класса.

После неоднократного анализа различных изображений петроглифов на фоне скалы, были выявлены недостатки схемы анализа изображения, в том первоначальном виде, в каком она приведена выше.

Была выявлена нежелательная закономерность: уменьшение точности классификации с уменьшением размера фрагментов классификации. Этот эффект связан с методом расчета мультифрактальных характеристик, в котором используется такой метод аппроксимации функции как метод среднеквадратичных погрешностей. Как известно этот метод чувствителен к количеству точек, по которым нужно проводить аппроксимацию. Из этого следовало, что достаточно малые для классификации фрагменты нельзя было брать, т.к. это увеличило бы количество неверно расклассифицированных фрагментов. Опытным путем было установлено, что оптимальный размер фрагмента равен 12-ти пикселям. Исходя из этого, результат классификации имеет, так сказать, «угловатый» вид, что конечно не приемлемо.

Также были обнаружены определенные проблемы в критерии классификации фрагментов. Данный критерий, описанный в пунктах 4 – 6, выше приведенного алгоритма, работает только для 5% проанализированных изображений. На данном этапе, за неимением подходящего критерия классификации без участия человека, было принято решение реализовать человеко-машинный алгоритм. Основной смысл этого решения заключается в том, чтобы перед запуском основного алгоритма, обучить программу, задавая ей область петроглифа и область скалы. Реализовать эту идею можно по-разному.

На данном этапе есть один вариант ее реализации, краткое описание которого выглядит так:

1) Указанный фрагмент петроглифа (скалы) разбивается на фрагменты, размер которых равен фрагментам, на которые разбито все изображение.

2) Для каждого фрагмента из области петроглифа (скалы) считается коэффициент убывания.

3) Для данного фрагмента на основании посчитанного на нем коэффициента убывания считается среднее расстояние от всех фрагментов петроглифа (скалы).

4) Фрагмент относится к классу «петроглиф» («скала») если среднее расстояние от него и до фрагментов петроглифа (скалы) меньше чем среднее расстояние от него и до фрагментов скалы (петроглифа).

После введения таких изменений в алгоритм и программу, результаты классификации заметно улучшились. Как уже говорилось, существует не единственная возможность реализации данной идеи. Необходимо провести статистический анализ результатов сегментации для каждого возможного варианта.

Множеством реализаций обладает, так же, сам метод мультифрактальной параметризации структур. Выбор меры распределения вещества $p_i(\varepsilon)$, определенного в определении спектра фрактальных размерностей Рении D_q, не определен самим методом. Потому, что данная характеристика выбирается из неких физических соображений распределения вещества по конкретному типу объектов. Меру можно определить, например, как распределение массы по объекту. В нашем случае, характеристика $p_i(\varepsilon)$ должна некоторым образом характеризовать меру распределения цвета по изображению, что, можно сделать не одним способом.

На данном этапе пока реализована одна из простейших возможностей:

1) Изображение переводится в оттенки серого (256 цветов) путем подсчета среднеарифметического трех компонент R,G,B цветного изображения и приравнивания им этого значения.

2) Затем мера распределения цвета $p_i(\varepsilon)$ для $i-го$ фрагмента со стороной ε будет равна $p_i(\varepsilon)=\frac{\sum_k Z_{ik}}{Z}$, где Z_{ik} - цвета пикселей $i-го$ фрагмента, а Z - сумма цветов всех пикселей изображения.

У этой меры есть определенные недостатки. Из-за того, что изображение переводится в оттенки серого, теряется информативность данного изображения, что не желательно. Поэтому, необходимо искать более подходящие методы расчета мер на изображении.

Учитывая выше приведенные проблемы, с которыми мы столкнулись, представляется следующее развитие данного алгоритма:

1) Необходимо искать методы решения «угловатости» результатов классификации. На этом пути возникает дилемма: Минимальный размер фрагмента, при котором результат классификации приемлем, равен 12-ти пикселям, а в идеале хотелось бы проводить классификацию по пикселям (в крайнем случае, по фрагментам 2×2). Выход из этого положения представляется в следующем виде: для классификации данного пикселя можно рассмотреть квадратный фрагмент размером, например, 12×12 пикселей, внутри которого этот пиксель находится. Затем проводить классификацию по этому фрагменту.

2) Далее развивать идею человеко-машинного алгоритма. В этом направлении представляется следующая возможность по модернизации программы: При классификации фрагментов используется коэффициент убывания, который считается на основе спектров фрактальных размерностей Рении. Можно подобрать более подходящий коэффициент или вообще от него отказаться, проводя классификацию фрагментов по самим спектрам Рении. Если двигаться по пути отказа от коэффициента, то возникает вопрос об оценке увеличении времени потраченного на классификацию фрагментов.

3) Поиск новых мер распределения цвета $p_i(\varepsilon)$ и сравнения их между собой.

Список литературы

1. Kenneth F. Fractal Geometry, UK: WILEY, 2003. 337 с.
2. С. В. Божокин, Д. А. Паршин, Фракталы и мультифракталы, М.: R&C Dynamics, 2001. 128 с.
3. Г. В. Встовский, А. Г. Колмаков и др., Введение в мультифрактальную параметризацию структур материалов, М: R&C Dynamics, 2001. 115 с.
4. Савватеев Ю. А. Наскальные рисунки Карелии Петрозаводск, 1983

Дифференциальные операторы Шимуры на модулярных формах Зигеля в трёхмерном случае

Татауров Павел

Целью доклада является рассказать о более простом методе вычисления дифференциального оператора Шимуры третьего порядка для модулярных форм.

Нам понадобится представление модулярной формы в виде ряда Фурье

$$f(z) = \sum_{\xi \in B_m} c(\xi) e_m(\xi z),$$

где $e_m(\xi z) = \exp(2\pi i \operatorname{tr}(\xi z))$, $B_m = \{\xi \in M_m(\mathbb{R}) | \xi = \xi^t, \xi_{ij}, 2\xi_{ii} \in \mathbb{Z}, \xi \geq 0\}$

Рассмотрим дифференциальный оператор

$$\delta_k(f) = (-\frac{1}{4\pi})^m \det(y)^{-1} \det(z - \overline{z})^{\kappa - k} \Delta(\det(z - \overline{z})^{k - \kappa + 1} f(z)),$$

где $\Delta = \det(\frac{1}{2}(1 + \delta_{ij})\partial_{ij})$ - оператор Мааса, а $\kappa = (m+1)/2$

Определение 1 *Оператором Шимуры называется оператор* $\delta_k^{(r)} = \delta_{k+2r-2} \circ \ldots \circ \delta_k$

Оператор Шимуры переводит модулярную форму веса k в модулярную форму веса $k + 2r$.

Courtieu [4] была доказана следующая теорема

Теорема 1

$$\delta_k^{(r)} f = \sum_{\xi \in B_m} c(\xi) \det(4\pi y)^{-r} R_m(4\pi \xi y; r, \beta) e_m(\xi z),$$

$$\textit{где } R_m(z; r, \beta) = (-1)^{mr} e^{tr(z)} \det(z)^{r+\beta} \Delta_m^r (e^{-tr(z)} \det(z)^{-\beta})$$

Случаи $m = 1$, $m = 2$ были разобраны в докторской диссертации M. Courtieu. Рассмотрим случай $m = 3$

Рассмотрим дифференциальный оператор

$$\Delta_3 = \det \begin{pmatrix} \partial_{11} & \frac{1}{2}\partial_{12} & \frac{1}{2}\partial_{13} \\ \frac{1}{2}\partial_{12} & \partial_{22} & \frac{1}{2}\partial_{23} \\ \frac{1}{2}\partial_{13} & \frac{1}{2}\partial_{23} & \partial_{33} \end{pmatrix} =$$

$$= \partial_{11}\partial_{22}\partial_{33} + \tfrac{1}{4}\partial_{12}\partial_{13}\partial_{23} - \tfrac{1}{4}\partial_{11}\partial_{23}^2 - \tfrac{1}{4}\partial_{22}\partial_{13}^2 - \tfrac{1}{4}\partial_{33}\partial_{12}^2$$

При дифференцировании $e^{\mathrm{tr(z)}} \det(z)^\alpha$ появляются всевозможные комбинации $e^{\mathrm{tr(z)}}$, $\det(z)$, $\mathrm{TR(z)}$ и $\mathrm{tr(z)}$ поэтому для дальнейшего дифференцирования возникает потребность в следующей лемме

Лемма 1

$$\Delta(e^{\mathrm{tr}} \det{}^\alpha \ \mathrm{TR}^\beta \ \mathrm{tr}^\gamma) =$$
$$A_1[\alpha,\beta,\gamma] \det{}^{\alpha+1} \mathrm{TR}^{\beta-1}\mathrm{tr}^\gamma + \det{}^\alpha \Big($$
$$A_2[\alpha,\beta,\gamma]\mathrm{TR}^\beta\mathrm{tr}^\gamma + A_3[\alpha,\beta,\gamma]\mathrm{TR}^\beta\mathrm{tr}^{\gamma-1} + A_4[\alpha,\beta,\gamma]\mathrm{TR}^\beta\mathrm{tr}^{\gamma-2} +$$
$$A_5[\alpha,\beta,\gamma]\mathrm{TR}^{\beta-1}\mathrm{tr}^{\gamma+1} + A_6[\alpha,\beta,\gamma]\mathrm{TR}^{\beta-1}\mathrm{tr}^\gamma + A_7[\alpha,\beta,\gamma]\mathrm{TR}^{\beta-1}\mathrm{tr}^{\gamma-1} +$$
$$A_8[\alpha,\beta,\gamma]\mathrm{TR}^{\beta-2}\mathrm{tr}^{\gamma+2} + A_9[\alpha,\beta,\gamma]\mathrm{TR}^{\beta-2}\mathrm{tr}^{\gamma+1}\Big)$$
$$+ \det{}^{\alpha-1}\Big(A_{10}[\alpha,\beta,\gamma]\mathrm{TR}^{\beta+1}\mathrm{tr}^\gamma + A_{11}[\alpha,\beta,\gamma]\mathrm{TR}^{\beta+1}\mathrm{tr}^{\gamma-1} +$$
$$A_{12}[\alpha,\beta,\gamma]\mathrm{TR}^{\beta+1}\mathrm{tr}^{\gamma-2} + A_{13}[\alpha,\beta,\gamma]\mathrm{TR}^\beta\mathrm{tr}^{\gamma+1} + A_{14}[\alpha,\beta,\gamma]\mathrm{TR}^\beta\mathrm{tr}^\gamma\Big),$$
где

$\det = z_{11}z_{22}z_{33} + 2z_{12}z_{13}z_{23} - z_{11}z_{23}^2 - z_{22}z_{13}^2 - z_{33}z_{12}^2,$
$\mathrm{TR} = z_{11}z_{22} - z_{12}^2 + z_{11}z_{33} - z_{13}^2 + z_{22}z_{33} - z_{23}^2,$
$\mathrm{tr} = z_{11} + z_{22} + z_{33},$
$A_{01}(\alpha,\beta,\gamma) = -\beta(\beta-1)(\beta-2),$
$A_{02}(\alpha,\beta,\gamma) = 1,$
$A_{03}(\alpha,\beta,\gamma) = 3\gamma,$
$A_{04}(\alpha,\beta,\gamma) = 3\gamma(\gamma-1),$
$A_{05}(\alpha,\beta,\gamma) = 2\beta,$
$A_{06}(\alpha,\beta,\gamma) = \frac{7}{2}\beta + 3\alpha\beta + \beta^2 + 4\beta\gamma,$
$A_{07}(\alpha,\beta,\gamma) = \frac{3}{2}\beta\gamma + 3\alpha\beta\gamma + \beta^2\gamma + 2\beta\gamma^2,$
$A_{08}(\alpha,\beta,\gamma) = -\beta(\beta-1),$
$A_{09}(\alpha,\beta,\gamma) = \beta(\beta-1)(\beta+1) - \alpha\beta(\beta-1) + \beta\gamma(\beta-1),$
$A_{10}(\alpha,\beta,\gamma) = \alpha,$
$A_{11}(\alpha,\beta,\gamma) = 2\alpha\gamma,$
$A_{12}(\alpha,\beta,\gamma) = -\alpha\gamma(\gamma-1),$
$A_{13}(\alpha,\beta,\gamma) = \alpha^2 + \alpha(\beta + \frac{1}{2}),$
$A_{14}(\alpha,\beta,\gamma) = \alpha^3 + \alpha^2(\frac{3}{2} + 2\beta + \gamma) + \alpha(\frac{1}{2} + \frac{3}{2}\beta + \beta^2 + \frac{1}{2}\gamma + \beta\gamma)$

Мы получим, что $\Delta^r(e^{\mathrm{tr}}\det{}^{\alpha}) = e^{\mathrm{tr}}\sum_{i=0}^{r}\det{}^{\alpha-i}P_i$, где

$$P_i = \sum_{\beta,\gamma:\ \beta+\gamma\le i} C_{\beta,\gamma}^{(i,r)}\mathrm{TR}^{\beta}\mathrm{tr}^{\gamma}$$

Для подсчёта Δ^r нам надо найти все $C_{\beta,\gamma}^{(i,r)}$ в этой сумме.

Найдём $C_{\beta,\gamma}^{(k,r)}$ - коэффициент при $e^{\mathrm{tr}}\det{}^{\alpha-k}\mathrm{TR}^{\beta}\mathrm{tr}^{\gamma}$, $k \le r$, $\beta+\gamma \le k$.

Мы получим, что он представим в следующем виде
$C_{\beta,\gamma}^{(k,r)} = A_{01}(\alpha-k-1,\beta+3,\gamma)\ C_{\beta+3,\gamma}^{(k+1,r-1)} + A_{02}(\alpha-k,\beta,\gamma)\ C_{\beta,\gamma}^{(k,r-1)} +$
$A_{03}(\alpha-k,\beta,\gamma+1)\ C_{\beta,\gamma+1}^{(k,r-1)} + A_{04}(\alpha-k,\beta,\gamma+2)\ C_{\beta,\gamma+2}^{(k,r-1)} +$
$A_{05}(\alpha-k,\beta+1,\gamma-1)\ C_{\beta+1,\gamma-1}^{(k,r-1)} + A_{06}(\alpha-k,\beta+1,\gamma)\ C_{\beta+1,\gamma}^{(k,r-1)} +$
$A_{07}(\alpha-k,\beta+1,\gamma+1)\ C_{\beta+1,\gamma+1}^{(k,r-1)} + A_{08}(\alpha-k,\beta+2,\gamma-2)\ C_{\beta+2,\gamma-2}^{(k,r-1)} +$
$A_{09}(\alpha-k,\beta+2,\gamma-1)\ C_{\beta+2,\gamma-1}^{(k,r-1)} + A_{10}(\alpha-k+1,\beta-1,\gamma)\ C_{\beta-1,\gamma}^{(k-1,r-1)} +$
$A_{11}(\alpha-k+1,\beta-1,\gamma+1)\ C_{\beta-1,\gamma+1}^{(k-1,r-1)} + A_{12}(\alpha-k+1,\beta-1,\gamma+2)\ C_{\beta-1,\gamma+2}^{(k-1,r-1)} +$
$A_{13}(\alpha-k+1,\beta,\gamma-1)\ C_{\beta,\gamma-1}^{(k-1,r-1)} + A_{14}(\alpha-k+1,\beta,\gamma)\ C_{\beta,\gamma}^{(k-1,r-1)}$

Мы получили формулу для $C_{\beta,\gamma}^{(k,r)}$. Программная реализация данной рекуррентной зависимости позволяет в разы облегчить подсчёт Δ_3, а как следствие и R_3.

Список литературы

1. Panchishkin, A.A., Courtieu M. Non-Archimidean L-Functions and Arithmetical Siegel Modular Forms, Lecture notes in Mathematics 1471, Springer-Verlag, 2004

2. Panchishkin, A.A. Admissible measures for standart L-functions and nearly holomorfic Siegel modular forms. MPI 2002 - 42

3. Panchishkin, A.A. Arithmetical differential operators on nearly holomorfic Siegel modular forms. MPI 2002 -41

4. Courtieu, M. PhD thesis

Особенности моделирования и оптимизация системы обслуживания цеха вулканизации

Фищенков Александр Николаевич,
Ярославский Государственный Технический Университет
г. Ярославль, Московский пр-т, д.88
e-mail: alex123@nordnet.ru

К работе автоматизированных комплексов, обеспечивающих нормальное функционирование вулканизационного производства, предъявляются достаточно высокие требования. Ускорение ритма технологического процесса, широкая номенклатура шин обусловили необходимость определения оптимальной структуры, параметров и схем управления транспортно-технологической системы производства в целом.

Рассматривается модель системы, для которой выполняются следующие условия: во время функционирования системы происходит изменение состояний по заданной программе состояния системы отличаются разной степенью ответной реакции системы на действие внешних и внутренних условий; интервал времени, в течение которого возможно действие условий, превосходит продолжительность отдельных состояний системы.

Схема функционирования системы с динамическим изменением состояний должна отражать следующие особенности: влияние схемы и параметров изменения состояний системы на вероятность прекращения функционирования и изменение характеристик системы; зависимость величин изменения параметров системы от условий вынужденного прекращения функционирования; влияние изменения характеристик системы на последующие этапы функционирования; влияние компонентов системы; взаимное влияние действий внешних условий по групповым средствам системы; единство действия условий второй группы (среды) на функциональную систему и на реагирующий компонент подсистемы обслуживания, который определяет действие условий первой группы.

Для учёта возможностей системы по гибкой подготовке транспортного средства можно выделить также систему коррекции времени отправки транспортного средства (ТС) и систему коррекции ошибок маршрута ТС до отправки

Количество состояний подсистемы обслуживания с различными устойчивостями и законом изменения параметров при действиях определяются конкретной конструктивной схемой системы и её составных элементов, а также схемой их совместного функционирования.

Модель действия условий на систему предназначена для разработки динамической модели функционирования системы с изменением состояний, оценки эффективности системы при обслуживании подсистемы обслуживания с учётом случаев прекращения функционирования подсистем и изменения параметров успешно функционирующих средств системы.

Для оценки вероятности прекращения функционирования средств системы и оценки фактических значений изменяемых параметров функционирующих подсистем обслуживания вся совокупность действий может быть представлена в виде двух совокупностей: совокупность действий по системе и её средствам; часть совокупности действий, вызывающая прекращение функционирования системы или её средств; часть совокупности действий, вызывающая изменение параметров системы или её средств.

Для определения совокупности действий, вызывающих прекращение функционирования системы или её средств, используются ступенчатые пространственные зоны влияния фактора действия условий на систему.

Для систем, функционирующих с отправкой ТС, необходимо учитывать влияние действий подсистемы обслуживания на функционирование ТС. Такое влияние возможно по следующим причинам: подсистема может осуществлять сопровождение ТС или ТС на маршруте движения может находиться в зоне влияния факторов действия условий группы, направленных на подсистему.

Основная особенность задач выбора рациональных параметров схемы гибкого изменения состояний системы при действиях условий сводится к тому, что результаты выбора во многом определяются самими параметрами действий по данной системе. В то же время исходные данные по этим параметрам определены не полностью, зависят от интересов подсистем обслуживания и обеспечиваются наивыгоднейшими с его точки зрения действиями. Искомое решение должно носить инвариантный характер по отношению ко всему диапазону условий применения. В соответствии с принципом сбалансированности при выборе параметров средства системы следует исключить те варианты модели действия, при которых система не выполняет поставленной задачи. В плане компенсации неопределённости схемы действия внешних условий при функционировании средств системы в модели системы предусмотрена система коррекции времени окончания функционирования. Диапазон неопределённости времени окончания функционирования определяется разбросом соотношений потребной и располагаемой (реализуемой на подсистеме) продолжительности действий условий от

момента начала действия до момента окончания функционирования. Таким образом, возникает необходимость разработки методики расчёта требуемой коррекции времени окончания функционирования подсистемы, а также продолжительности состояний для данного варианта компенсационной системы. Основные положение методики сводятся к следующему.

В зависимости от соотношения момента начала действия условий и момента начала функционирования системы(после получения команды) можно выделить:

1) действие условий начинается до начала изменения состояний подсистемы, т.е., $t_i < t_{нф}$,

где $t_{нф}$ – начало функционирования системы (подготовки ТС к отправке) после (i-1) – го действия;

2) действие условий начинается после начала функционирования подсистемы, т.е. $t_i \geq t_{нф}$.

Разрешённый момент отправки транспортного средства после i-го действия условий определяется зависимостью

$$t_{отп}^{треб} = \begin{cases} t_i + \tau_{сп}^{треб} \text{ при } & t_i + \tau_{сп_i}^{треб} > t_{отп_{i-1}}, \\ t_{отп_{i-1}} \text{ при } & t_i + \tau_{сп_i}^{треб} \leq t_{отп_{i-1}}, \end{cases}$$

где $\tau_{сп_i}^{треб}$ - требуемое время ожидания спала уровня до отправки транспортного средства после i-го действия условий.

В общем случае различие между требуемой и располагаемой задержками отправки ТС с момента действия до момента отправки определяется такими конкретными характеристиками компенсационной системы подсистемы обслуживания как располагаемая информация о действиях к моменту принятия решения о коррекции момента отправки ТС; достоверность прогноза внешней обстановки по маршруту ТС при выборе продолжительности коррекции; диапазон регулирования (коррекции) начала подготовки, прерывания подготовки, момента отправки относительно номинального (с учётом состояния подсистемы в момент принятия решения).

При исследовании эффективности с целью выбора рациональных параметров на этапе предварительного проектирования средств системы построение зависимостей затруднено ввиду необходимости подключения большого числа в специализированных подсистем по расчётам устойчивости множества предварительных вариантов системы и её средств к действиям условий.

В качестве наиболее общих составляющих задачи выбора параметров средств системы можно выделить сравнительную оценку эффективности вариантов проектируемого элемента, оценку влияния параметров на эффективность, оптимизацию параметров из условия максимальной эффективности с учётом ограничений материально-технических ресурсов.

Для суждения множества предварительных вариантов системы в условиях неопределённости особое место среди методов сравнительной оценки занимают косвенные методы, инвариантные к конкретным характеристикам действия усилий в заданном диапазоне.

Рассмотрим модель функционирования и методику оценки характеристик условий первой группы с учётом прекращения функционирования компонентов подсистемы обслуживания, обусловленного действием условий второй группы, т.е. с учётом влияния действий других подсистем на данную. Учёт прекращения функционирования необходим для соблюдения единства модели внешних условий для средств системы и средств подсистемы.

Для анализа распределения количества и интервалов действий подсистемы, а также расчёта вероятности осуществления действий удобно построить “дерево” исходов.

Математическая модель синтеза подсистемы:

Требуется определить матрицу $X_0 = \left\| x_{ji}^0 \right\|$, доставляющую максимум вида:

$$\psi(\mathrm{X}) = \sum_{i=1}^{S} A_j \left(1 - \prod_{j=1}^{m} \varepsilon_{ji}^{x_{ji}} \right) \qquad (1)$$

при линейном ограничении на переменные

$$\sum_{j=1}^{m} \sum_{i=1}^{s} a_{ji} x_{ji} \leq b \qquad (2)$$

и дополнительных условиях:

$$\left.\begin{array}{c} x_{ji} \in \{0,1,\ldots\}, \\ 0 \leq (\varepsilon_{ji} = 1 - \omega_{ji}) \leq 1, \\ A_i > 0, a_{ji} \geq 0, \end{array}\right\} \quad i = 1,\ldots,S \qquad (i \in I_S),\ j = 1,\ldots,m. \qquad (3)$$

Величина a_{ji} в этом случае рассматривается как стоимость затрат одного образца j-го типа. Через величины a_j и x_j будем обозначать соответственно полные затраты и общее количество задействованных средств j-го типа, вычисляя их по формулам:

$$a_j = \sum_{i=1}^{S} a_{ji}, x_{ji}, \qquad x_j = \sum_{i=1}^{S} x_{ji}, \qquad j = 1,..., m. \tag{4}$$

Состав отобранных средств при этом может быть представлен в виде $\overline{X} = \{x_j\} m.$

Общий ресурс, выделенный для организации по i-му объекту, обозначим через b_i:

$$b_i = \sum_{j=1}^{m} a_{ji}, x_{ji} \qquad \left(b = \sum_i b_i \right), \qquad i = 1,..., m. \tag{5}$$

Поскольку целевая функция не убывает, а ограничение линейно, то решение удет выполняться при знаке = в ограничении (2) с точностью, определяемой дискретностью величин a_{ji}.

Метод и алгоритм решения:

Первым интуитивным движением, которое возникает на основе знаний материала, является попытка решить задачу методом максимального элемента, предварительно вычислив матрицу $\|\upsilon_{ji}\|$, где

$$\upsilon_{ji} = \frac{\Delta^*_{ji}}{a_{ji}} = \frac{A_i \omega_{ji}}{a_{ji}}, \qquad i = 1,..., S,\ j = 1,..., m. \tag{6}$$

Элемент υ_{ji} по своему смыслу означает прирост целевой функции на единицу затрат. Такой критерий в условиях применения его в неизменном виде обеспечивает лишь приближённое решение.

Если известны условия оптимальной замены средств, то может быть определена зависимость изменения оптимального состава средств и максимального уровня достигаемого эффекта от величины бюджета b_i, выделяемого для организации воздействия по i-му объекту $\varphi_i(b_i)$. При наличии таких функций исходная задача может быть записана как задача максимизации функции

$$\varphi(\overline{b}) = \sum_{i=1}^{S} \varphi_i(b_i) \tag{7}$$

путём определения вектора $\overline{b} = \{b_i\} s$, компоненты которого удовлетворяют условиям:

$$\sum_{i=1}^{S} b_i \le b, \qquad b_i \ge 0, \qquad i = 1,..., S. \tag{8}$$

В целом, порядок решения следующий.

Алгоритм 1.1.

1. Вычислить элементы матрицы $\|\upsilon_{ji}\|$ по формуле (6).
2. Согласно принципу доминирования исключить из рассмотрения “лишние” элементы матрицы $\|\upsilon_{ji}\|$.
3. Последовательно, наращивая бюджет $b_i \le b$ и применяя условие, определить зависимости $\varphi_i(b_i)$ для i=1,….,S.
4. Определить вектор $\overline{b}_0 = \{b_i^0\}$, достигающий максимум функции (7)
5. Записать решение, прекратить вычисления.

Таким образом, предложены модели, учитывающие динамические возможные состояния системы на этапе функционирования. Для решения задачи эффективности транспортной системы предлагается динамическая модель, позволяющая осуществлять выбор программы изменения состояний системы. Вероятность успешного функционирования подсистем учитывает общую эффективность системы.

Предложены математические модели, описывающие работу подсистем транспортной системы и постановка задачи оптимизации.

Случайное расстояние и его применения в дешифрировании изображений

Фофанов В. Б., *доцент,*
420008, Казань, ул.Кремлевская, д.18, Казанский государственный университет,
Email: Viatcheslav.Fofanov@ksu.ru

В большинстве работ, посвященных дешифрированию изображений, сцена рассматривается как совокупность $(\xi_z)_{z\in Z^2}$ случайных величин, сопоставленных узлам целочисленной решетки. Случайные величины определены на вероятностном пространстве (Ω, A, P) и принимают значения в конечном множестве Y. Объект сцены определяется конечным подмножеством $A \subset Z^2$, называемым его проекцией, и семейством $\xi_A = (\xi_a)_{a\in A}$. Изображением сцены называется отображение $x: Z^2 \to Y$ вида $x_z = \xi_z(\omega)$, $\omega \in \Omega$, а изображением x_A объекта ξ_A - сужение x на A.

Современные технические средства позволяют одновременно измерять в каждом пикселе $\nu \geq 1$ свойств. В этих условиях пиксель с координатами z удобно описывать ν-мерной случайной величиной $\xi_z = (\xi_z^j)_{1\leq j\leq \nu}$, а в качестве изображения сцены рассматривается векторное отображение $x: Z^2 \to Y^\nu$ вида $x_z = (x_z^j)_{1\leq j\leq \nu}$. Непосредственный перенос методов анализа скалярных изображений на общий случай, когда $\nu \geq 1$, наталкивается на определенные трудности. Некоторые из них можно преодолеть, если воспользоваться понятием случайного расстояния.

Расстояние d на R^ν назовем Борелевским, если Борелевским является отображение $d: R^\nu \times R^\nu \to R$. Из свойств измеримых функций следует, что расстояние Евклида и другие, часто используемые расстояния, являются борелевскими.

Пусть d - борелевское расстояние на R^ν, а $\xi = (\xi_j)_{1\leq j\leq \nu}$ и $\eta = (\eta_j)_{1\leq j\leq \nu}$ - векторные случайные величины на вероятностном пространстве (Ω, A, P). В этом случае отображение $d(\xi,\eta): \Omega \to R$, определяемое равенством

$$d(\xi,\eta)(\omega) = d(\xi(\omega),\eta(\omega)),$$

будет случайной величиной на (Ω, A, P). При каждом $\omega \in \Omega$ будут выполняться аксиомы расстояния. Это позволяет назвать d случайным расстоянием на множестве ν-мерных случайных величин, а $d(\xi,\eta)$ - случайным расстоянием между ξ и η.

Пусть $\xi = (\xi_j)_{1\leq j\leq \nu}$ - векторная случайная величина и $m_\xi = (m_\xi^j)_{1\leq j\leq \nu}$ - вектор средних значений ее случайных компонент. Назовем дисперсией ξ неотрицательное число $D\xi$ вида $D\xi = Ed^2(\xi, m_\xi)$. С помощью этого определения удается установить целый ряд полезных свойств, включая знаменитое неравенство Чебышева.

Утверждение 1. *Если* $\xi = (\xi_j)_{1\leq j\leq \nu}$ *- векторная случайная величина с конечной дисперсией* $\sigma_\xi^2 = D\xi$, *то при любом вещественном* $k > 0$ *выполняется неравенство*

$$P(d(\xi, m_\xi) \geq k\sigma_\xi) \leq \frac{1}{k^2}.$$

Следующее утверждение позволяет оценить вероятность отклонения случайного расстояния $d(\xi,\eta)$

от расстояния $d(m_\xi, m_\eta)$ между средними значениями.

Утверждение 2. *Пусть* ξ *и* η *- векторные случайные величины,* m_ξ *и* m_η *- их векторы средних значений, а* σ_ξ^2 *и* σ_η^2 *- их конечные дисперсии. Тогда для любого вещественного* $k > 0$ *выполняется неравенство* $P(|\, d(\xi,\eta) - d(m_\xi, m_\eta)| \geq k(\sigma_\xi + \sigma_\eta)) \leq \frac{1}{k^2}$.

Далее в качестве d будет использоваться Евклидово расстояние. Из его определения тотчас следует, что $D\xi = D\xi_1 + ... + D\xi_\nu$.

Утверждение 3. *Если* ξ *и* η *- независимые векторные случайные величины,* $a \in R$ *и* $b \in R$, *то*

$$D(a\xi + b\eta) = a^2 D\xi + b^2 D\eta .$$

Пусть $\sigma_\xi^2 = D\xi$. Легко проверить непосредственно, что для любого вектора $m = (m_j)_{1 \leq j \leq \nu}$ из R^ν имеет место равенство $Ed^2(\xi, m) = \sigma_\xi^2 + d^2(m_\xi, m)$.

Из него тотчас следует неравенство $Ed^2(\xi, m) \geq \sigma_\xi^2$, обобщающее на векторные случайные величины важное свойство дисперсии скалярных случайных величин.

Если ξ и η независимы, то $Ed^2(\xi,\eta) = \sigma_\xi^2 + d^2(m_\xi, m_\eta) + \sigma_\eta^2$.

Будем говорить, что последовательность $\xi_n = (\xi_{n,j})_{1 \leq j \leq \nu}$, $n \geq 1$, векторных случайных величин сходится по вероятности к векторной случайной величине $\xi = (\xi_j)_{1 \leq j \leq \nu}$ и писать $\xi_n \to \xi$, если последовательность $(d(\xi_n, \xi))_{n \geq 1}$ скалярных случайных величин сходится по вероятности к нулю.

Утверждение 4. *Пусть* $(\xi_n)_{n \geq 1}$ *- последовательность независимых векторных случайных величин таких, что* $E\xi_n = m$ *и* $D\xi_n = \sigma^2$ *для любого* $n \geq 1$. *Тогда последовательность* $(\eta_n)_{n \geq 1}$ *вида* $\eta_n = (\xi_1 + ... + \xi_\nu)/n$ *сходится по вероятности к* m *при* $n \to +\infty$.

Рассмотрим применение случайного расстояния для решения двух задач, возникающих в ходе дешифрирования изображений с целью выявления на сцене заданных объектов по их геометрическим признакам. Для вычисления геометрических признаков объекта необходимо определить его проекцию. Эту задачу, называемую сегментацией, решить значительно легче для фрагмента сцены, содержащего один заданный объект и его некоторое окружение.

Далее изображение сцены будет использоваться в качестве случайной выборки. Поэтому будем предполагать, что случайные величины, образующие сцену, взаимно независимы, а случайные величины, образующие объект, имеют одно и тоже распределение. Пусть $C \subset Z^2$ - квадрат, назовем его границей $Fr(C)$ совокупность точек из C, имеющих соседей в $Z^2 \setminus C$. Семейство случайных величин $\xi_C = (\xi_z)_{z \in C}$ назовем зоной интереса объекта ξ_A с вектором $m_A = (m_A^j)_{1 \leq j \leq \nu}$ средних значений, если $A \subset C \setminus Fr(C)$, если все ξ_z, $z \in C \setminus A$, имеют одно и то же распределение с вектором $m_{C \setminus A} = (m_{C \setminus A}^j)_{1 \leq j \leq \nu}$ средних значений и если $d(m_A, m_{C \setminus A}) > 0$. В качестве первой задачи целесообразно рассмотреть поиск зон интереса. Сегментация и все остальные операции выполняются только в них. Из определения зоны следует, что

$$d(E\xi_a, E\xi_z) = d(m_A, m_{C \setminus A}) > 0, \; a \in A, \; z \in Fr(C) .$$

С другой стороны, если все случайные величины, образующие ξ_C, имеют одно и то же распределение с вектором средних m_C, то ξ_C зоной интереса не является. Более того, $d(E\xi_t, E\xi_z) = d(m_C, m_C) = 0, \; t, z \in C$.

Указанные различия между зоной интереса и пустым квадратом, можно использовать для замены

перебора зон интереса их поиском. Пусть $B(a,r) \subset A$ - круг с центром a и радиусом r. При определенных условиях среднее арифметическое вида

$$\bar{x}_a = \frac{1}{|B(a,r)|} \sum_{t \in B(a,r)} x_t$$

будет оценкой неизвестного вектора m_A. Разделим $Fr(C)$ на s частей Fr_j, $1 \le j \le s$, по $n = |B(a,r)|$ точек в каждой. Тогда

$$\bar{x}_{Fr} = \frac{1}{|Fr(C)|} \sum_{t \in Fr(C)} x_t \quad \text{и} \quad \bar{x}_j = \frac{1}{|Fr_j|} \sum_{t \in Fr_j} x_t \ , \ 1 \le j \le s ,$$

будут оценками неизвестного вектора $m_{C \setminus A}$. Используя сформулированные выше свойства векторных случайных величин, можно показать, что для зоны интереса ξ_C

$$P(d(\bar{x}_a, \bar{x}_{Fr}) > d(\bar{x}_j, \bar{x}_{Fr}), 1 \le j \le s) \to 1 \qquad (1)$$

при $r \to +\infty$. С другой стороны, если ξ_C не является зоной интереса , то

$$P(d(\bar{x}_z, \bar{x}_{Fr}) > d(\bar{x}_j, \bar{x}_{Fr}), 1 \le j \le s) \to 0 , \ B(z,r) \subset C \setminus Fr(C) \text{ , при } S \to +\infty .$$

Два последних утверждения позволяют построить решающее правило классификации квадратных фрагментов ξ_C сцены по ее векторному изображению. Пусть $d(A)$ - известный диаметр объекта. Квадрат ξ_C с центром $a \in Z^2$ и стороной $2d(A)+1$ следует считать зоной интереса, если выполняется (1). Правило обеспечивает вероятность правильного обнаружения зоны равную, практически, единице и одновременно позволяет снизить вероятность ложного обнаружения за счет выбора S.

Это же правило можно использовать для классификации пикселей зоны на два класса. Пиксель $z \in C \setminus Fr(C)$ следует отнести к A, если для него справедливо (1).

Отметим, что случайное расстояние позволяет естественным образом обобщить известные методы мод и квантилей сегментации скалярных изображений на общий случай, когда исходной информацией о сцене служит ее векторное изображение.

Байесовская сегментация изображений

Фофанов В. Б., *доцент*,
Сабиров Р. Г.,
420008, Казань, ул.Кремлевская, д.18,
Казанский государственный университет,
Email: RealRS@mail.ru, Viatcheslav.Fofanov@ksu.ru

Представим сцену как множество элементов, называемых далее пикселями. Пиксель характеризуется координатами $z = (z_1, z_2)$ на целочисленной решетке $Z^2 = \{z = (z_1, z_2) : z_1 \in Z, z_2 \in Z\}$ и случайной величиной ξ_z со значениями из множества $Y = \{0, 1, ..., |Y|-1\}$, $|Y| > 1$, описывающей его некоторое свойство. Все случайные величины определены на вероятностном пространстве (Ω, A, P) и взаимно независимы. Каждый элемент сцены (объект) задается конечным подмножеством $A \subset Z^2$ - его проекцией и семейством $\xi_A = (\xi_a)_{a \in A}$ случайных величин. Проекции A и B разных объектов не пересекаются. Случайные величины, образующие объект ξ_A, имеют одно и то же распределение. Пусть $\omega \in \Omega$, а $x_z = \xi_z(\omega)$, $z \in Z^2$. Отображение $Z^2 \to Y$

вида $x=(x_z)_{z\in Z^2}$ называется изображением сцены, а его сужение x_A на A - изображением объекта ξ_A.

Как известно, сегментацией называется задача о разбиении сцены на составляющие ее объекты по одному или нескольким изображениям. Если целью дешифрирования является поиск на сцене заданных объектов по их геометрическим признакам, то можно ограничиться сегментированием только тех ее участков, которые содержат по одному заданному объекту и некоторому его окружению. Такие участки называются зонами интереса. Более строго, пусть $C\subset Z^2$ - квадрат и $Fr(C)$ - его граница, состоящая из точек C, которые имеют соседей в $Z^2\setminus C$. Семейство $\xi_C=(\xi_z)_{z\in C}$ случайных величин называется зоной интереса для объекта ξ_A с распределением $P_A=(p_A(y))_{y\in Y}$, если $A\subset C\setminus Fr(C)$ и если случайные величины ξ_z, $z\in C\setminus A$, имеют одно и то же распределение $P_{C\setminus A}=(p_{C\setminus A}(y))_{y\in Y}$, отличающееся от P_A. Покажем, что при известном изображении $x_C=(x_c)_{c\in C}$ зоны интереса задачу сегментации можно сформулировать как задачу Байесовской классификации ее пикселей.

Если Y_A - некоторое подмножество Y, а $Y_{C\setminus A}=Y\setminus Y_A$, то отображение $h:Y\to\{1,2\}$, определяемое равенством вида

$$h(y)=\begin{cases}1, y\in Y_A\\ 2, y\in Y_{C\setminus A}\end{cases}, \qquad (1)$$

является решающим правилом для классификации пикселей зоны интереса на два класса. Пиксель с координатами $z\in C$ относят к объекту ξ_A (класс 1), если $x_z\in Y_A$, и к его окрестности $\xi_{C\setminus A}$ (класс 2) в противном случае.

Для построения Байесовского решающего правила h^* с минимальной вероятностью $e(h^*)$ ошибки классификации требуются априорные вероятности $P(A)$ и $P(C\setminus A)$ классов и распределения $P_A=(p_A(y))_{y\in Y}$ и $P_{C\setminus A}=(p_{C\setminus A}(y))_{y\in Y}$ признаков. Как правило, в явном виде эти сведения при решении конкретных задач отсутствуют. Однако, при сделанных предположениях о свойствах случайного поля, информацию об априорных вероятностях $P(A)$ и $P(C\setminus A)$ классов и их распределениях P_A и $P_{C\setminus A}$ можно извлечь из изображения x_C зоны интереса. Его рассматривают в качестве выборки объема $|C|$. Предполагается, что площадь $|A|$ заданного объекта известна, а площадь $|C|$ построенной зоны легко найти. Поэтому

$$P(A)=\frac{|A|}{|C|},\ P(C\setminus A)=\frac{|C\setminus A|}{|C|}.$$

Далее, для каждого $y\in Y$ обозначим символами $n_A(y)$ и $n_{C\setminus A}(y)$ количество пикселей объекта и его окрестности, соответственно, со значениями y, присутствующих на изображении x_C. Известно, что относительные частоты

$$\bar{p}_A(y)=\frac{n_A(y)}{|A|}\ \text{и}\ \bar{p}_{C\setminus A}(y)=\frac{n_{C\setminus A}(y)}{|C\setminus A|},\ y\in Y,$$

являются несмещенными и состоятельными оценками неизвестных вероятностей $p_A(y)$ и $p_{C\setminus A}(y)$, образующих распределения P_A и $P_{C\setminus A}$ классов. Для каждого $y\in Y$ по изображению x_C зоны интереса можно подсчитать количество $n_C(y)$ пикселей со значением y. Очевидно, что $n_C(y)=n_A(y)+n_{C\setminus A}(y)$, $y\in Y$. Если определить $\bar{p}_C(y)$ в виде

$$\bar{p}_C(y)=\frac{n_C(y)}{|C|},\ y\in Y,$$

то получим равенство $\bar{p}_C(y)=P(A)\bar{p}_A(y)+P(C\setminus A)\bar{p}_{C\setminus A}(y)$, $y\in Y$, с двумя неизвестными $\bar{p}_A(y)$ и $\bar{p}_{C\setminus A}(y)$. По определению границы зоны интереса, ее изображение $x_{Fr(C)}$ можно использовать для

получения оценок $\bar{p}_{Fr(C)}(y)=\frac{n_{Fr(C)}(y)}{|Fr(C)|}, y\in Y$,

неизвестных вероятностей $p_{C\setminus A}(y)$. После замены в предыдущем уравнении неизвестной оценки $\bar{p}_{C\setminus A}(y)$ на оценку $\bar{p}_{Fr(C)}(y)$ оно превращается в уравнение с одним неизвестным. Для каждого $y\in Y$ его решением является оценка $\bar{p}_A(y)$ неизвестной вероятности $p_A(y)$. Таким образом, вместо неизвестных распределений P_A и $P_{C\setminus A}$, необходимых для классификации пикселей зоны интереса, можно воспользоваться распределениями $\bar{P}_A=(\bar{p}_A(y))_{y\in Y}$ и $\bar{P}_{C\setminus A}=(\bar{p}_{C\setminus A}(y))_{y\in Y}$, полученными по изображению x_C зоны.

Сказанное позволяет предложить для классификации пикселей зоны интереса решающее правило $\bar{h}^*:Y\to\{1,2\}$, вероятность ошибки $e(\bar{h}^*)$ которого определяется априорными вероятностями $P(A)$ и $P(C\setminus A)$ классов и распределениями $\bar{P}_A$ и $\bar{P}_{C\setminus A}$. Другими словами, справедливо следующее утверждение.

Пусть $P(A)=\frac{|A|}{|C|}$ и $P(C\setminus A)=\frac{|C\setminus A|}{|C|}$ - априорные вероятности, $\bar{P}_A=(\bar{p}_A(y))_{y\in Y}$ и $\bar{P}_{C\setminus A}=(\bar{p}_{C\setminus A}(y))_{y\in Y}$ - распределения признаков, описывающих объект и его окрестность, а Y_A и $Y_{C\setminus A}$ - подмножества из Y, определяемые равенствами вида

$$Y_A=\{y\in Y:|A|\,\bar{p}_A(y)\ge|C\setminus A|\,\bar{p}_{C\setminus A}(y)\}\ \text{и}\ Y_{C\setminus A}=Y\setminus Y_A.$$

Тогда решающее правило $\bar{h}^*:Y\to\{1,2\}$ вида (1) является Байесовским с вероятностью ошибки $e(\bar{h}^*)$, определяемой равенством

$$e(\bar{h}^*)=\frac{|A|}{|C|}\sum_{y\in Y_{C\setminus A}}\bar{p}_A(y)+\frac{|C\setminus A|}{|C|}\sum_{y\in Y_A}\bar{p}_{C\setminus A}(y).$$

Сформулированное утверждение немедленно следует из теоремы Байеса, с учетом вида априорных вероятностей и распределений признаков обоих классов.

Абсолютная погрешность ε оценок, образующих распределение $\bar{P}_{Fr(C)}$, зависит от объема $|Fr(C)|$ выборки. Используя неравенство Чебышева, легко установить, что для уменьшения ε в t раз требуется увеличить $|Fr(C)|$ в t^2 раз.

Для иллюстрации метода воспользуемся изображением фрагмента сцены, проекция которой является квадратом со стороной равной 256 пикселям. Сцена состоит из пяти элементов. Четыре элемента являются прямоугольниками и называются объектами. Пиксели j-ого объекта имеют нормальное распределение с параметрами (m_j,σ_j^2). Все остальные пиксели образуют пятый элемент, который называется фоном. Пиксели фона так же имеют нормальное распределение с параметрами (m_0,σ_0^2). Мерой качества изображения зоны интереса служит отношение сигнал/шум $k_j=\frac{|m_0-m_j|}{\sigma_0+\sigma_j}$, $1\le j\le 4$.

Относительная частота ошибки классификации (сегментации) пикселей каждой зоны интереса с использованием двух выборок разного объема n представлена в таблице. Во второй строке приведены теоретические вероятности ошибки классификации.

k	0.5	1.0	1.5	2.0
$e(h)$	0.308	0.159	0.067	0.023
$n=196$	0.395	0,220	0.146	0.070
$n=5440$	0.355	0.161	0.076	0.027

Как и ожидалось, относительная частота ошибки при увеличении объема выборки приближается к теоретическому значению, а с ростом k убывает. Поэтому k можно использовать для оценки пригодности изображения к сегментации.

Об одном подходе к поиску зон интереса

Фофанов В. Б., *доцент,*
Демченко А. В.,
420008, Казань, ул.Кремлевская, д.18,
Казанский государственный университет,
Email: Viatcheslav.Fofanov@ksu.ru

Довольно часто целью дешифрирования изображений является выявление на сцене заданных объектов по их геометрическим признакам. В этом случае для экономии вычислительных ресурсов дешифрирование целесообразно проводить в два этапа. Вначале за счет простых и поэтому экономных методов осуществляется поиск участков сцены, содержащих возможные объекты. Такие участки называются зонами интереса. Детальный анализ, включающий сегментацию, вычисление признаков и принятие решения о наличии объекта, проводится только в найденных зонах интереса. Решение задачи поиска зон интереса для случая, когда сцена описывается семейством независимых скалярных случайных величин $(\xi_z)_{z\in Z^2}$ со значениями в конечном множестве Y, предложено в работе [1]. В настоящей работе рассматривается общий случай, когда для описания сцены используется семейство независимых ν-мерных случайных величин $\xi_z = (\xi_z^j)_{1\le j\le \nu}$ со значениями в Y^ν.

Сцену удобно рассматривать как совокупность заранее определенных ее элементов (объектов). Каждый объект задается своей проекцией A, которая является конечным связным подмножеством Z^2 и семейством $\xi_A = (\xi_a)_{a\in A}$ одинаково распределенных векторных случайных величин с вектором средних значений $m_A = (m_A^j)_{1\le j\le \nu}$ и вектором дисперсий $\sigma_A^2 = ((\sigma_A^j)^2)_{1\le j\le \nu}$. Предполагается, что проекции разных объектов не пересекаются. Пусть $d(A)$ - диаметр объекта. Для любого $l \ge d(A)+2$ существует квадрат C на Z^2 со стороной равной l такой, что $A \subset (C \setminus Fr(\mathrm{C}))$, здесь $Fr(C)$ - граница квадрата, состоящая из принадлежащих ему точек, каждая из которых имеет соседей из $Z^2 \setminus C$. Будем называть семейство случайных величин вида $\xi_C = (\xi_c)_{c\in C}$ зоной интереса объекта $\xi_A = (\xi_a)_{a\in A}$, если все случайные величины ξ_c, $c \in C \setminus A$, имеют одно и то же распределение с вектором средних значений $m_{C\setminus A} = (m_{C\setminus A}^j)_{1\le j\le \nu}$ и вектором дисперсий $\sigma_{C\setminus A}^2 = ((\sigma_{C\setminus A}^j)^2)_{1\le j\le \nu}$ и если $d(m_A, m_{C\setminus A}) > 0$.

Из определения зоны интереса следует, что каждый ее пиксель может принадлежать объекту ξ_A или его окрестности $\xi_{C\setminus A}$. Если $a \in A$, то $d(E\xi_a, m_{C\setminus A}) > 0$. В противном случае, когда $z \in C \setminus A$, получаем $d(E\xi_z, m_{C\setminus A}) = 0$. Покажем, как эти свойства можно использовать для поиска зон интереса.

Пусть $\omega \in \Omega$. Для каждого $z \in Z^2$ назовем изображением пикселя ξ_z вектор $x_z = (x_z^j)_{1\le j\le \nu}$ с координатами $x_z^j = \xi_z^j(\omega)$. Соответственно изображением x_A объекта будет называться набор изображений его пикселей $x_A = (x_a)_{a\in A}$. Пусть $B(a,r) \subset A$ - круг с центром a и радиусом r. Очевидно, что среднее арифметическое значение $\bar{x}_a$, определяемое равенством

$$\bar{x}_a = \frac{1}{|B(a,r)|}\sum_{z\in B(a,r)} x_z,$$

является оценкой для вектора средних значений m_A. Построим на Z^2 квадрат C с центром в a и стороной $2(d(A)+1)$. В этом случае проекция A объекта ξ_A независимо от его ориентации не

пересекается с границей $Fr(C)$. Разобьем границу $Fr(C)$ на s непересекающихся частей Fr_j, $1 \le j \le s$, по $n = |B(a,r)|$ пикселей в каждой. Если квадрат C является зоной интереса, то каждое из $s+1$ значения вида

$$\bar{x}_{Fr} = \frac{1}{n} \sum_{z \in Fr(C)} x_z \ , \ \bar{x}_j = \frac{1}{n} \sum_{z \in Fr_j} x_z \ , \ 1 \le j \le s \ ,$$

является оценкой вектора средних значений $m_{C \setminus A}$ фона. Из [2] (теорема 4 и 2) следует, что скалярная случайная величина $d(\bar{x}_a, \bar{x}_{Fr})$ сходится по вероятности к $d(m_A, m_{C \setminus A})$ при $r \to +\infty$. Это означает, что при больших значениях r для нее выполняется приближенное равенство $d(\bar{x}_a, \bar{x}_{Fr}) \approx d(m_A, m_{C \setminus A}) > 0$.

Аналогично, для каждого круга $B(z,r)$ из $C \setminus A$ скалярная случайная величина $d(\bar{x}_z, \bar{x}_{Fr})$ сходится по вероятности к $d(m_{C \setminus A}, m_{C \setminus A}) = 0$ при $r \to +\infty$, а при больших значениях r для нее будет выполняться приближенное равенство $d(\bar{x}_z, \bar{x}_{Fr}) \approx 0$.

Сформулированные результаты подсказывают вид решающего правила h, обеспечивающего классификацию квадратных участков сцены на два класса. Один класс состоит только из зон интереса, другой – из квадратов, образованных пикселями фона (не содержащих объектов). В самом деле, если $d(\bar{x}_a, \bar{x}_{Fr}) > d(\bar{x}_{Fr}, \bar{x}_k)$, $1 \le k \le s$, то квадрат с центром a и стороной $2(d(A)+1)$ следует считать зоной интереса. В противном случае этот квадрат следует считать пустым.

В качестве обоснования предлагаемого решающего правила можно сослаться на следующие два утверждения, доказательство которых опирается на свойства случайных расстояний, установленных в [2].

Во-первых, при больших значениях r имеет место приближенное равенство $P(d(\bar{x}_a, \bar{x}_{Fr}) > d(\bar{x}_{Fr}, \bar{x}_k), 1 \le k \le s) \approx 1$, означающее, что пропуск зон интереса, практически, невозможен.

Во-вторых, $P(d(\bar{x}_z, \bar{x}_{Fr}) > d(\bar{x}_{Fr}, \bar{x}_k), 1 \le k \le s) \to 0$. Последнее утверждение означает, что вероятность ложного обнаружения зоны интереса убывает с ростом числа s фрагментов, на которые разбивается граница зоны.

Для применения изложенного метода поиска зон интереса, необходимо научиться выбирать для каждого объекта ξ_A окрестность $B(a,r)$, содержащуюся в его проекции A. Она используется для вычисления оценки $\bar{x}_a$ неизвестного среднего значения m_A. Покажем, как это можно сделать.

Пусть A - проекция заданного объекта с неизвестными координатами и ориентацией, $2r_{\max}$ - сторона квадрата решетки Z^2 с наибольшей площадью, принадлежащего A и $2r$ - сторона окрестности квадратной формы, используемой при сглаживании, $1 \le r \le r_{\max}$. Выберем произвольную точку $z_0 \in Z^2$ и зададим на Z^2 семейство квадратов со сторонами равными $2r$ и координатами z левых верхних вершин, определяемыми равенством

$$z = z_0 + i\,\Delta e_1 + j\,\Delta e_2 \ , \ i \in Z \ , \ j \in Z \ , \ 1 \le \Delta \le 2(r_{\max} - r) \ .$$

Можно показать, что всегда существует квадрат $B(a,r)$ из построенного выше семейства такой, что $B(a,r) \subset A$.

Список литературы

1. S. A. Burylin and V. B. Fofanov. Generalized Spot Criterion as Applied for Image Deciphering. “Pattern Recognition and Image Analysis”, Vol. 14, No. 2, 2004, pp. 243-248.

2. Фофанов В. Б. Определение и свойства случайных расстояний. В настоящем документе.

Сегментация зон интереса на основе случайного расстояния

Фофанов В. Б. , *доцент,*
Кулеев Р. Ф..,
420008, Казань, ул.Кремлевская, д.18,
Казанский государственный университет,
Email: RamilKuleev@mail.ru, Viatcheslav.Fofanov@ksu.ru

При распознавании объектов по геометрическим признакам одним из этапов дешифрирования является сегментация. Используя случайное расстояние, известные методы сегментации сцены по ее единственному изображению можно обобщить на случай, когда исходной информацией о ней служит набор изображений, сформированных одновременно, в частности, в разных спектральных зонах оптического диапазона.

Будем рассматривать сцену как множество пикселей. Пиксель определяется своими координатами $z=(z_1,z_2)\in Z^2$ на целочисленной решетке и ν-мерной случайной величиной $\xi_z=(\xi_z^j)$, $1\le j\le\nu$. Все ее координаты определены на (Ω,A,P) и принимают значения в конечном множестве Y. Объектом сцены назовем пару, состоящую из конечного подмножества $A\subset Z^2$, называемого его проекцией, и семейства $\xi_A=(\xi_a)_{a\in A}$. Пусть $\omega\in\Omega$. Для каждого $z\in Z^2$ и j, $1\le j\le\nu$, можно вычислить $x_z^j=\xi_z^j(\omega)$. Изображением сцены назовем векторное отображение $x:Z^2\to Y^\nu$ вида $x_z=(x_z^j)_{1\le j\le\nu}$, а изображением x_A объекта ξ_A - сужение x на A.

Далее изображение сцены будет использоваться в качестве случайной выборки. Поэтому будем предполагать, что случайные величины, образующие сцену, взаимно независимы, а случайные величины, образующие объект ξ_A, имеют одно и тоже распределение. Пусть $C\subset Z^2$ - квадрат, назовем его границей $Fr(C)$ совокупность точек из C, имеющих соседей в $Z^2\setminus C$. Семейство случайных величин $\xi_C=(\xi_z)_{z\in C}$ назовем зоной интереса объекта ξ_A с вектором $m_A=(m_A^j)_{1\le j\le\nu}$ средних значений, если $A\subset C\setminus Fr(C)$, если все ξ_z, $z\in C\setminus A$, имеют одно и то же распределение с вектором $m_{C\setminus A}=(m_{C\setminus A}^j)_{1\le j\le\nu}$ средних значений и если $d(m_A,m_{C\setminus A})>0$.

Для продолжения изложения требуется воспользоваться случайным расстоянием. Поэтому приведем его определение. Расстояние d на R^ν назовем Борелевским, если Борелевским является отображение $d:R^\nu\times R^\nu\to R$. Из свойств измеримых функций следует, что расстояние Евклида и другие, часто используемые расстояния, являются борелевскими.

Пусть d - борелевское расстояние на R^ν, а $\xi=(\xi_j)_{1\le j\le\nu}$ и $\eta=(\eta_j)_{1\le j\le\nu}$ - векторные случайные величины на вероятностном пространстве (Ω,A,P). В этом случае отображение $d(\xi,\eta):\Omega\to R$, определяемое равенством

$$d(\xi,\eta)(\omega)=d(\xi(\omega),\eta(\omega)),$$

будет случайной величиной на (Ω,A,P). При каждом $\omega\in\Omega$ будут выполняться аксиомы расстояния. Это позволяет назвать d случайным расстоянием на множестве ν-мерных случайных величин, а $d(\xi,\eta)$ - случайным расстоянием между ξ и η.

Пусть $\xi=(\xi_j)_{1\le j\le\nu}$ - векторная случайная величина и $m_\xi=(m_\xi^j)_{1\le j\le\nu}$ - вектор средних значений ее случайных компонент. Назовем дисперсией ξ неотрицательное число $D\xi$ вида

$D\xi = Ed^2(\xi, m_\xi)$. С помощью этого определения удается установить целый ряд полезных свойств, включая знаменитое неравенство Чебышева.

Из определения зоны интереса следует, что

$$d(E\xi_a, m_{C\setminus A}) = d(m_A, m_{C\setminus A}) > 0,\ a \in A,\ \text{и}$$

$$d(E\xi_z, m_{C\setminus A}) = d(m_{C\setminus A}, m_{C\setminus A}) = 0,\ z \in C \setminus A.$$

Если известно изображение $x_C = (x_C^j)_{1\le j\le \nu}$ зоны интереса, то для неизвестных средних значений легко вычислить оценки. Пусть $B(z,r)$ - круг с центром z и радиусом r, тогда векторная случайная величина вида

$$\bar{x}_z = \frac{1}{|B(z,r)|}\sum_{t\in B(z,r)} x_t$$

будет оценкой неизвестного среднего значения $E\xi_z$. Если упорядочить расстояния $d(\bar{x}_z, m_{C\setminus A})$, $z \in C \setminus Fr(C)$, по возрастанию, то при больших значениях r, очевидно, что расстояния, соответствующие пикселям объекта, окажутся в самом конце последовательности. Поэтому для сегментации достаточно располагать только площадью заданного объекта и вектором средних значений $m_{C\setminus A}$. Для последнего, используя изображение $x_{Fr(C)}$ границы зоны интереса, легко получить оценку $\bar{x}_{Fr(C)}$ при помощи равенства

$$\bar{x}_{Fr(C)} = \frac{1}{|Fr(C)|}\sum_{z\in Fr(C)} x_z .$$

Описанный способ сегментации зоны интереса по ее векторному изображению естественно рассматривать как обобщение метода квантилей. Так же естественно обобщается и метод мод. В самом деле, есть все основания полагать, что скалярные случайные величины $d(\bar{x}_z, m_{C\setminus A})$, $z \in C \setminus Fr(C)$, будут группироваться около нуля, а скалярные случайные величины $d(\bar{x}_a, m_A)$, $a \in A$, вокруг $d(m_A, m_{C\setminus A}) > 0$. Поэтому, начиная с некоторого значения r, можно рассчитывать на появление точки локального минимума.

Для иллюстрации возможностей сегментации методом мод воспользуемся векторным изображением фрагмента сцены, проекция которой является квадратом со стороной равной 256 пикселям. Сцена состоит из пяти элементов. Четыре элемента являются прямоугольниками и называются объектами. Пиксели j-ого объекта описываются 3-мерными случайными величинами. Все скалярные координаты взаимно независимы и имеют нормальные распределения с одними и теми же параметрами (m_j, σ_j^2). Все остальные пиксели образуют пятый элемент, который называется фоном. Пиксели фона так же описываются 3-мерными случайными величинами, координаты которых взаимно независимы и имеют нормальные распределения с параметрами (m_0, σ_0^2). Мерой качества изображения зоны интереса служит отношение сигнал/шум

$$k_j = \frac{|m_0 - m_j|}{\sigma_0 + \sigma_j}, 1 \le j \le 4.$$

Относительная частота ошибки классификации (сегментации) пикселей каждой зоны интереса представлена в таблице 1. Во второй строке приведены теоретические вероятности ошибки классификации.

Таблица 1

h	**0.5**	**1.0**	**1.5**	**2.0**
$e(h)$	0.308	0.159	0.067	0.023
ν **=1**	-	0,018	0,013	0,003
ν **=2**	0,071	0,021	0,008	0,002
ν **=3**	0,035	0,007	0,005	0,001

Как и ожидалось, с ростом k и количества ν используемых изображений вероятность ошибки уменьшается. Вместе с тем эксперименты показали, что при $k \le 1$ вычисление точки локального минимума y_0 наталкивается на серьезные трудности, вызванные сходством объекта и фона. Пропуск в таблице означает, что точку локального минимума определить не удалось.

Отметим, что для реальных сцен предположения, сделанные в настоящей работе, в общем случае не выполняются. Однако, компьютерные эксперименты, проведенные с изображениями реальных сцен, подтвердили выявленные тенденции зависимости ошибки сегментации от k и ν.

Динамически настраиваемые WEB-ориентированные информационные системы

Хлызов А. В.,
мех-мат факультет Пермского государственного университета,
г. Пермь, 614051, ул. Старцева, д.21, кв 79.
E-mail: andrew_khlyzov@mail.ru
Чичагова М. В.,
кафедра математического обеспечения вычислительных систем
Пермского государственного университета
E-mail: mvch83@mail.ru

Введение.

Функционирование современных организаций связано с обработкой больших объемов информации. Поэтому возникает необходимость в автоматизации процесса обработки данных. Этой цели и служат информационные системы (ИС). В связи с постоянной изменчивостью в структуре и составе информационных ресурсов возникает потребность в создании средств расширения функциональности ИС, реструктуризации предметной области (ПО). Информационные системы, обладающие данными возможностями, называются динамически настраиваемыми.

Представленная работа основывается на CASE-технологии METAS [1,2], которая поддерживает весь жизненный цикл ИС и обеспечивает разработчиков средствами реструктуризации данных, настройки и генерации интерфейса, компонентами репортинга, управления бизнес-процессами и защиты.

Системы, созданные по технологии METAS, основаны на интерпретации многоуровневых метаданных. Метаданные описывают *модель данных*, *пользовательский интерфейс*, *запросы* к БД и *шаблоны документов*, генерируемых в системе. Функциональность расширяется также за счет использования *бизнес-операций* и средств *управления бизнес-процессами*. Компоненты, реализующие бизнес-операции, могут быть реализованы сторонними разработчиками и подключены в ходе эксплуатации системы.

Изменения в составе и структуре информационных ресурсов тесно связаны с изменениями в структуре организации, использующей ИС (расширение штата сотрудников, реорганизация и т.п.). Windows-приложения уступают Web-аналогам в отношении объема работ, направленных на увеличение числа пользователей системы и развертывание приложений. Кроме того, наличие Web-интерфейса увеличивает мобильность пользователей, что дает им возможность получать доступ к ресурсам ИС в любой момент времени вне зависимости от их местоположения. Таким образом, мы приходим к необходимости создания Web-ориентированных ИС.

Целью данной работы является разработка на основе технологии METAS CASE-средства для создания динамически настраиваемых Web-ориентированных информационных систем.

Разрабатываемое CASE-средство должно удовлетворять следующим требованиям:

- От пользователя, создающего информационные системы при помощи CASE-средства, не должны требоваться навыки программиста. Интерфейс CASE-средства должен быть максимально прозрачным в использовании.
- CASE-средство должно предоставлять возможность разработки Web-приложений.
- CASE-средство должно предоставлять возможность создания информационных систем с возможностью динамической реструктуризации, подразумевающей изменение структуры предметной области без перекомпиляции Web-приложения.
- Информационные системы, создаваемые при помощи CASE-средства, должны обеспечивать возможность создания, изменения, удаления информации обо всех объектах ПО.
- CASE-средство должно предоставлять возможность настройки пользовательского интерфейса, обеспечивающего навигацию по объектам ПО и редактирование информации об объектах.
- CASE-средство должно иметь гибкую систему разграничения доступа, обеспечивающую конфиденциальность хранимой информации.

Архитектура информационных систем.

Системы, создаваемые при помощи технологии METAS, имеют трехуровневую архитектуру, включающую физическую, логическую и презентационную модели данных. На физическом уровне описываются таблицы и связи между ними, в которых хранится информация об объектах. Логическая модель основывается на физической и обеспечивает объектно-ориентированную надстройку, позволяющую оперировать данными в терминах предметной области. Презентационная модель описывает интерфейс пользователя.

ИС, создаваемые при помощи инструментария METAS, представляют собой Windows-приложения. Данная работа направлена на создание Web-аналога ИС METAS.

В данной работе используются метаданные логического и презентационного уровня, а также создается многоуровневая модель безопасности.

Разграничение доступа.

Описание ИС с помощью метаданных дает возможность описания объектов в терминах предметной области, а также учитывать взаимосвязи между объектами. Это позволяет реализовать в рамках CASE-средства модель распространения прав доступа [3], которая существенно облегчит администрирование системы.

Основными понятиями данной модели являются понятия объект доступа, субъект и права доступа. Понятие объекта доступа совпадает с понятием объекта ИС. Субъектами являются пользователи и группы. Использование групп в механизме управления правами доступа позволят объединять пользователей с похожими, общими правами в группы, что существенно снижает трудоемкость администрирования. Модель распространения прав доступа позволяет учесть взаимосвязи между объектами и между субъектами системы и организовать наследование прав доступа. Кроме того данная модель позволяет учитывать при обращении к объекту контекст доступа.

Генерация WEB-интерфейса.

Пользовательский интерфейс состоит из дерева объектов, которое играет роль карты сайта, и главного окна. Главное окно предоставляет возможность для редактирования информации об объектах.

Настройка пользовательского интерфейса подразумевает генерацию Web-форм для редактирования данных и заполнение дерева объектов.

Дерево объектов предназначено для отображения связей между объектами предметной области. Его структура задается метаданными, а заполнение выполняется динамически, т.е. дочерние вершины добавляются в момент обращения к родительской. Это обосновано потенциально большим количеством объектов в ИС. Вершины дерева представляют соответствующие им объекты. При этом каждый объект может быть представлен несколькими вершинами. В связи с этим между двумя объектами в дереве может существовать несколько путей. Навигация по объектам предметной области подразумевает последовательный просмотр вершин дерева объектов. Пользователь получает доступ к вершинам дерева

в зависимости от назначенных ему прав. Причем права зависят не только от объекта, который представлен данной вершиной, но и от контекста доступа (пути от корня до текущей вершины в дереве).

Каждый объект представляется набором своих атрибутов. Поэтому редактирование информации об объектах предусматривает изменение его атрибутов. Каждый атрибут на форме редактирования представлен соответствующим элементом управления. Причем имеет место иерархия элементов управления, что позволяет унифицировано обрабатывать события изменения атрибутов объекта, проверять права на совершение операций над атрибутами и расширять функциональность системы за счет добавления новых типов элементов управления. Для элементов управления задается их расположение на форме, что позволяет динамически настраивать форму редактирования.

В связи с наличием связей между объектами появляется возможность создания более гибкого интерфейса, а именно возможность выносить на форму редактирования атрибуты связанных объектов. Кроме того возможна навигация к родительским и дочерним объектам, которая также контролируется правами текущего пользователя с учетом контекста.

Заключение.

В результате проделанной работы был реализован Web-интерфейс для системы METAS, обеспечивающий полный жизненный цикл объектов. Разработанный интерфейс имеет гибкие возможности динамической настройки и расширения, не требующие перекомпиляции Web-приложения.

В рамках Web-системы была разработана и реализована модель распространения прав доступа, предоставляющая возможность создания иерархии групп и контролирования доступа к объектам с учетом контекста.

На основе реализованного приложения стала возможной разработка Web-ориентированных ИС. В дальнейшем планируется работа по увеличению функциональности за счет поддержки бизнес-операций и интеграции с информационными системами, разработанными в соответствии с технологией METAS.

Список литературы

1. Лядова Л. Н., Рыжков С. А. CASE-технология METAS// Межвузовский сб. науч. трудов / Математика программных систем. Пермь: Перм. ун-т. 2003. С. 4-19.

2. Куделько Е. Ю. Генерация и настройка экранных форм на основе метаданных // Математика программных систем: Межвуз. сб. науч. тр. / Перм. ун-т. – Пермь, 2003. С.51-59.

3. Чичагова М. В. Модель распространения прав доступа // Инфокоммуникационные технологии в науке, производстве и образовании: Вторая Международная научно-техническая конференция / Северо-кавказский государственный технический университет – г. Ставрополь, 2006 г. С. 115-120.

4. Хлызов А. В. Разработка средств создания порталов с использованием многоуровневых метаданных // Технологии Microsoft в теории и практике программирования. Материалы конференции / Изд-во Нижегородского госуниверситета – Нижний Новгород, 2006. С.310-313.

Моделирование флаттера вязкоупругих пластин и цилиндрических панелей

Худаяров Б. А., Бандурин Н. Г.

Рассмотрим пологой вязкоупругой оболочки, обтекаемой с внешней стороны сверхзвуковым потоком газа; невозмущенную скорость потока V будем считать направленной вдоль образующей. Будем пользоваться обычными гипотезами теории упругих оболочек [1, 2], считая справедливой гипотезу Кирхгоффа-Лява и полагая прогибы малыми по сравнению толщиной оболочки.

Уравнения Маргерра в декартовой системе координат относительно перемещений u, v и w с учетом вязкоупругих свойств материала конструкций, можно записать в следующем виде:

$$
\begin{aligned}
&(1-R^*)\left\{\frac{\partial^2 u}{\partial x^2}+\frac{1-\mu}{2}\frac{\partial^2 u}{\partial y^2}+\frac{1+\mu}{2}\frac{\partial^2 v}{\partial x\partial y}+L_1(w)\right\}-\rho\frac{1-\mu^2}{E}\frac{\partial^2 u}{\partial t^2}=0\,,\\
&(1-R^*)\left\{\frac{\partial^2 v}{\partial y^2}+\frac{1-\mu}{2}\frac{\partial^2 v}{\partial x^2}+\frac{1+\mu}{2}\frac{\partial^2 u}{\partial x\partial y}+L_2(w)\right\}-\rho\frac{1-\mu^2}{E}\frac{\partial^2 v}{\partial t^2}=0\,,\\
&D(1-R^*)\nabla^4 w+L_3^*(u,v,w)+\rho h\frac{\partial^2 w}{\partial t^2}=q\,,
\end{aligned}
\qquad (1)
$$

где $q = -B\frac{\partial w}{\partial t} - BV\frac{\partial w}{\partial x} - B_1 V^2\left(\frac{\partial w}{\partial x}\right)^2 - \ldots$ аэродинамическое давление, определяемое по теории Ильюшина [3]; $B = \frac{\chi p_\infty}{V_\infty}$, $B_1 = \frac{\chi(\chi+1)p_\infty}{4V_\infty^2}$, χ – показатель политропы газа; p_∞, V_∞ - соответственно давление и скорость звука на бесконечности; $L_1(w)$, $L_2(w)$, $L_3^*(u,v,w)$ - операторы.

Для полного определения задачи к ИДУ (1) необходимо добавить граничные и начальные условия. Граничные условия будут иметь вид:

при $x=0$, $x=a$ $\quad w=0;\ v=0;\ N_x=0;\ M_x=0;$

при y=0, y=b $\quad w=0;\ u=0;\ N_y=0;\ M_y=0.$

При интегрировании основных уравнений должны быть удовлетворены также следующие начальные условия, относящиеся к перемещениям и скоростям точек срединной поверхности оболочки:

$$u(x,y,0)=\varphi_1(x,y),\quad \dot{u}(x,y,0)=\psi_1(x,y), \tag{2}$$

$$v(x,y,0)=\varphi_2(x,y),\quad \dot{v}(x,y,0)=\psi_2(x,y), \tag{3}$$

$$w(x,y,0)=\varphi_3(x,y),\quad \dot{w}(x,y,0)=\psi_3(x,y). \tag{4}$$

Представим функцию $u(x,y,t)$, $v(x,y,t)$, $w(x,y,t)$ в виде разложения по функциям $\varphi_{nm}(x,y)$, $\psi_{nm}(x,y)$, $\theta_{nm}(x,y)$ удовлетворяющим соответствующим граничным условиям

$$\begin{aligned} u(x,y,t) &= \sum_{n=1}^{N}\sum_{m=1}^{M} u_{nm}(t)\varphi_{nm}(x,y) \\ v(x,y,t) &= \sum_{n=1}^{N}\sum_{m=1}^{M} v_{nm}(t)\psi_{nm}(x,y) \\ w(x,y,t) &= \sum_{n=1}^{N}\sum_{m=1}^{L} w_{nm}(t)\theta_{nm}(x,y) \end{aligned} \tag{5}$$

где $u_{nm}=u_{nm}(t)$ $v_{nm}=v_{nm}(t)$ $w_{nm}=w_{nm}(t)$ – искомые функции времени; $\varphi_{nm}(x,y)$, $\psi_{nm}(x,y)$, $\theta_{nm}(x,y)$ - известные ортогональные функции, зависящие от граничных условий.

Подставляя (5) в (1) и выполняя процедуру Бубнова-Галеркина для определения $u_{nm}(t)$, $v_{nm}(t)$, $w_{nm}(t)$ получим систему нелинейных ИДУ в виде:

$$\sum_{n=1}^{N}\sum_{m=1}^{M} N_{1k\ln m}\ddot{u}_{nm} - \Omega(1-R^*)\left\{\sum_{n=1}^{N}\sum_{m=1}^{M}\left(A_{1k\ln m}u_{nm} + \frac{1+\mu}{2}B_{1k\ln m}v_{nm} - (k_x+\mu k_y)C_{1k\ln m}w_{kl}\right) + \sum_{n,i=1}^{N}\sum_{m,r=1}^{M} D_{1k\ln mir}w_{nm}w_{ir}\right\} = 0, \tag{6}$$

$$\sum_{n=1}^{N}\sum_{m=1}^{M} N_{2k\ln m}\ddot{v}_{nm} - \Omega(1-R^*)\left\{\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\frac{1+\mu}{2}A_{2k\ln m}u_{nm} + B_{2k\ln m}v_{nm} - (k_y+\mu k_x)C_{2k\ln m}w_{kl}\right) + \sum_{n,i=1}^{N}\sum_{m,r=1}^{M} D_{2k\ln mir}w_{nm}w_{ir}\right\} = 0, \tag{7}$$

$$\begin{aligned} &\sum_{n=1}^{N}\sum_{m=1}^{M} N_{3k\ln m}\left(\ddot{w}_{nm} + M_\lambda\dot{w}_{nm}\right) + (1-R^*)\Omega\left\{\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\frac{1+\mu}{2}A_{3k\ln t}u_{nm} + B_{3k\ln m}v_{nm} + C_{3k\ln m}w_{nm}\right) - \sum_{n,i=1}^{N}\sum_{m,r=1}^{M} P_{k\ln mir}w_{nm}w_{ir}\right\} - \sum_{n,i=1}^{N}\sum_{m,r=1}^{M} w_{nm}\times \\ &\times(1-R^*)\left\{A_{4k\ln mir}u_{ir} + B_{4k\ln mir}v_{ir} + + C_{4k\ln mir}w_{ir}\right\}\Omega + \\ &+ \chi M_p\left(\lambda_1 M^*\sum_{n=1}^{N}\gamma_{k\ln m}w_{nm} + \frac{\chi+1}{4}M^{*2}\sum_{n,i=1}^{N}\sum_{m,r=1}^{M}\Gamma_{kenmi}w_{nm}w_{ir}\right) = 0, \end{aligned} \tag{8}$$

$$u_{nm}(o) = u_{onm}, \quad \dot{u}(o) = \dot{u}_{onm}, \quad v_{nm}(o) = v_{onm}, \quad \dot{v}(o) = \dot{v}_{onm},$$

$$w_{nm}(o) = w_{onm}, \quad \dot{w}(o) = \dot{w}_{onm}, \quad n = \overline{1, N}; \quad m = \overline{1, M},$$

где $u_{nm} = u_{nm}(t)$, $v_{nm} = v_{nm}(t)$, $w_{nm} = w_{nm}(t)$ – искомые функции времени.

Для вычисления критической скорости флаттера предлагается численный алгоритм, построенный на основе метода Ф.Бадалова [4].

Одно из основных преимуществ метода Ф.Бадалова является его универсальность и простота реализации алгоритмов решении на персональных компьютерах. Здесь этот метод применим и для систем (6)-(8). Запишем эту систему в интегральной форме. Полагая затем $t=t_i$, $t_i=i\Delta t$, $i=1,2,\ldots$ $(\Delta t=const)$ и заменяя интегралы некоторыми квадратурными формулами для вычисления $u_{nm} = u_{nm}(t)$, $v_{nm} = v_{nm}(t)$, $w_{nm} = w_{nm}(t)$, получим следующее рекуррентное соотношение:

$$\sum_{n=1}^{N}\sum_{m=1}^{M} N_{1k\ln m} u_{pnm} = \sum_{n=1}^{N}\sum_{m=1}^{M} N_{1k\ln m}(u_{0nm} + \dot{u}_{onm} t_p) -$$
$$- \Omega\sum_{j=0}^{p-1} A_j(t_p - t_j)\left\{\sum_{n=1}^{N}\sum_{m=1}^{M}\left(A_{1k\ln m}\left[u_{jnm} - \frac{A}{\alpha}\sum_{s=0}^{j} B_s \exp(-\beta t_s) u_{j-snm} \right] + \right.\right.$$
$$+ \frac{1+\mu}{2} B_{1k\ln m}\left[v_{jnm} - \frac{A}{\alpha}\sum_{s=0}^{j} B_s \exp(-\beta t_s) v_{j-skl} \right] -$$
$$\left.\left. - (k_x + \mu k_y) C_{1k\ln m}\left[w_{jnm} - \frac{A}{\alpha}\sum_{s=0}^{j} B_s \exp(-\beta t_s) w_{j-snm} \right]\right) + \right. \tag{9}$$
$$\left. + \sum_{n,i=1}^{N}\sum_{m,r=1}^{M} D_{1k\ln mir}\left(w_{jnm} w_{jir} - \frac{A}{\alpha}\sum_{s=0}^{j} B_s \exp(-\beta t_s) w_{j-snm} w_{j-sir} \right)\right\},$$

$$\sum_{n=1}^{N}\sum_{m=1}^{M} N_{2k\ln m} v_{pnm} = \sum_{n=1}^{N}\sum_{m=1}^{M} N_{2k\ln m}(v_{0nm} + \dot{v}_{onm} t_p) -$$
$$- \Omega\sum_{j=0}^{p-1} A_j(t_p - t_j)\left\{\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\frac{1+\mu}{2} A_{2k\ln m}\left[u_{jnm} - \frac{A}{\alpha}\sum_{s=0}^{j} B_s \exp(-\beta t_s) u_{j-snm} \right] + \right.\right.$$
$$+ B_{2k\ln m}\left[v_{jnm} - \frac{A}{\alpha}\sum_{s=0}^{j} B_s \exp(-\beta t_s) v_{j-skl} \right] -$$
$$\left.\left. - (k_y + \mu k_x) C_{2k\ln m}\left[w_{jnm} - \frac{A}{\alpha}\sum_{s=0}^{j} B_s \exp(-\beta t_s) w_{j-snm} \right]\right) + \right.$$
$$\left. + \sum_{n,i=1}^{N}\sum_{m,r=1}^{M} D_{2k\ln mir}\left(w_{jnm} w_{jir} - \frac{A}{\alpha}\sum_{s=0}^{j} B_s \exp(-\beta t_s) w_{j-snm} w_{j-sir} \right)\right\}, \tag{10}$$

$$\sum_{n=1}^{N}\sum_{m=1}^{M} N_{3k\ln m} w_{pnm} = \frac{1}{1 + A_p M_\lambda}\left\{\sum_{n=1}^{N}\sum_{m=1}^{M} N_{3k\ln m}\left\langle w_{0nm} + \left(\dot{w}_{0nm} + M_\lambda w_{0nm} \right) t_p \right\rangle - \right.$$
$$\left. - \sum_{j=0}^{p-1} A_j\left(M_\lambda \sum_{n=1}^{N}\sum_{m=1}^{M} N_{3k\ln m} w_{jnm} - (t_p - t_j)\right\rangle \aleph M_p\left(\lambda_1 M^* \sum_{n=1}^{N}\sum_{m=1}^{M} \gamma_{k\ln m} w_{jnm} + \right.\right.$$

$$+\frac{\aleph+1}{4}M^{*2}\sum_{n,i=1}^{N}\sum_{m,r=1}^{M}\Gamma_{k\ln mir}w_{jnm}w_{jir}\Bigg)+\frac{1+\mu}{2}\Omega\sum_{n=1}^{N}\sum_{m=1}^{M}A_{3k\ln m}\Bigg(\ u_{jnm}-$$

$$-\frac{A}{\alpha}\sum_{s=0}^{j}B_s\exp(-\beta t_s)u_{j-snm}\Bigg)+\Omega\sum_{n=1}^{N}\sum_{m=1}^{M}B_{3k\ln m}\Bigg(v_{jnm}-\frac{A}{\alpha}\sum_{s=0}^{j}B_s\exp(-\beta t_s)v_{j-snm}\Bigg)+$$

$$+\Omega\sum_{n=1}^{N}\sum_{m=1}^{M}C_{3k\ln m}\Bigg(w_{jnm}-\frac{A}{\alpha}\sum_{s=0}^{j}B_s\exp(-\beta t_s)w_{j-snm}\Bigg)-$$

$$-\Omega\sum_{n,i=1}^{N}\sum_{m,r=1}^{M}p_{k\ln mir}\Bigg(w_{jnm}w_{jir}-\frac{A}{\alpha}\sum_{s=0}^{j}B_s\exp(-\beta t_s)w_{j-snm}w_{j-sir}\ \Bigg)-$$

$$-\Omega\sum_{n,i=1}^{N}\sum_{m,r=1}^{M}w_{jnm}\Bigg\langle A_{4k\ln mir}\Bigg(u_{jir}-\frac{A}{\alpha}\sum_{s=0}^{j}B_s\exp(-\beta t_s)u_{j-sir}\Bigg)+$$

$$+B_{4k\ln mir}\Bigg(v_{jir}-\frac{A}{\alpha}\sum_{s=0}^{j}B_s\exp(-\beta t_s)v_{j-sir}\Bigg)+$$

$$+C_{4k\ln mir}\Bigg(w_{jir}-\frac{A}{\alpha}\sum_{s=0}^{j}B_s\exp(-\beta t_s)w_{j-sir}\Bigg)\Bigg\rangle\Bigg]\Bigg\} \qquad (11)$$

$$p=1,2,...; \qquad k=\overline{1,N}; \quad l=\overline{1,M};$$

где A_j, B_s - числовые коэффициенты применительно к квадратурным формулам трапеции:

$$A_0=\frac{\Delta t}{2}; \quad A_j=\Delta t, \quad j=\overline{1,i-1}; \quad A_i=\frac{\Delta t}{2}; \quad B_0=\frac{\Delta t^{\alpha}}{2}; \quad A_j=\frac{\Delta t^{\alpha}(j^{\alpha}-(j-1)^{\alpha})}{2}, \quad s=j;$$

$$B_s=\frac{\Delta t^{\alpha}((s+1)^{\alpha}-(s-1)^{\alpha})}{2}.$$

Результаты вычислений представлены в таблице.

Таблица

A	α	β	λ	λ_1	β_1	V_{kp}
0 0,001 0,01 0,1	0,25	0,05	6	100	0,01	1528 1497 1386 871
0,1	0,1 0,3 0,5	0,05	6	100	0,01	575 918 1281
0,1	0,25	0,1 0,3	6	100	0,01	869 867
0,1	0,25	0,05	3,5 4 4,5 5,5	100	0,01	964 1251 1143 1658
0,1	0,25	0,05	6	50 75 100	0,01	1184 988 871
0,1	0,25	0,05	6	100	0,005 0,008 0,015	723 807 1324

В качестве критерия, определяющего критическую скорость $V_{кр}$, принимаем условие, предложенное в работе [5-7]. В таблице приводятся результаты расчетов для вязкоупругой цилиндрической панели, обтекаемой сверхзвуковым потоком газа.

Как видно из анализа результатов, приведенных в таблице, критические значение скорости флаттера V_{kp} оказывается в упругом (A=0) и в вязкоупругом случае (A=0,1) соответственно равным 1528 и 871. Таким образом, вязкоупругие свойства материала конструкций приводит к уменьшению критической скорости флаттера.

Обратим внимание влияния параметра сингулярности α на критической скорости флаттера. Для α = 0,1; 0,3 критическая скорость флаттера, соответственно равна 575 и 918 и отличается друг от друга на 59,6 %. Следовательно, учет этого эффекта при проектировании авиа и ракетостроение имеет большое значение для обеспечения безопасности полета.

Изучено влияние параметра затухания β на критическую скорость V_{kp} флаттера. Расчеты были проведены при β=0,1 и β=0,3. Полученные результаты показывает, что влияния параметра затухания β на критической скорости флаттера панели незначительно.

Таблица показывает, что увеличение параметра λ_1, приводит к уменьшению критической скорости флаттера, а с увеличением параметров λ и β_1, критическое число V_{kp} увеличивается, т.е. уменьшается опасность флаттерных движений.

В заключение отметим, что влияние параметра затухания β ядра наследственности на скорость флаттера пластинки по сравнению с влиянием параметра вязкости A и сингулярности α незначительно, что еще раз подтверждает общеизвестные выводы о том, что экспоненциальное ядро релаксации неспособно полностью описать наследственные свойства материала конструкций.

Список литературы

1. Вольмир А. С. Нелинейная динамика пластинок и оболочек. - М.: Наука, 1972. - 432 с.

2. Вольмир А. С. Устойчивость упругих систем. - М.: Физматгиз, 1963. - 880 с.

3. Ильюшин А. А. Закон плоских сечений в аэродинамике больших сверхзвуковых скоростей // Прикладная математика и механика. - 1956. Т.XX. - Вып.6. - С.733-755.

4. Бадалов Ф. Б. Методы решения интегральных и интегро-дифференциальных уравнений наследственной теории вязкоупругости. Ташкент: Мехнат. 1987. 269 с.

5. Khudayarov B. A. Numerical Analysis of the Nonlinear Flutter of Viscoelastic Plates // INTERNATIONAL APPLIED MECHANICS. New-York, USA. – 2005. –Vol. 41. – No 5. – P. 538-542.

6. Худаяров Б. А., Бандурин Н. Г. Нелинейный флаттер вязкоупругих отротропных цилиндрических панелей // Ж. Математическое моделирование. РАН. – 2005. Том 17. – №10. – С. 79–86.

7. Худаяров Б. А. Численное решение задачи о флаттере вязкоупругих трехслойных пластин // Известия НАН Армении, Механика. - 2004. - Т.57. - №1. – С. 59-62.

Интеллектуальные системы для прогнозно-металлогенического анализа перспективных площадей

Чижова И. А.

Ведущий научный сотрудник, кандидат физико-математических наук

Институт геологии рудных месторождений, петрографии, минералогии и геохимии Российской Академии Наук (ИГЕМ РАН),

119017, Москва, Старомонетный переулок, 35,

e-mail: tchijova@igem.ru

При прогнозно-металлогеническом анализе перспективных площадей наиболее важными задачами являются определение рудноформационного типа месторождений и оценка их масштабности. Для решения этих задач в автоматизированном виде требуется следующая последовательность действий:

постановка геологической задачи;

формализация задачи;
выбор метода решения;
проверка условия его применимости;
подготовка геологического материала к компьютерному анализу;
разработка целевых баз данных;
применение метода;
интерпретация результата (модели);
построение компьютерных интеллектуальных систем;
апробация технологии.

Распознающие системы строятся по принципу общности свойств (тип 1) или по принципу перечисления (тип 2) [1]. В первом случае опираясь на принцип общности свойств, который основан на предположении, что объекты, образующие один класс, обладают свойством подобия, отраженного в их характеристиках, необходимо выделить системы информативных признаков, описывающих группу объектов в целом (отождествляющие признаки) и каждый тип в отдельности (делящие признаки), и на их основе построить решающее правило отнесения вновь предъявляемого объекта к одному из выделенных типов объектов. Во втором случае используется некоторая мера сходства для оценки экспертируемого объекта для определения степени его близости к одному из эталонов. При этом возможен выбор аналога по любому набору признаков, что позволит пользователю проверить различные гипотезы и сравнить работоспособность различных критериев. Особенности построения баз знаний при использовании этих подходов представлены на рис. 1.

Строящаяся система состоит из четырех основных взаимосвязанных блоков (рис. 2):

1. информационно-аналитического;

2. интеллектуально-графического;

3. блока экспертного выбора;

4. формирования результатов.

Блок 1 на основе ГИС-технологии (в графической форме) обеспечивает визуализацию исходной информации по целевым параметрам из базы данных.

Блок 2 на основе технологии искусственного интеллекта обеспечивает работу системы принятия решений для получения заключений о формационном типе, масштабности оруденения и выборе ближайшего аналога для экспертируемого объекта по имеющейся базе знаний.

Блок 3 на основе выбранных пользователем системы информативных признаков или меры близости проводит поиск ближайшего аналога.

Блок 4 является блоком формирования результатов и оценки их достоверности и обеспечивает на основе ГИС-технологии визуализацию полученных заключений.

Рис. 1. Особенности структуры базы знаний экспертной системы при использовании различных принципов ее построения

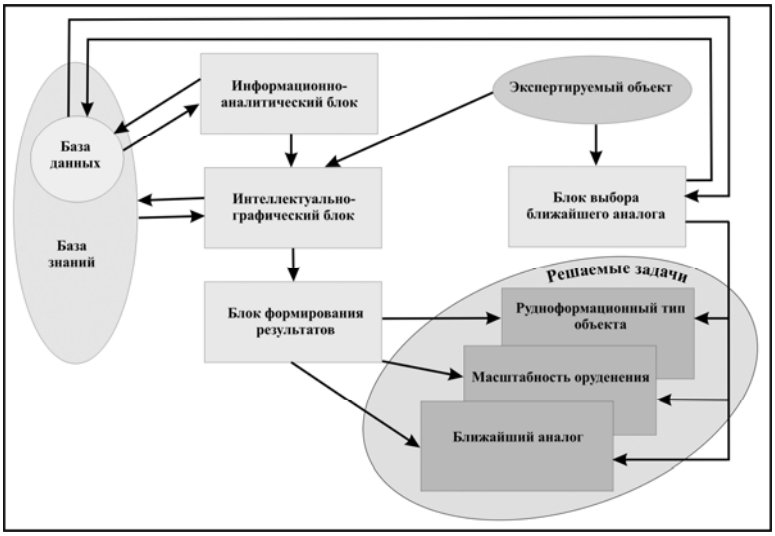


Рис. 2 . Схема процесса функционирования системы АНАЛОГ

Экспертная система предназначена для помощи эксперту в оценке потенциала перспективных площадей. Опыт разработки экспертных систем показал, что наилучшим образом поставленным целям отвечают модели решающих правил.

Они могут быть построены как на основе знаний экспертов, так и в результате математической обработки имеющейся базы данных по эталонным объектам (эвристические и расчетно-логические правила).

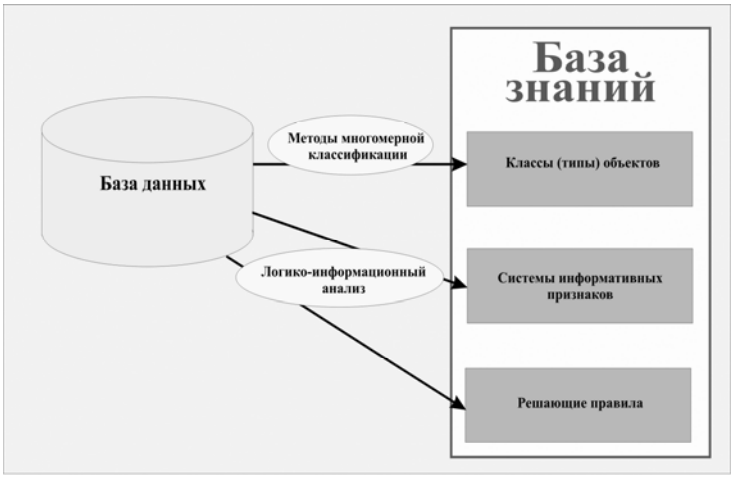


Рис. 3 . Схема формирования базы знаний с использованием математических методов для системы АСТРА

Для построения распознающих систем первого типа предложена технология построения интеллектуальных систем АСТРА, база знаний которых формируется на основе логико-информационного анализа имеющейся базы данных изучаемых объектов (рис. 3).

Преимуществом логико-информационных методов анализа данных являются отказ от строгих предположений относительно свойств объекта (предположений метризуемости, подчиненности вероятностным законам и др.) и возможность вести распознавание при наличии довольно малой обучающей выборки. Наиболее общее свойство алгоритмов комбинаторно-логического типа состоит в том, что наряду с учетом информации по отдельным признакам производится целенаправленный поиск и использование сочетаний признаков, содержащих полную информацию о разделении классов эталонных объектов [2]. При малых значениях функций принадлежности экспертируемого объекта к выделенным классам, предполагается, что объект не принадлежит этим классам.

С использованием данной технологии создано несколько специализированных экспертных систем. Среди них “Автоматизированный прогнозно-поисковый комплекс на объекты золото-кварцевой и золото-редкометалльно-кварцевой формации Северо-Востока" (Чижова И. А., Курбанаев Г. М. и др.), Автоматизированный прогнозно-поисковый комплекс на россыпные объекты золота Магаданской области" (Чижова И. А., Матвеева Е. В. и др.), “Автоматизированная экспертная система прогноза золоторудных месторождений на примере Охотско-Чукотского вулканогенного пояса” (Чижова И. А., Стружков С. Ф., Константинов М. М). Схема принятия решения для экспертной системы АСТРА представлена на рис. 4.

Для построения распознающих систем второго типа предложена технология построения интеллектуальных систем АНАЛОГ (рис. 2), база данных которых содержит набор правил для попарного сравнения объектов исходя из природы анализируемой информации (возраст, минеральный состав и т.д.) с учетом возможных типов представления исходных данных (количественный, качественный, логический).

Технология апробирована при создании информационно-аналитической системы АНАЛОГ для поиска ближайшего аналога золоторудных месторождений (Константинов М. М., Чижова И. А., Стружков С. Ф., Покровский Д.А.) и решения прогнозно-металлогенических задач по методу аналогий (рис. 5).

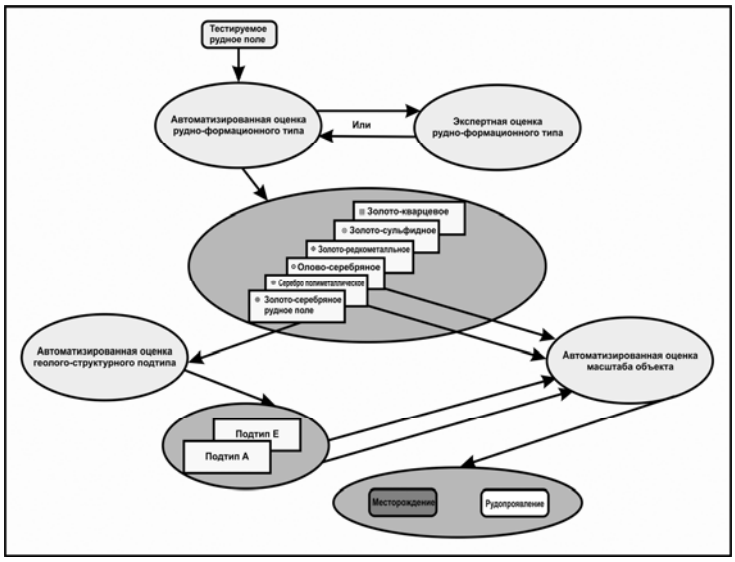


Рис. 4. Схема процесса принятия решений в экспертной системе АСТРА для золоторудных месторождений Охотско-Чукотского вулканогенного пояса

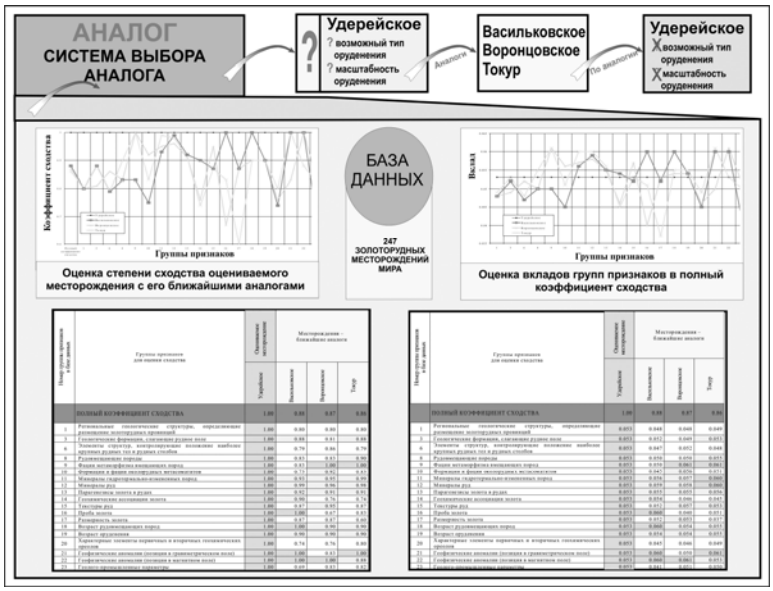


Рис. 5. Схема процесса принятия решений в экспертной системе АНАЛОГ для оценки золоторудных месторождений Мира

Список литературы

[1] Ту Дж. Гонсалес Р. Принципы распознавания образов. М.: Мир, 1978.

[2] Кудрявцев В. Б., Алешин С. В. Комбинаторно-логический подход к распознаванию образов. Интеллектуальные системы. Том 1, вып. 1-4, 1996.

Применение автоматного программирования при моделировании одного класса мультиагентных систем

Шалыто А. А.,
заведующий кафедрой, shalyto@mail.ifmo.ru
Царев Ф. Н., *tsarev@rain.ifmo.ru*
Паращенко Д. А., *parashchenko@rain.ifmo.ru*
Санкт-Петербургский государственный университет информационных технологий, механики и оптики.
Россия, Санкт-Петербург, Кронверкский пр., 49

Введение

Рефлексные агенты [1] исследовались психологами [2], которые использовали отображение "вход-выход". Такому отображению соответствует модель автомата без памяти – комбинационная схема.

В результате прогресса в психологии возникло понимание того, что необходимо учитывать внутренние состояния агента [3]. В большинстве работ по искусственному интеллекту модель рефлексных агентов с внутренними состояниями (реактивных агентов) рассматривалась как слишком простая для того, чтобы на ее основе можно было добиться значительных успехов, но исследования, описанные в работах [4, 5], "позволили поставить под сомнение это предположение" [1].

Отметим также, что в *NASA* используется аналогичный подход [6] при создании программного обеспечения, входящего в проект *Mars Science Laboratory*.

В 1991 году было предложено автоматное программирование [7] и выполнены исследования по его применению при создании:

- агентов, осуществляющих логическое управление [8];
- реактивных агентов [9, 10];
- объектно-ориентированных реактивных агентов [11].

Для отработки технологии автоматного программирования в рамках *Движения за открытую проектную документацию* [12] выполняется ряд студенческих проектов по созданию мультиагентных систем. Пример построения реактивной системы со сложным поведением описан в работе [13], а примеры построение объектно-ориентированных реактивных агентов – в работах [14–17]. В ходе выполнения этих проектов программирование агентов выполнялось вручную – без использования инструментальных средств.

Цель настоящей работы – описание технологии моделирования одного класса мультиагентных систем на основе автоматного программирования. Эта технология базируется на применении инструментального средства *UniMod* [18, 19], предназначенного для поддержки технологии автоматного программирования, называемой также *SWITCH*-технология [8].

1. Автоматное программирование

При автоматном программировании программы предлагается строить как автоматизированные системы, которые состоят из системы управления (система взаимодействующих автоматов), объектов управления и контуров обратной связи между объектами и системой. При этом автоматы по входным воздействиям (события и входные переменные) выполняют переходы между состояниями, которые явно выделяются, и формируют выходные воздействия, соответствующие методам, реализуемым в объектах управления. Такой взгляд на программирование является естественным для решения задач управления разных уровней, в том числе, для определенных классов мультиагентных систем.

В рамках автоматного программирования предлагается использовать только два типа диаграмм: схема связей автоматов с поставщиками событий и объектами управления, определяющая интерфейс автоматов, а также граф переходов (диаграмма состояний), который описывает поведение автомата.

2. Инструментальное средство UniMod

Открытое инструментальное средство *UniMod* [19] базируется на трех открытых компонентах: унифицированный язык моделирования *UML*, *SWITCH*-технологию и среду разработки *Eclipse* [20]. Это средство является встраиваемым модулем (плагином) для указанной среды разработки.

UniMod позволяет строить и редактировать схемы связей и диаграммы состояний, обеспечивать проверку формальной корректности этих диаграмм, проводить отладку диаграмм в графическом режиме и т. д.

После построения диаграмм и автоматической проверки корректности, по ним строится их *XML*-описание. Далее вручную пишутся следующие фрагменты программы на языке *Java*: для поставщиков событий – их объявления, инициализация и преобразование системных событий в автоматные, а для объектов управления – методы, реализующие входные переменные и выходные воздействия.

Для реализации программ могут использоваться интерпретационный и компиляционный подходы. В первом случае *XML*-описание интерпретируется, вызывая методы скомпилированных *Java*-классов, написанных вручную. Во втором – по *XML*-описанию с помощью шаблонов *Velocity* [21] создается код программы на языке *Java*, реализующий систему автоматов. Этот код компилируется совместно с фрагментами программы, написанными вручную. В обоих случаях в процессе работы программы ведется протокол в терминах автоматов.

3. Описание беспилотных объектов

В настоящей работе технология моделирования одного класса мультиагентных беспилотных объектов с использованием инструментального средства *UniMod* иллюстрируется на примере задачи заочного тура *VI Открытой Всесибирской олимпиады по программированию им. И.В. Поттосина* [22]. Автор задачи – Дмитрий Иртегов.

Беспилотные объекты в этой задаче называются летающими тарелками, каждая из которых представляет собой дискообразное "летающее крыло" заданного радиуса.

На тарелке размещен компьютер, который управляет расходом топлива (тягой двигателя) и положением аэродинамических рулей для обеспечения поворотов. Летающая тарелка может передвигаться со скоростями, не ниже заданного предела. Тарелка, скорость которой упала ниже предела, не может держаться в воздухе. Если при этом ее топливный бак пуст, то она нормально завершает движение, если же в топливном баке осталось топливо, то тарелка завершает движение аварийно.

Тарелка движется в соответствии со вторым законом Ньютона. Ее движение определяется двумя силами: сопротивлением воздуха $\boldsymbol{F}$ и тягой двигателя $\boldsymbol{T}$. Сопротивление воздуха определяется по формуле $F = c_1 + c_2 v^2$, где $\boldsymbol{v}$ – скорость тарелки, а коэффициенты $\boldsymbol{c_1}$ и $\boldsymbol{c_2}$ определяются ее аэродинамическими характеристиками. Тяга двигателя определяется по формуле $T = c_4 q$, где q – расход топлива. Расход топлива контролируется компьютером и может изменяться в заданных пределах. Константа $\boldsymbol{c_4}$ определяется характеристиками двигателя.

Тарелки (агенты) группируются в команды по N, и команды «сражаются». В дальнейшем будем называть каждое такое сражение «игрой». В каждой игре участвуют две команды. В начале игры агенты первой команды располагаются случайным образом в 25 метрах от линии старта в левой половине коридора, а агенты второй – симметрично им в правой половине. Ширина коридора задана, его длина бесконечна. Для каждого агента заданы начальная скорость и направление движения. После команды «Старт» все агенты двигаются по коридору с целью максимально удалиться от линии старта.

Агенты могут сталкиваться друг с другом, так как имеют возможность изменять расход топлива и поворачивать. Столкновения при этом предполагаются абсолютно упругими. Если относительная скорость столкновения превышает заданную величину, то оба агента, считаются выбывшими из игры.

4. Описание поведения объектов

В ходе настоящей работы было сделано предположение, что победной является стратегия, при которой одна половина агентов движется вперед и уклоняется от границ коридора и находящихся рядом агентов, а вторая половина агентов разбивается на пары. При этом агенты в парах движутся, поддерживая взаимное расположение для использования аэродинамического взаимодействия.

При моделировании поставщики событий использовались для реализации «органов чувств» агентов, конечные автоматы – для описания их поведения, а объекты управления – для влияния на окружающий мир. В данной версии программы поведение всех агентов «нашей» команды описывается одной и той же системой взаимодействующих автоматов. Автомат *Состояние агента* имеет три состояния: «Летит», «Нормальное завершение», «Аварийное завершение». В первое состояние этого автомата вложены автомат *Режим полета* и автомат *Радар*. В состояние «Полет в одиночку» автомата *Режим полета* вложены автомат *Уклонение от границ коридора и агентов справа и слева* и автомат *Уклонение от агентов спереди и сзади*. В состояние «Полет первым в паре» автомата *Режим полета* вложен автомат

Полет первым в паре, а в состояние «Полет вторым в паре» автомата *Режим полет* – автомат *Полет вторым в паре*. В каждое из указанных состояний автомата *Режим полета* вложен автомат *Прием и обработка сообщений от других агентов.*

Заключение

Обычно считается, что научная новизна работы в рассматриваемой области должна состоять в изобретении некоторой «хорошей» стратегии. Авторы считают, что новизна настоящей работы состоит в формализации процесса программной реализации выбранной стратегии.

Простая автоматная модель, описываемая в настоящей статье, позволяет обеспечить достаточно сложное поведение системы, а инструментальное средство *UniMod* его эффективно реализовать [23]. Обратим внимание, что при построении диаграмм выполняется проверка корректности их построения, и при участии авторов начались проводиться работы по верификации автоматных программ на основе метода *model checking* [24].

Список литературы

1. Рассел С., Норвиг П. Искусственный интеллект. Современный подход. М.: Вильямс, 2006.

2. Skinner B. F. Science and Human behavior. London: Macmillan, 1953.

3. Putnam H. Mind and machines /Dimensions of Mind. London: Macmillan, 1960, pp.138–164.

4. Rosenschein S. J. Formal Theories of Knowledge in AI and Robostics New Generation Computing. 1985. 3(4), pp.345–357.

5. Brooks R. A. A Robust Layered Control System for a Mobil Robot /IEEE Journal of Robotics and Automation. 1986. 2, pp.14–23.

6. Regan P., Hamilton S., NASA's Mission Reliable. Computer. 2004. January, pp. 59–68.

7. Шалыто А. А. Технология автоматного программирования /Труды первой Всероссийской научной конференции "Методы и средства обработки информации" М.: МГУ. 2003, с.528–535. http://is.ifmo.ru/works/tech_aut_prog

8. Шалыто А. А. Switch-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.

9. Шалыто А. А., Наумов Л. А. Методы объектно-ориентированной реализации

реактивных агентов на основе конечных автоматов // Искусственный интеллект. 2004. № 4, с. 756–762. http://is.ifmo.ru/works/_aut_oop.pdf

10. Шалыто А. А., Туккель Н. И. SWITCH-технология — автоматный подход к созданию программного обеспечения «реактивных» систем // Программирование. 2001. № 5, с.45–62. http://is.ifmo.ru/works/switch/1/

11. Shalyto A., Naumov L., Korneev G. Methods of Object-Oriented Reactive Agents Implementation on the Basis of Finite Automata. Proceedings of International Conference Integration of Knowledge Intensive Multi-Agent Systems: Modeling, Exploration and Engineering. KIMAS-05. Boston: IEEE Boston Section. 2005, pp. 460–465. http://is.ifmo.ru/articles_en/_kimas05-1.pdf

12. Шалыто А. А. Новая инициатива в программировании. Движение за открытую проектную документацию // Мир ПК. 2003. № 9, с. 52–56. http://is.ifmo.ru/works/open_doc/.

13. Туккель Н. И., Шалыто А. А. Система управления дизель-генератором (фрагмент). Программирование с явным выделением состояний. Программная документация. http://is.ifmo.ru/projects/dg/

14. Yartsev B., Korneev G., Shalyto A., Kotov V. Automata-Based Programming of the Reactive Multi-Agent Control Systems. Proceedings of International Conference Integration of Knowledge Intensive Multi-Agent Systems: Modeling, Exploration and Engineering. KIMAS-05. Boston: IEEE Boston Section. 2005, pp. 449–453.

http://is.ifmo.ru/articles_en/_kimas05-2.pdf

15. Ярцев Б. М., Шалыто А. А. Разработка программного обеспечения роботов Lego Mindstorms на основе автоматного подхода (Проект Isenguard)

http://is.ifmo.ru/projects/lego/

16. Шалыто А. А., Туккель Н. И. Танки и автоматы //BYTE/Россия. 2003. № 2, с 69–73. http://is.ifmo.ru/works/tanks_new/.

17. Шалыто А. А., Кузнецов Д. В. Система управления танком для игры «Robocode». http://is.ifmo.ru/works/tanks_new

18. Гуров В. С., Мазин М. А., Нарвский А. С., Шалыто А. А. UML. SWITCH-технология. Eclipse //Информационно-управляющие системы. 2004. № 6, с.12–17, http://is.ifmo.ru/works/uml-switch-eclipse/

19. UniMod project. http://unimod.sourceforge.net

20. Eclipse project. http://eclipse.org

21. Velocity project. http://jakarta.apache.org/velocity/

22. VI Открытая Всесибирская олимпиада по программированию им. И. В. Поттосина. http://olimpic.nsu.ru/widesiberia/archive/wso6/2005/rus/index.shtml

23. Model Driven Architecture. http://www.omg.org/mda/

24. Clarke E., Grumberg O., Peled D., Model checking. The MIT Press. 2000.

GRID-агенты самоконфигурации системы обмена сообщениями

Шарапов А. А., *ФГУП*
"Концерн "Системпром",
дом. адрес: Московская область, г.Щелково-3, ул. Стефановского, д.4, кв.5.,
E-mail: fatcrazer@itslan.ru
Коротков С. В. , *начальник НТЦ ВИВОСС и ОИ,*
дом. Адрес: г. Москва, 107392, Зельев пер., д.9, кв.25,
E-mail: skorotkov@vivos.ru

Создание современных территориально-распределенных информационно-управляющих систем требует наличия надежных и эффективных средств информационного взаимодействия между структурными элементами. Программно-технологической основой обеспечения информационного взаимодействия между программными компонентами может служить система обмена сообщениями.

Системы обмена сообщениями (Message Oriented Middleware) относят к одному из видов промежуточного программного обеспечения. Классическим примером MOM является система IBM MQSeries, часто применяемая при создании сложных ERP-систем.

Сложность целевых систем приводит к сложности администрирования обеспечивающих их функционирование систем обмена сообщениями. Необходимость интеграции нескольких гетерогенных сред в вычислительные системы выводит задачу администрирования на новый уровень.

Решить большую часть возникающих проблем в этой ситуации позволяет концепция саморегулирующихся вычислений (autonomic computing), т.е. построение вычислительных систем, способных самостоятельно управлять своей работой с учетом целей, поставленных администратором [1].

В работах, посвященных вопросам создания сложных вычислительных систем, упоминают о четырех аспектах самоуправления, а именно: самоконфигурация, самооптимизация, самовосстановление, самозащита.

В данной работе рассматривается применение одного из видов самоуправления при построении систем обмена сообщениями – самоконфигурации. Задача обеспечения самоконфигурации системы в изменяющихся условиях функционирования решается с использованием программных GRID-агентов методом построения промежуточного слоя самоконфигурации, накладывающегося на структуру системы.

GRID-технология – одна из наиболее активно развивающихся и актуальных концепций построения распределенных информационных систем, которая может быть применена не только для организации совместного использования распределенных вычислительных ресурсов.

Современные реализации Grid-сетей [3] базируются на концепции автономного агента – отдельной составляющей Grid-сети, способной на автономное функционирование и взаимодействие с другими агентами в сети.

Система обмена сообщениями включает следующие основные элементы:

- менеджер очередей – распределенная программная компонента, располагаемая на серверах и занимающаяся обработкой и передачей сообщений с промежуточным хранением в очередях;
- локальная служба – программная компонента, которая обеспечивает взаимодействие программных приложений, выполняемых на вычислительных средствах информационно-управляющей системы, с менеджером очередей.

В целях решения задачи самоконфигурации системы обмена сообщениями формируется GRID, которая состоит из автономных программных агентов, взаимодействующих с менеджерами очередей. Агенты собирают информацию о состоянии привязанных к ним менеджеров очередей и состоянии

транспортного канала до соседних менеджеров очередей. Используя собранную информацию и совместно взаимодействуя, агенты могут управлять привязанными менеджерами очередей, тем самым воздействуя на функционирование системы в целом.

Программные агенты обеспечивают подстройку системы по следующим критериям:

- гарантированность доставки сообщений;
- скорость доставки сообщений;
- равномерность нагрузки на узлы.

Одной из основных проблем при создании слоя самоконфигурации является разработка алгоритмов поведения программных агентов, особенно в условии неполной информации (в крупной динамической системе такая ситуация вполне возможна и допустима). Агенты должны быть способны принимать решения о конфигурировании привязанных к ним менеджеров очередей, используя всю доступную им на данный момент информацию, и принятое решение не должно нарушать функционирование системы в случае, даже если данные у агентов неполные и/или устаревшие.

Использование автономных агентов в создаваемой системе обмена сообщениями позволяет осуществить:

- мониторинг состояния отдельных менеджеров очередей;
- мониторинг состояния ближайших менеджеров и системы в целом;
- проведение воздействий на конфигурацию менеджеров очередей в зависимости от встроенных в агент алгоритмов;
- проведение воздействий на функционирование менеджера очередей в зависимости от анализа ситуации и глобальных настроек системы.

Список литературы

1. Джеффри Кефарт, Дэвид Чесс. Концепция саморегулирующихся вычислений. Открытые Системы №02 2003.

2. Thomas Myer. Grid computing: Conceptual flyover for developers. – IBM Developerworks, May 2003.

http://www-128.ibm.com/developerworks/grid/library/gr-fly.html

3. Леонид Бараш. Grid Computing - новая парадигма Internet-вычислений. - ITC Online, Компьютерное обозрение, №32, 22-29 августа 2001.

http://itc.ua/article.phtml?ID=7249

4. Alberto Montresor, Hein Meling. Messor: Load-balancing through a Swarm of Autonomous Agents. - Technical Report UBLCS-02-08, Dept. of Computer Science, University of Bologna, May 2002.

5. http://www.cs.unibo.it/projects/anthill/papers/2002-11.pdf

Инструментальный программный комплекс для проектирования мультиагентных систем

Швецов А. Н., *профессор каф. АВТ,*
e-mial: avt@vstu.edu.ru
Сергушичева М. А.,
e-mial: marias@list.ru
Котов С. С.,
e-mial: kotsobaka@mail.ru
Сорокин С. И.,
e-mial: sonvin@inbox.ru,
Вологодский государственный технический университет,
г. Вологда, ул. Ленина, 15

Проблема создания инструментальных средств проектирования мультиагентных систем (МАС) весьма интенсивно исследуется в последние годы. Разработан ряд инструментариев, ориентированных на решение подобных задач, но практически не существует средств, удовлетворяющих комплексу предъявляемых к ним на современном уровне требований [1]. Можно констатировать, что существует проблема создания инструментальных средств разработки МАС, сочетающих теоретически обоснованную методологию проектирования и эффективную реализацию в объектно-ориентированной среде.

В рамках совместных исследований кафедры "Автоматизированных систем обработки информации и управления" Санкт-Петербургского Государственного Электротехнического Университета (ЛЭТИ) с кафедрой "Автоматики и вычислительной техники" Вологодского Государственного Технического Университета ведется разработка интегрированной системы проектирования распределенных интеллектуальных систем (Distributed Intellectual System Integrated Toolkit – DISIT).

Процесс проектирования в среде DISIT рассматривает построение МАС на всех необходимых уровнях описания, начиная с обобщенной модели предметной области и заканчивая моделью информационного объекта, взаимодействующего с конкретным корпоративным приложением. Кроме того, в отличие от ряда инструментальных средств создания МАС, в этой концепции постулируется принципиальный отказ от использования конкретного языка программирования в качестве базового средства реализации распределенной системы. Вместо этого закладывается принцип интерпретации моделей интеллектуальных составляющих МАС соответствующими интерпретаторами подсистемы исполнения [2].

Комплекс DISIT включает в себя три подсистемы: концептуального проектирования, технического проектирования и времени исполнения. Подсистема концептуального проектирования содержит конструктор фрейм-концептов, логический конструктор, конструктор моделей поведения. Подсистема технического проектирования состоит из транслятора логической модели и конструктора баз знаний. Подсистема времени исполнения включает 4 менеджера: менеджер логического вывода, менеджер формальной объектной системы, менеджер поведения ИА, менеджер обмена сообщениями.

В концептуальной модели предметной области (КМПО), с описания которой с помощью конструктора фрейм-концептов (КФК) начинается проектирование МАС, используется фрейм-ориентированный подход, объединяющий фреймоподобные структуры с конструкциями концептуальных графов. Фрейм-концепт (ФК) определяется как

$$FK=<NF, TF, AI, SSB, SK, SA>,$$

где *NF* – имя фрейма, *TF* – тип фрейма, *AI* – информация о применении, *SSB* – структура сценариев поведения, *SK* – структура концептов, *SA* – структура атрибутов.

$$SK=\{(NK_1, KO_1), (NK_2, KO_2), \ldots, (NK_n, KO_n)\},$$

$$CA=\{(NA_1, SO_1, VA_1), (NA_2, SO_2, VA_2), \ldots, (NA_m, SO_m, VA_m)\}.$$

где *NK* – имя концепта, *KO* – тип концептуального отношения, *NA* – имя атрибута, *SO* – множество определения, *VA* – значение атрибута.

ФК такой структуры объединяются в сеть с определенной структурой по вложению концептов - FK-проекцию.

Логический конструктор (ЛК) на следующем этапе преобразует FK-проекцию в FKM-проекцию путём добавления в неё модулей с утверждениями. Для всех нетерминальных фрейм-концептов FK_i существует модуль концептуального графа M_i. Пусть в FK_i имеется структура концептов, включающая список $((FK_{i1}, KO_{i1}), (FK_{i2}, KO_{i2}), \ldots, (FK_{il}, KO_{il}))$. Тогда M_i может содержать утверждения над множеством $\{(FK_{ij}, KO_{ij})\}$ и вышестоящих по иерархии фрейм-концептов.

Конструктор моделей поведения (КМП) позволяет создавать необходимую иерархию интеллектуальных агентов (ИА). Структура поведения при этом может быть как одноуровневой (отметим, что даже в этом случае применяется арбитр сообщений для централизации управления другими агентами), так и многоуровневой. Сами агенты также могут иметь самые разные структуры: от простых - до сложных иерархических деревьев, элементами которых являются промежуточные абстракции - Информационные Объекты (ИО), в отличие от агента не способные взаимодействовать с окружающей средой и обладающие меньшей интеллектуальностью [3] – [5].

ИА можно представить следующей структурой: $LA = <NIA, SA, MVB, VO>$,
где *NIA* – имя интеллектуального агента, *MVB* – механизм выбора поведения, $VO=\{O\}$ - множество информационных объектов, реализующих сценарий поведения ИА, *SA* – структура атрибутов определяемая как $SA = <NA, DA, TA, VA>$, где *NA* - имя атрибута, *DA*- диапазон на котором определяется значение атрибута, *TA* – тип атрибута, *VA* – текущее значение в данный момент времени.

Для большей универсальности в структуре КМП заложен механизм описания ИА с использованием контекстно-свободных языков, определяемых пользователем.

Процесс трансляции концептуальной модели предметной области (КМПО) в логическую модель МАС начинается с завершенной FKM-проекции и состоит из следующих этапов:

1. Решение задачи выделения интеллектуальных компонентов (задача ВИК).
2. Построение дерева интеллектуальных компонентов. На данном шаге известны имена ИК, – это те модули, которые остались в результате решения задачи ВИК, содержание же ИК ещё не определено.

3. Построение дерева фрейм-концептов (декомпозиция FK-проекции по дереву ИК).
4. Трансляция дерева фрейм-концептов в логическую модель МАС. Для этого необходимо раскрыть структуру ИК, отображая дерево интеллектуальных компонентов в терминах ИК, ИА и ИО.

Конструктор баз знаний (КБЗ) создает структуру баз знаний (БЗ) для выделенных Транслятором логической модели ИК, отображая модели ИА и ИО в предусмотренные архитектурой базы знаний ИК структуры данных. В концепции DISIT была избрана смешанная стратегия построения БЗ, которая заключается в совмещении стратегий распределенного и централизованного подхода к конструированию БЗ. На КБЗ возложены две основные функции: создание базы знаний формальной объектной системы (БЗ ФОС), включающее определение областей действия аксиом (локальные, глобальные или распространяемые вниз по подчиненным ИК), и распределение баз знаний ИК по корпоративной сети.

Полученные БЗ, структуры ИК и ИА распределяются по вычислительной сети совместно с менеджерами подсистемы исполнения.

Менеджер формальной объектной системы (МФОС) позволяет изменять содержимое базы знаний ФОС: базу фактов (БФ), списки предикатных символов (ПС), локальных аксиом (ЛА), глобальных аксиом (ГА) и распространяемых сверху вниз аксиом (РА). Изменения, затрагивающие базы знаний подчиненных ИК (в списках ГА, РА и, возможно, ПС), запускают механизм сопоставления и обновления этих баз знаний.

Менеджер логического вывода (МЛВ) обслуживает запросы на осуществление операций логического вывода, поступающие от интеллектуальных агентов и объектов (ИА и ИО) через менеджера обмена сообщениями. Программа поддерживает два типа вывода: локальный и аддитивный, при котором базы знаний ИК, подчиненных данному, объединяются.

В качестве механизма логического вывода были использованы соответствующие средства среды Visual Prolog, для чего структуры базы знаний ФОС, представленные в формате KIF (Knowledge Interchange Format), транслируются в правила и факты Visual Prolog.

Менеджер поведения ИА (МИА) позволяет задать параметры выполнения моделей поведения, а также параметры связи и передачи данных. ИА, используя заложенные в него стратегии, выбирает определенные модели поведения, заложенные в ИО, которые имеют разные иерархические структуры. Модуль позволяет наблюдать за ходом работы агента и динамически изменять некоторые параметры.

Кроме непосредственного выполнения моделей поведения, МИА включает функции транспортировки сообщений и связи агента с внешним миром. С внешней средой и с другими агентами ИА связан через транспортный модуль. МИА постоянно поддерживает связь с сервером, получая через него сообщения от других агентов.

Менеджер обмена сообщениями (МОС) имеет две части: серверную и клиентскую. Сервер управляет транспортным потоком сообщений. С каждым подключенным клиентом связывается его идентификатор, очередь для простых сообщений, очередь для системных сообщений, очередь для событий. При его реализации использовался протокол TCP/IP с блокирующими соединениями и многопоточная архитектура. Для каждого присоединившегося клиента создается отдельный поток, что позволяет учесть применение двуядерных процессоров (AMD Athlon 64 X2, Intel Pentium D).

Транспортный модуль ИА (клиент) получает и отправляет данные во внешнюю среду по нескольким типам каналов, в его обязанность также входит получение и отправка сообщений другим агентам через сервер обмена сообщениями. Получая сообщения от разных агентов, сервер взаимодействует с диспетчером управления агентами, с помощью которого пользователь может видеть, что происходит с агентом данный момент, а также частично управлять ИА.

Текущая версия комплекса DISIT занимает 13,32 Мб на жестком диске. Минимальные требования для нормального функционирования МАС: процессор Intel Pentium II 266MHz, 16 Мб видео памяти, 64 Мб оперативной памяти. Для поддержания сетевого соединения достаточно модемной линии, однако для полноценной работы агентов могут потребоваться линии в 1 или 10 Мб/с. Быстродействие спроектированной МАС зависит от пропускной способности канала и режима работы агентов. При объеме БЗ в 64 кб среднее время обработки 1 логического запроса составляет менее 1 с.

Комплекс DISIT используется в научно-исследовательской и учебной работе кафедры Автоматики и вычислительной техники, в частности, для выполнения лабораторных работ по дисциплине «Интеллектуальные системы управления».

Список литературы

1. Нарушев Е. С., Хорошевский В. Ф. AgSDK: инструментарий разработки мультиагентных систем// Тр. VII национальной конф. по искусственному интеллекту с международным участием КИИ-2000. Т. 2. - М.: Физматлит, 2000. – С. 830-840.

2. Швецов А. Н., Яковлев С. А. Распределенные интеллектуальные информационные системы. – СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2003. – 318 с.

3. Швецов А. Н. Модель целеполагания интеллектуального агента в распределенных интеллектуальных системах/ Информатизация процессов формирования открытых систем на основе СУБД, САПР, АСНИ и искусственного интеллекта: Мат-лы 2-й межд. науч.-техн. конф. – Вологда: ВоГТУ, 2003. – с. 42 - 45.

4. Яковлев С. А., Швецов А. Н. Обобщенная модель интеллектуального агента в архитектуре распределенных интеллектуальных информационных систем/ Информатизация процессов формирования открытых систем на основе СУБД, САПР, АСНИ и искусственного интеллекта: Мат-лы 2-й межд. науч.-техн. конф. – Вологда: ВоГТУ, 2003. – с. 38-41.

5. Швецов А. Н., Черепанов А. В. Изучение проблематики построения интеллектуальных информационных систем в курсе дисциплины «Проектирование систем управления»/ Современные технологии обучения «СТО-2003»: Мат-лы IX международ. конф. 23 апреля 2003 г. – Т.2. – СПб.: СПбГЭТУ, 2003. – с. 50 -51.

Заметка об организации решения задач в Интеллсист

Шестаков М. А.
Отдел интеллектуальных систем
125252, Москва, Волоколамское шоссе, 1 оф. 412
E-mail: Shestakov@galaktika.ru

В докладе дается краткий обзор методов организации программирования, а также рассказывается о новых подходах к организации решения задач в Интеллсист, связанных с введением понятий анкет и шаблонов.

При решении задач методами программирования существует ряд критериев [1], удовлетворение которым позволяет оценить качество предлагаемого метода. Среди основных таких критериев можно выделить следующие:

1. ускорение решения задач (т.е. применение таких методов организации программирования, которые позволяют максимально сократить время программиста, требующееся для решения данной задачи и задач, связанных с данной);
2. уменьшение числа ошибок (применение методов, позволяющих уменьшить вероятность появления ошибок при программировании решения задачи);
3. распространение программы: расширение области исходных данных;
4. повторное использование программ (reuse, использование программ, созданных для решения некоторой задачи многократно для решения однотипных или связанных задач) [2].

Многолетняя практика программирования выработала значительное число методов организации решения. Кратко перечислим некоторые из них [1]:

1. последовательный (или линейный, традиционный подход с выполнением последовательности операторов шаг за шагом)
2. параллельный (выполнение набора связанных задач в различных процессах или потоках с синхронизацией)
3. циклический (перебор вариантов исходных данных, позволяет использовать одни и те же программы на разных наборах данных)
4. рекурсивный (вызов программы изнутри самой себя, что влечет сильное сокращение размеров программ)
5. функциональный (образование почек и кластеров для их многократного использования)
6. блочный (или уровневый, выделение некоторому блоку программы фрагмента памяти, который освобождается после выхода из блока)
7. управление языком (применение машины грамматического разбора)
8. табличный (представление условий, выполняемых в зависимости от них видов работ и переходов к следующему условию в табличном виде)
9. использование метаданных (применение так называемых настраиваемых программ, в которых некоторые параметры (или целые блоки программы) задаются извне и легко изменяемы)

10. модельный (создание программы по явно заданной физической модели)

11. прототипирование (быстрая разработка программ, не входящих в конечную реализацию, для отработки идеи, отладки, тестирования)

Интеллсист использует в том или ином виде значительную часть этих методов, но не ограничивается ими. В Интеллсист предлагается несколько новых подходов к организации решения задач, которые удовлетворяют перечисленным выше критериям качества. Эти подходы связаны с вводимыми в Интеллсист понятиями анкет и шаблонов.

1. Использование анкет в Интеллсист

Работа с помощью анкеты позволяет реализовать экспертные системы в обычном их понимании. Анкета – это модель экспертной системы, представляющая собой табличный список пар «термин-значение».

При создании анкеты разработчик определяет ряд вопросов (или терминов), на которые пользователь должен дать ответы (проставить значения) при заполнении. После заполнения анкета используется как форма запроса (или как его часть). После вставки анкеты в запрос над терминами, определенными в анкете, возможно применение любых операций, совместимых с их типом, т.е. их дальнейшее использование в запросе как данных. В Интеллсист существует два вида анкетных сущностей: исследовательская анкета и статистическая анкета.

Исследовательская анкета – это простейший вариант экспертной системы, о которой говорилось выше. Она представляет собой список выражений следующего вида:

<термин1> <отношение1> <значение1>; -- возможный комментарий1
<термин2> <отношение2> <значение2>; -- возможный комментарий1
…
<терминN> <отношениеN> <значениеN>; -- возможный комментарийN

При создании анкеты программист определяет термины, в нее входящие, и отношения, которые задают выражения. Отношения выбираются из предопределенного списка. После создания анкеты пользователь должен определить значения выражений (правые части) для всех терминов, входящих в анкету. Для создания и заполнения анкет в Интеллсист имеется удобный инструментарий с табличным интерфейсом. На этапах создания и заполнения исследовательские анкеты хранятся во внешних файлах. После заполнения исследовательская анкета может быть использована в запросе. При этом вставка в запрос осуществляется автоматически при указании файла с анкетой. Место вставки анкеты в запрос определяется положением курсора.

Статистическая анкета. Более интересными решением являются ***статистические анкеты***. По сути, они представляют собой расширение исследовательской анкеты до целого набора (множества) анкет, связанных единством использующихся в них терминов и отношений над ними. Подобная структура является очень полезной, если есть необходимость работы с сериями (последовательностями) различных значений для одних и тех же данных. Таким образом, статистическая анкета – это не один файл, а некоторый набор файлов, идентичных в указанном смысле. Однако значения терминов в каждом файле статистической анкеты могут быть различны.

Отличием статистической анкеты от исследовательской на этапе заполнения является отсутствие необходимости задавать отношение для термина – для простоты в данной версии Интеллсист считается, что в статистических анкетах термины и значения связаны операцией присваивания. По сути, при создании статистической анкеты программисту нужно разработать лишь некий эталон, в котором определяются термины анкеты. Далее, на этапе заполнения, пользователь должен «размножить» эталонный экземпляр, занося в него значения терминов для каждой итерации последовательности (или серии).

При вставке статистической анкеты в запрос пользователю нужно указать лишь один из файлов анкеты (который будет эталоном), а также директорию, содержащую остальные файлы анкеты. К анкете будут относиться все файлы, которые лежат в указанной директории и идентичны эталону в указанном смысле. Место вставки анкеты в запрос определяется положением курсора.

Преимущества использования анкет очевидны. Исследовательская анкета весьма полезна, если нужна возможность быстрого изменения данных или выражений для некоторой задачи. Используя одну и ту же анкету и изменяя в ней значения терминов, пользователь может легко получать результаты на всех требуемых наборах параметров, содержащихся в анкете. В первую очередь это ценно для начальной инициализации данных, использующихся в задаче. Кроме того, очень удобно использовать анкету для задания данным интересующего нас конкретного набора значений, например, для тестирования «узких» мест и аномалий программы.

Статистическая анкета также может найти очень хорошее применение. Вообще, такой подход естественно характерен на практике для решения задач, связанных с обработкой реальных анкет (выборы, опросы общественного мнения и т.д.). При наличии соответствующего оборудования для ввода данных из реальных анкет Интеллсист могла бы стать удобным инструментом для их обработки. Не ограничиваясь этим, в Интеллсист посредством статистических анкет можно

производить вычисления с любыми наборами данных, значения которых задаются сериями. Это может быть построение графиков для таблично заданных функций, обработка показаний приборов, статистические выкладки для характеристик однотипных продуктов и товаров и т.п.

2. Использование шаблонов в Интеллсист

Вопрос повторного использования данных, целых программ и их отдельных блоков является очень важным в организации программирования при решении однотипных и связанных задач. Это позволяет экономить время разработки программ, сокращает число ошибок в программах и вообще позволяет более структурно подходить к решению задач. Для достижения этих целей в Интеллсист введено понятие шаблона.

Шаблон в Интеллсист представляет собой запрос (или фрагмент запроса), в котором присутствуют неизменяемые (жесткие) блоки, выражения или текст и так называемые «окна» для ввода переменной информации. При создании шаблона программистом определяется «жесткий» текст шаблона, а также местоположения «окон» изменяемого текста. На этапе заполнения пользователь вносит в окна те значения (или текст), которые требуются для решения задачи в каждом конкретном случае.

Шаблон создается либо как запрос (шаблон-запрос), либо как произвольный набор текстов (шаблон-фрагмент). В первом случае за основу берется отработанный и отлаженный запрос, в котором определяются и формируются «окна» для пользовательского набора конкретных сведений, после чего получается новый запрос, который может быть задан на выполнение. Рутинная работа над одинаковыми (однотипными) заданиями поддержана таким видом шаблонов. Во втором случае шаблон служит в качестве набора фрагментов текста, которые используются для ускорения набора произвольных заданий. Это позволяет уменьшить число случайных ошибок. При создании запроса из шаблонов-фрагментов необходимо осуществлять вставку каждого шаблона в запрос. Инструментарий Интеллсист позволяет разработчику шаблона указывать комментарии к каждому «окну», чтобы пользователь при его заполнении имел представление о том, где и как будет применяться введенное им выражение.

В Интеллсист наиболее удобным методом хранения шаблонов является заведение специальных библиотек шаблонов. Это облегчает как поиск нужного шаблона, так и организацию хранения. Различные шаблоны могут храниться в разных библиотеках в зависимости от области их применения. Так, шаблон-запрос может использоваться для проведения различного вида тестирований, в которых одни и те же вычисления должны производиться на разных наборах данных, или наоборот, на одних и тех же данных применяются разные методы, вызовы которых могут ставиться в «окна».

Использование шаблонов-фрагментов весьма полезно в тех случаях, когда необходимо компоновать быстро программу (запрос) из готовых блоков. Это характерно, например, для процедурно-ориентированного программирования: часто использующаяся процедура или функция может быть описана в шаблоне-фрагменте, и при создании запроса осуществляется его подстановка.

В целом реализация анкетных и шаблонных сущностей в Интеллсист пока окончательно не завершена и может быть расширяема, однако уже можно сказать, что эти методы организации программирования достигают своих целей с точки зрения экономии времени, сокращения числа возможных ошибок и лучшей структурности при построении задачи. Вставка анкет и шаблонов обеспечивает выполнение указанных критериев.

Список литературы

1. www.intellsyst.ru

2. Красилов А. А., Лейтес В. Л. Риюз - повторное использование программ и язык Ада. - Киев, УСиМ, 1991, N 6, с. 57 - 68

Социологические проблемы дескриптивной теории принятия решений

Шкурко А. В.

К.соц.н., ассистент каф. социологии культуры и духовной жизни факультета социальных наук Нижегородского государственного университета им. Н.И.Лобачевского

603146, Нижний Новгород, ул. Норвежская, д.4, кв.123

Email: khanovey@rambler.ru, lost-future@yandex.ru

В данной работе рассматриваются основания социологически значимой теории принятия решения. В отличие от традиционных теорий инструментальной рациональности, основанных на понятии решения как рационального выбора из заранее заданных альтернатив, обосновывается влияние предварительного этапа моделирования ситуации на

логику принятия решения в реальной жизни. Проблемы типологии решений, моделирования и социальной структуры ситуации рассматриваются как важные составляющие дескриптивной теории принятия решения

Ключевые слова: Теория принятия решений, теория инструментальной рациональности, теория ограниченной рациональности, моделирование, социальная структура.

Понятие решения играет ключевую роль в ряде экономических, психологических и социологических теорий. Решение в данных теориях рассматривается как выбор из имеющихся альтернатив. Нормативные теории принятия решений исследуют вопрос о выборе оптимального, то есть максимизирующего функцию полезности варианта при заданных условиях задачи. Дескриптивные теории пытаются объяснить логику принятия решения в реальных ситуациях.

Ниже рассмотрены основные проблемы, которые должны быть решены в социологически значимой дескриптивной теории принятия решения (ТПР). Предполагается, что в сферу интересов социальных наук попадают решения, принимаемые субъектом социального действия (актором) и ориентированные на других субъектов или непосредственно затрагивающие их интересы.

Можно выделить два класса социологически значимых ТПР, разработанных в различных областях социальных наук. Экономические ТПР основаны на идеализации актора как рационального субъекта действия, поведение которого основано на принципе максимизации полезности и который в большинстве случаев, основываясь на доступной информации, совершает выбор наилучшего варианта. Различные варианты таких теорий, именуемых в дальнейшем теориями инструментальной рациональности, основаны на базовом принципе методологического индивидуализма и включают в себя теорию ожидаемой полезности (классический утилитаризм), применения байесовских методов (напр., Дж.Харсани), теоретико-игровые модели, экономический подход (Г.Беккер), теории коллективного действия (Дж.С.Коулмен, Дж.Элстер) и др., и исследуют как ситуации принятия индивидуального решения, так и взаимодействие различных акторов, а также – условия достижения равновесного состояния в масштабе всей популяции акторов (равновесие Нэша, Парето-оптимальность, и т.д.). В более современных вариантах теории инструментальной рациональности распространяют идеализацию рационального актора в том числе на неэкономические действия (вступление в брак, научное исследование, преступность и т.д.).

Альтернативный подход, разрабатывающийся, в первую очередь, психологами А.Тверским и Д.Канеманом, основан на экспериментальной критике основных допущений теорий инструментальной рациональности. В серии экспериментов было продемонстрировано, что принципы, в соответствии с которыми осуществляется выбор в реальных ситуациях, не соответствует ожиданиям теорий инструментальной рациональности, и включают в себя действие различных психологических механизмов. Альтернативная теория (которую иногда называют теорией ограниченной рациональности) предполагает выделение в процессе принятия решения двух фаз: фазы предварительного анализа ситуации («редактирования»), на которой действуют различные психологические механизмы, структурирующие ситуацию и упрощающие последующий выбор, и собственно фаза оценки как выбора наиболее предпочтительного варианта.

Социологические проблемы дескриптивной ТПР во многом также связаны с анализом условий и процессов, предваряющих собственно решение как выбор из эксплицитных альтернатив. Рассмотрим три основные группы таких проблем: типология решений, моделирование ситуации и социальная структура ситуации принятия решения.

1. Типология решений.

Представление о решении как о выборе из ряда альтернатив, основанном на рациональном анализе и прогнозировании возможных исходов, явно упускает из виду многообразие видов решений, принимаемых в реальной жизни. Классификация решений может осуществляться в соответствии с различными критериями. Перечислим некоторые важные критерии классификации и типы решений:

a) В соответствии с когнитивной структурой решения (основным когнитивным механизмом, лежащим в основе принятия решения) можно выделить проспективные решения (связанные с прогнозированием возможных исходов и ориентацией на достижение конкретной цели) и нормативные решения (связанные с оценкой объекта, установлением степени его соответствия некоторым нормативным критериям),

b) В соответствии с общей структурой ситуации принятия решения могут быть закрытыми (имеющими конечный, фиксированный набор альтернатив) или открытыми (не имеющими фиксированного набора альтернатив);

c) В зависимости от времени и иных ресурсов, доступных для принятия решения: стратегические (основанные на тщательном анализе достаточного количества релевантной информации,

достаточных временных и иных ресурсах, возможности и готовности просчитывать отдаленные последствия принимаемых решений) и реактивные (принимаемые в условиях дефицита времени и/или иных необходимых ресурсов). Эта типология, безусловно, является ключевой для всей ТПР и, строго говоря, должна быть разработана более подробно, что, к сожалению, не может быть сделано в рамках данной работы.

d) По степени эксплицированности ситуации принятия решения можно выделять рефлексивные, нерефлексивные и рутинные решения;

e) В зависимости от числа субъектов принятия решения: индивидуальные и коллективные;

f) В зависимости от цели принятия решения: ее конкретности, новизны и т.д., определяющей сравнительную значимость и вес разных стадий принятия решения.

Значение указанных типологий для дескриптивной ТПР связано не только с изучением принципов, лежащих в основе различных типов решений, но также с анализом социального распределения различных типов ситуаций принятия решений и др. Можно, в частности, отметить, что применимость и уместность различных формальных моделей принятия решений, основанных на идеализации рационального субъекта (в частности, применения теоретико-игровых моделей в формировании политики во время холодной войны, в экономических процессах и т.д.) является отнюдь не универсальной, а исторически обусловленной. Успех в применении таких методов для анализа реальных процессов зависит от степени приближенности объекта моделирования к набору идеальных свойств, заложенных в саму модель (предположение о субъекте как о рациональном агенте, который стремится к максимизации собственных выгод, предположение о стабильности предпочтений, исходные оценки ожидаемой полезности различных исходов и т.д.).

2. Моделирование ситуации принятия решения.

Различные подходы, основанные на понятии решения, исходят из представления о наличии некоторых четко различающихся альтернативных вариантов как из чего-то само собой разумеющегося, как если бы любая данная ситуация была *действительно* разбита на ряд дискретных состояний чьей-то заботливой рукой, как если бы эти варианты существовали *до* и *независимо* от ситуации принятия решения.

Мы утверждаем, что адекватная дескриптивная ТПР должна включать в себя не только анализ процесса принятия решения, но и анализ моделирования самой ситуации принятия решения.

Таковой анализ должен включать в себя следующие компоненты:

- изучение механизмов схематизации, действующих в той или иной ситуации, которые могут быть связаны с предыдущим опытом субъекта, имеющимися когнитивными схемами (например, дисциплинарными), контингентными условиями, и др.

– анализ логической сложности модели ситуации принятия решения,

– оценку вычислительной сложности принятия решения,

– оценку времени, доступного для принятия решения,

– анализ компетентности субъекта принятия решения,

– анализ социальной структуры ситуации (см. следующий раздел).

Принятие решения очевидно зависит от модели ситуации, имеющейся у актора. Моделирование ситуации включает в себя артикуляцию альтернатив, оценку вероятностей и ожидаемой полезности, и включает в себя ряд этапов: идентификацию ситуации принятия решения, предварительную оценку важности ситуации, ее логической сложности, времени, необходимого как для ее моделирования, так и для собственно принятия решения.

Логическая и математическая структура ситуации при этом не может приниматься как данность – в любой социологически релевантной теории. Она является следствием моделирования и может быть разной для разных субъектов, включенных в ситуацию. Различные субъекты, обладая различными наборами предпочтений, а также различным уровнем компетентности, будут формировать различные модели ситуации.

Компетентность в данном случае определяется как способность актора идентифицировать и моделировать ситуацию, основываясь на имеющемся запасе знаний и наборе предпочтений, и принимать решение, соответствующее данной модели и подчиняющееся принципу рациональности. В качестве аксиомы можно принять, что компетентность положительно коррелирует с логической сложностью модели, то есть, более высокий уровень компетентности позволяет давать более детальное описание

ситуации. Однако различные описания одной и той же ситуации (даже если нечто одинаково идентифицируется в качестве одной ситуации) – это различная логическая структура ситуации, воспринимаемой разными субъектами, а следовательно, различные функции оптимальности (см. следующий раздел).

Логическая сложность ситуации (воспринимаемой посредством модели), вычислительная сложность моделирования и собственно процесса принятия решения, а следовательно, время и иные затраты (например, связанные с использованием вычислительных мощностей) на процесс принятия решения (а не на сами действия, являющиеся его следствием), в совокупности составляют транзакционные издержки при принятии решения. Наш главный тезис заключается в следующем: транзакционные издержки входят в полную стоимость решения, а их предварительная оценка влияет на логику принятия решения. В частности, это означает, что при значительном росте транзакционных издержек (например, при увеличении логической сложности ситуации) точка оптимальности в принятии решения будет смещаться в сторону реактивной модели.

3. Социальная структура ситуации

Любая социологически релевантная теория принятия решений, как мы уже указывали, должна анализировать не только сам процесс «взвешивания альтернатив», но и структуру самой ситуации, в пределах которой это взвешивание осуществляется. Существенным компонентом структуры ситуации принятия решений является ее социальная структура. В особенности это касается решений, принимаемых коллективно, однако не только. В тезисной форме ее важность в анализе принятия решений может быть обозначена следующим образом.

a) Логическая структура ситуации принятия решения для различных субъектов, составляющих релевантную популяцию, не является одинаковой. Различные субъекты обладают, как правило, различными моделями ситуации, разной степенью ее эксплицированности, различными функциями полезности, и т.д.

b) Входящие в данную популяцию индивиды обладают различными возможностями участия в процессе принятия решений – причем к числу этих возможностей относится как компетентность, так и статусная позиция (см. п.3).

c) Социальная структура данной популяции образована сетью социальных отношений, обладающих различной силой и вектором, полюса которых формируют позиции, *которые и следует рассматривать в качестве субъекта принятия решений*. Таким образом, субъект принятия решений не идентичен индивиду – как это предполагается в базовых версиях теорий рационального выбора. Выражаясь ортодоксальным социологическим языком, социальная структура образована дифференцированными социальными статусами и ролями, сеть которых не совпадает с сетью индивидов. Представление о том, что в качестве основного субъекта принятия решения выступает статусная единица, позволяет, во-первых, дать социологически более корректную формулировку представлениям экономического подхода о наличии «конкурирующих целей» (а следовательно – функций полезности для данного индивида), а во-вторых, позволяет говорить о наличии социальной структуры ситуации принятия решения даже когда речь идет об *индивидуально принимаемом* решении.

d) Социальная структура ситуации принятия решения напрямую влияет на логическую сложность как модели ситуации, так и самого процесса принятия решения, увеличивая транзакционные издержки при количественном и качественном возрастании сложности социальной структуры. Идет ли речь об индивидуально или коллективно принимаемых решениях, степень ролевой дифференциации при прочих равных условиях увеличивает время и иные затраты на принятие решения.

e) Одним из основных факторов, структурирующих ситуацию принятия решения, является характер социального распределения экспертного знания и компетентности в пределах данной популяции. Анализ характера распределения экспертного знания в данной популяции важен по следующим причинам. Во-первых, уровень компетентности субъекта принятия решения влияет на когнитивную модель ситуации, а следовательно, и набор артикулированных альтернатив. Конкретный вектор такового влияния может быть различным: возрастание компетентности может приводить как к увеличению и/или усложнению набора альтернатив (вследствие более глубокого анализа ситуации), так и его уменьшению/упрощению (вследствие обнаружения скрытого сходства в эмпирически различных ситуациях). Однако и в том, и в другом случае логическая и математическая структура ситуации принятия решения будет отличаться в зависимости от уровня компетентности субъекта. Во-вторых, как показывают исследования в области социологии научного знания, наличие несовпадающих интересов, ассоциированных с различными статусными позициями в пределах некоторой социальной сети,

значительно увеличивает мотивацию участников к артикуляции различных интеллектуальных позиций. В нашем случае это означает, что социально обусловленная экспертная гетерогенность популяции будет способствовать формированию различных моделей ситуации принятия решения, а как следствие – различных представлений об оптимальном решении.

Перспективы алгоритмизации процессов творческого мышления в форме искусственного интеллекта

Шкурко Ю. С.
ННГУ им. Н.И. Лобачевского
к. соц. н., старший преподаватель кафедры
социологии культуры и духовной жизни
Нижний Новгород, 603146, ул. Норвежская 4, кв. 123
e-mail: kiouls@mail.ru

В когнитивных науках – междисциплинарной исследовательской области, объединяющей усилия когнитивных психологов, нейрофизиологов, специалистов в области искусственного интеллекта (AI) и компьютерного моделирования – сегодня доминирует и, соответственно, активно используется информационно-кибернетическая модель когнитивных процессов, сформировавшаяся во второй половине XX века. В этой модели мозг/ психика человека рассматривается как сложная информационная система, в которой протекают процессы восприятия, переработки, хранения, продуцирования и передачи информации. Неоднократно предпринимались попытки алгоритмизации этих процессов, как по отдельности, так и в их совокупности. Это привело, в частности, к созданию мощных компьютерных программ, в которых эти попытки приняли форму решателей задач, экспертных систем, генераторов литературных произведений и других форм AI. Более того, алгоритмизация мыслительных процессов зашла так далеко, что вычисления, которые производит машина, не всегда под силу человеку. Так, в математике известен факт доказательства компьютером теоремы о четырех красках, которое не может быть проконтролировано человеком-математиком, что вызывает естественные споры относительно валидности полученного машиной научного результата [этот инцидент в социологическом ключе рассмотрен в: 1].

Несмотря на успехи в области AI, свидетелями которых мы являемся, перспективы дальнейшего развития в этой области выглядят не очень радужно. Прошло достаточно много времени с первых попыток алгоритмизации и моделирования процессов мышления, чтобы оценить все возможности выбранной информационной стратегии. Основным барьером для дальнейшего продвижения в этой области, по-видимому, является упор исследователей в области математического и компьютерного моделирования мыслительных процессов на достижения только когнитивной психологии и когнитивной нейронауки (то есть, акцентирование внутренних процессов, протекающих в мозге и на уровне психики). На наш взгляд, выходом из сложившейся тупиковой ситуации является переориентация внимания исследователей на социологические компоненты процесса творческого мышления и поиск закономерностей творческой деятельности (это нужно сделать и потому, что представление о мозге как о единственном органе, в котором протекают процессы творческого мышления и «создается» новая идея, кажется сомнительным). Первый шаг, как нам кажется, следует сделать в направлении выявления детерминант творческого мышления – процесса, в результате которого человек создает новые идеи. Соответственно, речь должна идти о математической модели и компьютерной имитации не столько мыслительных процессов, сколько творческой деятельности вообще.

В качестве детерминант творческого мышления, которые должны включаться в математические модели AI и, соответственно, пройти через процедуру алгоритмизации, прежде всего, необходимо обратить внимание на особенности информационный среды, которая окружает человека. Соответственно, основные усилия исследователей должны быть направлены на попытки алгоритмизации связей между особенностями этой среды и конкретными продуктами творческого мышления, их содержанием и способом (алгоритмом) его создания. Иными словами, *принципиально важными являются связи между тремя переменными: информационной средой, алгоритмом создания новой идеи и самой этой новой идеей (результатом акта мышления).*

Информационная среда.

Наиболее важными параметрами информационной среды, оказывающими влияние на процесс мышления, по-видимому, являются следующие:

– информация из прочитанной актором процесса мышления литературы (научной и художественной), которая в определенной степени воздействует на него. При этом имеет значение, о какой именно информации идет речь. Один из вариантов ее кодификации – это представление по тематическим рубрикам. Число этих рубрик достаточно велико. В качестве предварительной классификации этих тематически-информационных рубрик можно взять принятые сегодня в науке рубрикаторы научных дисциплин (например, ГРНТИ и УДК) и библиотечные классификаторы, которые, безусловно, не могут рассматриваться как окончательные;

– информация из СМИ (печать, радио, электронные медиа). Кодификация этой информации также должна проходить по тематическим рубрикам, в которых обращается информация. К таким рубрикам относятся новости, комментарии, развлекательные программы, документальные программы, реклама и др. [один из вариантов рубрикации СМИ предложен в: 3];

– информация, которая поступает к актору процесса мышления из повседневного окружения. Данную информацию, как нам кажется, имеет смысл кодифицировать для дальнейшего включения в общую модель процесса творческой деятельности путем указания на характер этой информации (скажем, является ли она положительным или отрицательным подкреплением для AI).

Что касается математической кодификации перечисленных выше аспектов информационной среды, то здесь возможны, по крайней мере, три стратегии:

– AI самостоятельно черпает информацию из окружающего мира;

– человек вводит случайным образом отобранную информацию о мире (из перечисленных выше первых двух информационных блоков);

– человек снабжает AI максимально полной информацией о мире, а «он(а)» случайным образом отбирает определенные идеи и на их основе генерирует новые.

Дополнительно в процессе машинного творчества (или при его завершении) необходимо вводить информацию о желательности либо нежелательности того или иного результата (как это происходит иногда, скажем, в процессе творческой деятельности писателя/ ученого, которые порой вносят коррективы в свою работу по требованию заказчика/ после разговоров с коллегами).

Алгоритм создания нового.

Этот блок процесса творческого мышления в наибольшей степени алгоритмизирован. Так, например, большой вклад в алгоритмизацию процесса творческого мышления был внесен в 1980-е годы Г. Альтовым-Альтушеллером – создателем ТРИЗ и автором этажного метода создания новых идей. В эти же годы П. Амнуэлем было предложено порядка двадцати алгоритмов создания новой идеи, рассматриваемых на примере научно-фантастической литературы [См., в частности, 2]. Достаточно много добротного материала, который требует своего обобщения в рамках рассматриваемой здесь концепции творческой деятельности, накоплено и в области программирования творческих процессов, решателей задач и других областях. Задача, требующая своего первоочередного решения, заключается в перечне всех возможных алгоритмов создания новой идеи. Возможно, уже на этом этапе рассмотрения проблемы удастся выявить некоторую специфику относительно, скажем, алгоритмов мышления ученого и писателя, что при установлении связей с рассматриваемыми двумя другими переменными позволит установить причину этого.

В настоящее время «изобретатели» алгоритмов мышления обычно действуют в двух направлениях. Одни из них выносят за скобки процессы, происходящие на уровне человеческой психики и мозга, также как и в окружающей актора среде при создании новой идеи, беря во внимание только поступающую на «вход» информацию и ту, которая в итоге из нее получается («выход»). Иными словами, в этой модели алгоритмизации творческих процессов считается: если A, то B посредством C, где A – информация на входе (рассматриваемая в содержательном аспекте), B – информация на выходе (также рассматриваемая в содержательном плане), C – возможный алгоритм (алгоритмы), приводящий от A к B, который придумывается «изобретателем» алгоритма *post factum.*

Как мы видим, в данном подходе исследователей интересует только информационный уровень – процесс превращения информация из одной в другую (новую) и совсем не берется во внимание, что данную операцию с информацией производит сам человек, будучи существом в определенной степени зависимым от своего окружения. Рассуждая подобным образом, представители данного направления исследования процесса творческого мышления, ставят перед собой цель найти все возможные алгоритмы создания «нового». В рамках данного подхода, действительно, становится неважным, в какой степени осознан самим актором процесс мышления, насколько он контролирует его. Возникает закономерный вопрос – является ли этот путь наилучшим? Не упускаем ли мы что-то важное в процессах творческого мышления? Не превращаются ли наши рассуждения о творческом процессе в реконструкцию процессов, которые реально протекают иначе? Является ли это расхождение между реальным творческим процессом (который зачастую принимает форму инсайта) и реконструируемым столь уж принципиальным для перспектив машинной алгоритмизации творческого мышления?

Ответ на этот вопрос, безусловно, будет положительным, если мы придерживаемся другого подхода к алгоритмизации процессов творческого мышления, сформированного в «сильной» версии AI,

представители которой ставят перед собой цель создать аналог человеческого интеллекта. Как уже было сказано выше, в «сильной» версии AI акцент делается на имитации нейро-когнитивных процессов, протекающих в мозгу и психики человека, разумеется, без учета социологических факторов. Учет социологических факторов предполагает алгоритмизацию связей между «информационными ситуациями» (образуемых из разного сочетания трех вышеперечисленных параметров информационной среды) и конкретным алгоритмом создания нового. «Выбор» алгоритма реальным человеком, отчасти, зависит от информационной ситуации, что должно найти отражение в алгоритмизации указанной связи. Безусловно, на «выбор» алгоритма оказывает влияние и тип мышления (конвергентный или дивергентный) и опыт и др., которые в перспективе также необходимо включать в математическую модель творческой деятельности.

И, наконец, *содержание новых идей*.

Видимо, есть смысл кодифицировать все многообразие получаемых в ходе размышления новых идей, посредством выявления того, каким образом они отличаются от родственных и предшествующих им идей. Они могут выступать, прежде всего, в качестве

– альтернативы уже существующим идеям;

– замены существующим идеям;

– принципиально необычного и несравнимого с предшествующими идеями.

На первый взгляд может показаться, что данная переменная - «содержание новых идей» - может быть опущена без ущерба для модели творческой деятельности, подлежащей дальнейшей алгоритмизации в форме AI. Казалось бы, какой смысл усложнять себе задачу? От такого «усложнения», как нам кажется, ни в коем случае нельзя отказываться. С первых шагов создания творящего AI мы должны иметь механизмы контроля над содержанием производимого «им» знания, которые отчасти получаем в форме двух других связанных между собой параметров.

Таким образом, перспективы алгоритмизации процессов творческого мышления в форме AI, как нам кажется, в создании математической модели творческой деятельности и дальнейшей ее компьютерной имитации. Ее создание оказывается возможным при выходе за пределы алгоритмизации когнитивных процессов, имеющих место в мозге и психики человека, а также при расширении границ рассмотрения алгоритмов мышления, что предполагает увязывание воедино трех переменных – параметров информационной среды, алгоритмов мышления и содержательных особенностей новых идей. Кроме того, установление указанных связей оказывается эвристичным и с прогностической точки зрения: если мы хотим предсказать, когда и какое открытие будет сделано, то здесь без учета факторов внешней среды не обойтись - только она и сможет нам это подсказать.

Список литературы

1. MacKenzie D. Slaying the Kraken: The Sociohistory of a Mathematical Proof // *Social Studies of Science*. 29/1 (February), 1999, p. 7-60

2. Амнуэль П. Магический кристалл фантазии (отрывки из учебного пособия для инженеров и изобретателей) // URL: http://www.trizminsk.org/e/246006.htm

3. Луман Н. Реальность массмедиа. М.: Праксис, 2005

Анализ бизнес - процессов компании на основе использования инструментальных средств

Шматко А. Д.
канд. эконом. наук, декан факультета повышения квалификации преподавателей ЗТУ
Тел.: (812) 8-906-2568077, mailto: Shmat2000@yandex.ru
Шаповалова Н. А.
Руководитель отдела сертификации
ООО «Полиграфоформление ОФСЕТ XXI век»
Тел.: (812) 8-911-9463502, mailto: katzerine@mail.ru

В современных условиях жесткой конкуренции компании обеспечивают эффективное функционирование только создавая управляемые условия развития бизнеса и определяя наименее затратные пути выполнения процессов выпуска продукции или оказания услуги. Перед руководителями организаций стоят задачи построения, моделирования, разработки и дальнейшего совершенствования

системы управления компанией. Для решения последней задачи необходимо осуществлять мониторинг и анализ модели системы управления с целью осуществления контроля над организацией и установления возможностей совершенствования и перепроектирования системы управления для достижения позитивных деловых результатов по сравнению с конкурентами.

Как известно, различные компании применяют сопоставимые технологии и сходные ресурсные компоненты, удовлетворяющие требованиям рынка, включающим минимизацию себестоимости продукции при обеспечении ее высокого качества.

Себестоимость продукции включает укрупненно:

- стоимость сырья и материалов;

- затраты на обслуживание инфраструктуры;

- обеспечение необходимых человеческих ресурсов;

- стоимость изготовления продукции и др.

Авторские исследования определили перспективность минимизации себестоимости продукции за счет уменьшения затратной части бюджета процессов жизненного цикла продукции; при этом следует учитывать, что фактически при традиционном подходе к определению стоимости бизнес - процессов поддается оценке не более половины выполняемых функций.

В этой связи авторами предлагается использование инструментальных средств позволяющих проанализировать работу компании, установить возможности по оптимизации процессов жизненного цикла продукции, в том числе оптимизации стоимости выполнения процессов.

В контексте данного подхода предлагается осуществлять классификацию процессов компании по признаку создания добавочной стоимости на основные и вспомогательные.

Последовательность основных процессов отражена на диаграмме *Value-added chain diagram* (рис. 1). В диаграмме *Value-added chain diagram* в объектах *Objective* и *Key performance indicator* задаются цель выполнения процесса и показатель результативности процесса. В соответствии с ГОСТ Р ИСО 9001 [1] результативность процесса определяется степенью достижения установленной цели. Оценка результативности осуществляется через анализ показателей процесса и последующим сравнением показателей с установленным критерием достижения цели процесса. Эффективность определяется отношением результата процесса выраженного в виде финансового мультипликатора к использованным ресурсам на выполнение процесса.

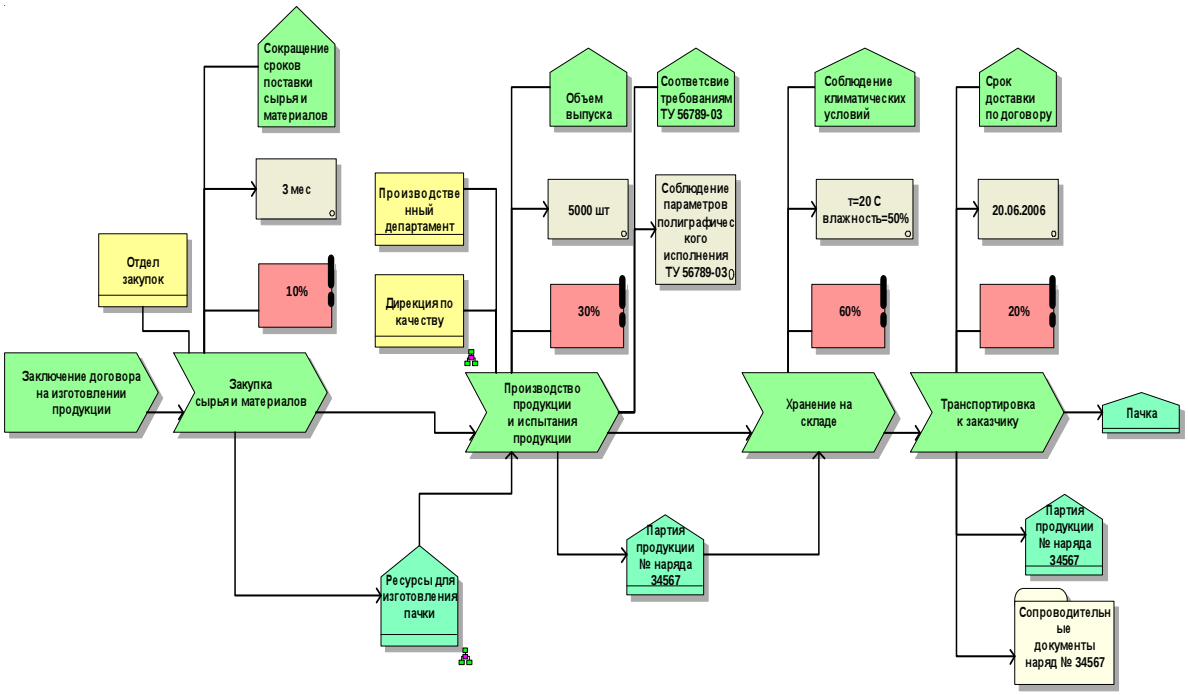


Рисунок 1 - Цепочка основных процессов компании

В процессе авторских исследований выявлены проблемы адекватного представления системы управления организацией при разработке модели: установление оптимального количества элементов системы управления, идентификация и классификация связей между элементами системы управления. Для разрешения выявленных проблем предлагается на системной основе осуществлять построение модели деятельности компании в виде описания связанных процессов. Для этого рекомендуется к использованию следующие проекции: организационная структура; база знаний компании; ресурсы компании, процессная структура. Модель деятельности компании строится в перечисленных проекциях, описываются элементы в каждой из проекций, устанавливая связи между элементами первоначально в рамках одной проекции, затем между проекциями.

Авторами осуществлен анализ возможностей описания управления компании в вышеуказанных проекциях на основе использования инструментального средства *Aris Toolset*.

В диаграмме *Organizational Chat* формируются данные о составе и характере подчиненности сотрудников компании. С применением данной диаграммы можно описать структуру и иерархию подразделений компании, отобразив подчиненность отделов и сотрудников компании. Возможности диаграммы позволяют описать организационно - структурные единицы, сотрудников, указать местонахождение сотрудников и дать развернутое описание организационно-структурной единицы.

В диаграмме *Knowledge structure diagram* отражается информация о структуре документированных и недокументированных знаний компании. Информация упорядочивается по принципу иерархичности и типу хранилища данных: бумажный носитель, электронный носитель. В меню *Attributes* можно задать характеристики знаний, включая период актуализации знания и ответственного за актуализацию.

Диаграмма *Knowledge Map* позволяет разработать адекватную модель структурированных корпоративных знаний по различным признакам, а также позволяет путем последовательной детализации исходной совокупности информации последовательно разложить ее до уровня необходимой глубины, показав вложенность одного массива знаний в другой. Существует возможность классификации используемых знаний по носителю информации и по источнику информации [2].

В диаграмме *Material diagram* и *Product tree* описываются ресурсы необходимые для изготовления конечного продукта.

В диаграмме *eEPC* формируется событийная цепочка последовательности выполнения процесса. С помощью операторов правил возможно задать необходимые пути прохождения процесса. Первоначально задается последовательность выполнения процесса, далее из диаграммы *Organizational Chat* добавляется информация об исполнителях функций процесса устанавливая тип связи – характеризующий вид ответственности исполнителя по отношению к функции. Для отражения ресурсного обращения добавляется информация из диаграммы *Product tree*. Для установления документального обращения процесса и составления информация о необходимых документальных источниках процесса информация добавляется из *Knowledge structure diagram*. В меню *Attributes* для каждой функции процесса задается стоимость выполнения. Таким образом, появляется возможность определить стоимость выполнения процесса при различных вариантах его прохождения, задаваемых логическими операторами правил и установить эффективность процесса взяв информацию о результативности процесса из данных диаграммы *Value-added chain diagram*.Для вычисления стоимости процесса, необходимо знать стоимость каждой функции процесса и периодичность ее выполнения. В опции *Variants*, задаются варианты реализации процесса. Варианты моделей позволяют рассмотреть несколько моделей исследуемого процесса.

Создавая варианты модели процесса и осуществляя анализ стоимости выполнения каждого варианта руководство компании определяет наименее затратный вариант достижения цели процесса. Анализ процессов компании на основе использования инструментальных средств в части вышеуказанных диаграмм является эффективным инструментом управления как отдельными процессами, так и системой управления в целом.

Список литературы

1. ГОСТ Р ИСО 9001:2001 Система менеджмента качества. Требования. М.: ИПК Издательство стандартов. 2001;

2. Шаповалова Н. А., Шматко А. Д. Обеспечение конкурентоспособности компаний на основе организации базы знаний. Региональные аспекты управления, экономики и права Северо-западного федерального округа России. Выпуск 2. Межвузовский сборник научных трудов/ Под ред. д-ра экон. наук, академика МАНЭБ А. Д. Макарова, д-ра экон. наук, проф. М. Д. Медникова, д-ра воен. наук, проф. А. А. Целыковских. — СПб.:Изд-во ВАТТ, 2006

Вербальные компоненты NFL-континуума как структура и система

Якимов Владимир Николаевич,
доктор технических наук, профессор кафедры
информационных технологий
Самарского государственного технического университета
Служебный адрес: 443100 Самара, ул. Молодогвардейская, 244, Гл.корпус
Домашний адрес: 443082 Самара, ул. Пензенская, дом 64, кв. 6
Дьяконов Герман Николаевич,
старший преподаватель кафедры информационных технологий
Самарского государственного технического университета (СамГТУ)
Служебный адрес: 443100 Самара, ул. Молодогвардейская, 244, Гл. корпус
Домашний адрес: 443020 Самара, ул. Самарская, дом 40, кв. 10
Машков Андрей Валерьевич,
кафедра информационных технологий
Самарского государственного технического университета
Служебный адрес: 443100 Самара, ул. Молодогвардейская, 244, Гл. корпус
Домашний адрес: 446006 Самарская обл., пр. Победы, дом 3-Б, кв. 32

Работа посвящена рассмотрению проблемы генерации вербальных компонент NFL-континуума: от планирования содержания до получения самого NFL-континуума. Понятие NFL–континуум было введено в [1] и под ним понимается вся совокупность машино- и человекочитаемых источников информации на естественных и формализованных языках. Далее для простоты используется термин текст. Общая теория генерации текста представляет собой соединение ряда частных теорий (теория дискурса, теория фокуса внимания, теория референции, теория коммуникативных неудач) в комбинации с такими фундаментальными науками, как психология, семиотика, семантика, математическая лингвистика и т.п.

Необходимость в проведении подобных исследований возникла в связи с созданием в лаборатории интеллектуальных технологий при кафедре информационных технологий Самарского государственного технического университета системы СИГМА (Система Интерпретации Генерируемых Моделей Актантов). Под актантом понимается субъект некоторых действий: например, Персонаж, Животное, Дерево, Степь и т.п. («Степь дремала под лунным светом»). Каждый Актант обладает двумя наборами атрибутов — атрибуты класса и атрибуты индивида. Атрибуты класса частично наследуются от атрибутов суперкласса. Атрибуты индивида состоят из свойств конкретного объекта, которые делятся на обязательные, факультативные и запрещающие.

Интерпретация Моделей Актантов заключается в том, что сгенерированный на промежуточном языке моделей код преобразуется в текст на естественном языке. При генерации и интерпретации используются две операции: развёртывания и конкретизации. Для операции развёртывания используется генератор псевдослучайной последовательности целых чисел. Он определяет направление развития сюжета. Операция конкретизации состоит в том, чтобы придать сгенерированному фрагменту промежуточного кода черты реального текста. Для этого, прежде всего, необходимо учесть хронотопические ограничения и допущения. Далее надо учитывать уже введённые в накопленном коде налагаемые им ограничения и предоставляемые им возможности.

Сначала определим некоторые базовые ограничения. Во-первых, тема последующих заметок не претендует на всеобъемлющее изучение любого текста. Она касается лишь тех аспектов текста в его обыденном понимании, которые релевантны проблемам, возникающим при порождении текстов системой СИГМА. Иначе говоря, это теория текста *ad hoc*. Во-вторых, следует тем же образом ограничить понятия структуры и системы. Для начала, предположим, наличие некоторого набора элементов. Находясь в некоторых отношениях друг к другу, элементы образуют структуру. Эта структура, эволюционируя с течением времени, образует (динамическую) систему. С первого взгляда может показаться, что текст, рассматриваемый как нечто уже существующее, является статичным, неподвижным, и должен рассматриваться, по нашим соглашениям, лишь как структура. Однако выдающийся теоретик театра Патрис Пави указывает: «Было бы неверно рассматривать драматургический текст как фиксированную сущность, понимаемую один раз и навсегда. В действительности текст существует только по прочтении, которое всегда исторически обусловлено. Это прочтение зависит от социального контекста читателя и его знания контекста вымышленного текста» [2]. Хотя напрямую Патрис Пави говорит о драматургическом тексте, вышеприведённое высказывание можно отнести к текстам любого типа – от инструкции по применению какого-либо более или менее сложного технического устройства до трагедий Шекспира и сценариев «мыльных опер». Таким образом,

при прочтении текст имеет свойство «меняться». Ещё большими возможностями к изменению своей структуры текст обладает на этапе его создания, генерации, порождения, либо автором, либо программой.

В цели рассмотрения не входит определение понятия «текст». Опыт показывает, что споры о дефинициях способны занять время и силы участников любого творческого процесса, причём часть участников будет создавать определения, а другая часть – подбирать опровергающие примеры. Под термином «Текст» в п. 10.2 «Словаря терминов по информатике» [3] понимается «сообщение, выраженное в письменном виде, т.е. при помощи какой-либо системы графических знаков … ; значащая, информативная часть письменного документа». Понимая под документом любой материальный объект в аспекте использования его как источника информации, информатика даёт максимальное по экстенсионалу определение текста.

Будем считать текстом то, что кажется текстом на интуитивном уровне, то есть совокупность грамматически и семантически связанных предложений, описывающих одну или несколько ситуаций, возможно, статических, либо систему взаимно преобразующихся ситуаций. В последнем случае будем говорить также о сюжете текста.

Необходимо отметить, каким образом определяется общий ход сюжета. Базовой сюжетной линией, так называемым «нулевым», или «скелетным» сюжетом, является в конечном итоге некоторое псевдослучайное целое число. Его можно интерпретировать как один из вариантов гадальных карт. В связи с этим соображением в зависимости от количества карт N число нулевых сюжетов может ограничиваться числами N! В частности, исходя из аналогии с игральными картами, оно ограничивается числами 36!, 52! или 104! Очевидно, эти числа неадекватно передают число разных сюжетных линий, Даже один и тот же расклад приводит к значительным вариациям в дальнейшем развитии сюжета, ибо управление им носит случайный характер.

Существует большое количество традиционных интерпретаций раскладов игральных карт: по нескольку вариантов русских, французских, итальянских, цыганских систем. Все они порождают расплывчатые тексты-предсказания, сформулированные в абстрактных терминах: таких как «казённый дом», «пустые хлопоты», «неблагоприятное известие» и тому подобные. Эти абстрактные термины можно представить как ситуации, соединенные базовой сюжетной линией: $A = \{a_1, \ldots, a_i, \ldots, a_n\}$, где A – сюжет, а a_i - ситуации. В рамках проекта СИГМА создан тезаурус такого рода терминов, к которым итеративно применяются операции развёртывания и конкретизации.

Анализируя реальные тексты, имеющие литературное значение, можно придти к существованию ещё одного плана, в дополнение к плану выражения и плану содержания. Назовём его планом изложения, или планом повествования. Именно благодаря наличию этого плана структура художественного текста превращается в систему. По мере развития текста этот план заставляет читателя пересматривать первичный способ интерпретации уже прочитанного фрагмента текста.

Введём понятие о нескольких временных атрибутах текста. Линейное время есть время разворачивания его при прочтении, это просто введение линейного порядка на последовательности элементов текста. Линейное есть время синтаксическое. Оно определяется синтаксической структурой фразы. Под синтаксической структурой фраза понимается заданное на множестве слов фразы ориентированное дерево синтаксических зависимостей, отражающее актуальное членение переданного фразой сообщения.

Следующий временной атрибут – время, в котором существует сюжет текста. Этот вид времени едва ли может существовать вне пространственного континуума. В силу этого будем говорить о хронотопе текста. Хронотопическое время может изменяться внутри текста – в этом случае как следствие наступает необходимость смены социокультурного дискурса текста. При этом сюжет A развивается в хронотопическом временном пространстве, тогда как и ситуации a_i также могут развиваться, как во временном пространстве сюжета A, так и в своем собственном временном пространстве, которое будет внутренним по отношению к хронотопическому временному пространству всего сюжета A. Хронотопическое время T сюжета A является базовым временем на котором необходимо отложить временные отрезки ситуаций a_i, т.е. $T^i = t_0^i + \Delta t$, где t_0^i - время начала a_i, а Δt характеризует продолжительность всей ситуации. При этом внутреннее время повествования T^i определено планом изложения. Оно может как течь в «естественном» направлении, так и совершать «скачки» как вперёд, так и в обратном направлении, поэтому основной задачей является динамическое определение значения Δt. Главное, что отличает внутреннее время от времени хронотопического, это то, что при «скачках» внутреннего времени не обязательна смена социокультурного антуража. Это время является мощным средством передачи авторского замысла.

Связь предложений внутри текста может быть простой синтаксической связью соположенности. Например, два предложения, связанные через анафорическую ссылку, являются именно таковыми, то есть грамматически и текстуально соположенными. Крайним случаем такого вида связи может наблюдаться тогда, когда два (или более) рядом расположенных предложения могут быть без потери семантических свойств преобразованы в одно предложение. Рассмотрим следующий классический пример: два предложения «Девочка лежала на траве» и «Она читала книгу» с синтаксической и семантической точек зрения полностью эквивалентны предложению «Девочка лежала на траве и читала книгу». Тем не менее, с точки зрения прагматической (в данном случае с точки зрения стиля автора) это преобразование не сохраняет эквивалентность.

Для порождения текста как реализации моделей Актантов необходимо большое число баз знаний и баз данных. Здесь предполагается, что база данных фиксирует отношения между объектами, а база знаний – между классами понятий.

Наиболее трудным этапом является этап порождения текста. При работе над ним авторы руководствовались идеями, изложенными в фундаментальных работах И.А. Мельчука «An Explanatory Combinatorial Dictionary of Modern Russian» и «Толково-комбинаторный словарь русского языка» [4]. В этих работах И.А. Мельчук предлагает создать словарь, в котором единица вхождения – слово – сопровождалась бы информацией о возможных синтаксических, семантических и функциональных связей данной словарной единицы. Всего предлагается набор такого окружения, определяющий 61 связь. Толково-комбинаторный словарь даёт неплохое основание для составления предложений и их парафразирования.

Более крупными, после слов и устойчивых словосочетаний, элементами текста являются фразовые комплексы. Введение в рассмотрение такого понятия обусловлена необходимостью определения минимальных структурах, описывающих замкнутый фрагмент ситуации: Актанта, действия, обстановки и т.п. Границы фразового комплекса могут полностью совпадать с границами синтаксического предложения. Однако фразовый комплекс может быть как собственной частью предложения, так и распространяться на несколько предложений.

Предварительный анализ функциональных ролей фраз позволил составить следующий их перечень:

– фраза-описание антуража;

– фраза-описание персонажа;

– фраза-описание поведения персонажа;

– нейтральная фраза;

– отвлекающая фраза (внешне похожа на нейтральную фразу, но позднее выявляется её подлинная роль в плане повествования).

Определены базы знаний необходимые при работе с системой СИГМА. Это было сделано исходя из основных этапов её работы, к которым относятся:

– интерпретация метаязыкового представления в собственное внутреннее представление (в ряде случаев метаязыковое и внутреннее представления могут совпадать);

– определение участвующих во фрагменте персонажей и определение антуража с учетом дискурса и его хронотопических характеристик;

– генерирование предложения естественного целевого языка для передачи смысла;

– ввод анафорических указателей для приближения сгенерированного фрагмента к «обычному» тексту;

– корректировка дискурса;

– вывод фрагмента для редактирования (данный этап реализуется при осуществлении процедуры редактирования человеком);

– после выполнения процедуры редактирования встроить полученный фрагмент в текст более высокого уровня.

В СИГМА предусмотрено наличие разноуровневых баз данных и баз знаний.

В базах данных, фиксируются все особенности дискурса: как ограничения, так и реализованные возможности. Эти базы данных фиксирует путь, по которому текст реализован в дереве возможностей. Как ограничения, так и возможности могут быть как постоянные, так и временные.

Создание баз знаний, прежде всего, зависит от хронотопических ограничений, налагаемых на генерируемый текст. Ведение подобных баз знаний необходимо осуществлять по хронотопам, которые включены в систему. Сюда также должны быть включены характерные реалии и понятия, относящиеся к данной географической точке в данное историческое время. Кроме того, здесь же специфицируются атрибуты включённых реалий. Очевидно, в хронотопе «Франция XVII века» дальней считалась дорога длиной в несколько десятков лье, а казённым домом можно было назвать Бастилию или Лувр. В современной же Франции эти характеристики иные: дальняя дорога может уже представлять собой трансатлантический перелёт, Бастилии больше не существует, а в бывшем королевском дворце находится всемирно известный музей.

Уровнем выше хронотопических баз знаний лежит база знаний по онтологии мыслимых миров. В ней эксплицируется неявная семантическая связь между лексическими единицами (словами и устойчивыми словосочетаниями) и смыслами, которые они выражают. Как было сказано выше, основной инструментарий для создания такой базы знаний базируется на концепциях И.А. Мельчука [4]. При этом необходимо учитывать не только внешнюю семантику, но и семантическую структура лексических единиц.

Таким образом, проект СИГМА предоставляет возможность многоуровневой иерархической организации знаний. Идеи проекта и способы их решения позволяют использовать, полученные результаты для решения таких подчиненных вопросов, как: создание интеллектуально-лингвистических программ, информационно-поисковых машин, декларирующих обработку запросов на естественном языке, средств фиксации и платформенно- и регионально- независимого распространения знаний.

Список литературы.

1. Якимов В. Н., Дьяконов Г. Н., Машков А. В. Формирование онтологии предметной области на основе анализа NFL-континуума // Информационные технологии. – 2006. – №3, С. 36-39.

2. Пави П. Словарь театра: Пер. с фр. – М.: Прогресс, 1991. – 504 с.: ил.

3. Словарь терминов по информатике на русском и английском языках. //Жданова Г. С., Колобродова Е. С., Полушкин В. А., Чёрный А. И . – М.: Наука, 1971. – 359 с.

4. Мельчук И. А. Русский язык в модели «Смысл – Текст». – Москва – Вена: Школа «Языки русской культуры», Венский славистический альманах, 1995. – XXVIII с., 682с.

Особенности применения радиально-базисных нейронных сетей для решения краевых задач математической физики[1]

Яничкина Е. В., *ynev@mail.ru*
кафедра Вычислительных систем и моделирования
Пензенского Государственного Педагогического Университета
имени В.Г. Белинского
440028, г. Пенза, ул. Г. Титова, д.8, кв.73

Нейронные сети – мощный аппарат для решения дифференциальных уравнений в частных производных (ДУЧП). Хорошо исследовано применение клеточных нейронных сетей для решения ДУЧП [1]. Способность нейронной сети хорошо аппроксиматоровать функции позволяет реализовать с помощью нейронных сетей приближенные методы решения ДУЧП, например, метод Галеркина [2].

В настоящее время большой интерес вызывают методы решения ДУЧП с применением радиально-базисных функций (RBF) [3]. Эти методы могут быть эффективно реализованы на радиально-базисных нейронных сетях (RBFNN). В [4] описаны общие принципы применения RBFNN для решения ДУЧП. Некоторые практические подходы к использованию RBFNN показаны в работах [5,6].

[1] Работа выполнена по тематическому плану научно-исследовательских работ Пензенского государственного педагогического университета, проводимых по заданию Федерального агентства по образованию в 2006 г.

Целью данной работы является анализ алгоритмов решения ДУЧП на RBFNN как типовых процедур работы сети.

Радиально-базисная нейронная сеть – это сеть с двухслойной структурой, первый слой выполняет нелинейное отображение, реализуемое нейронами с базисными радиальными функциями, выходной слой линеен.

Рассмотрим работу RBFNN при решении ДУЧП. На вход сети подаются координаты точек области (точек коллокации) $x=(x_1,x_2,\dots x_n)$, n-размерность пространства. Радиальная функция каждого нейрона характеризуется своими параметрами: центром $c^i=\left(c_1^i,c_2^i,\dots c_n^i\right)$ и шириной $a^i>0$, которые уточняются в процессе обучения. Каждый нейрон радиально-базисного слоя выполняет нелинейное преобразование $\phi^{(i)}$, аргументом которого является расстояние от точки x до соответствующего центра c^i. Роль выходного слоя сводится к взвешенному суммированию сигналов, поступающих от нейронов скрытого слоя $\sum_{i=1}^{m} w^{(i)}\phi^{(i)}(x)$. На выходе получаем значение искомой функции в точке x $u(x)=\sum_{i=1}^{m} w^{(i)}\phi^{(i)}(x)$. Обучение сети сводится к нахождению неизвестных параметров w, a, c.

Для создания оптимальной сети можно использовать различные виды RBF, наиболее часто используемыми являются мультиквадрик (MQ) $\phi=\left(r^2+a^2\right)^{n/2}$, $n=-1,1,2,3,\dots$, и гауссиан $\phi=\exp\left(-r^2/a^2\right)$ где a – определенная пользователем константа.

Продемонстрируем работу сети на примере решения двумерного уравнения Пуассона

$$\Delta u=p(x),\quad x\in\Omega, \tag{1}$$

$$u=p_1(x),\ x\in\partial\Omega_1 \qquad n\times\nabla u=p_2(x),\ x\in\partial\Omega_2, \tag{2}$$

где n – единичная внешняя нормаль; ∇ – оператор градиента; $\partial\Omega_1$ и $\partial\Omega_2$ –границы области, такие, что $\partial\Omega_1\cup\partial\Omega_2=\partial\Omega$ и $\partial\Omega_1\cap\partial\Omega_2=\varnothing$; p_1 и p_2 – известные функции x.

Выбирая в качестве радиально-базисной функции MQ, определяемую как $\phi^{(i)}\left(\left\|x-c^{(i)}\right\|\right)=\sqrt{r^2+a^{(i)2}}$, рассматривают RBF-сеть как аппроксиматор функции:

$$u(x)=\sum_{i=1}^{m} w^{(i)}\phi^{(i)}(x)=\sum_{i=1}^{m} w^{(i)}\sqrt{r^2+a^{(i)2}}, \tag{3}$$

где m – число радиально-базисных функций (скрытых нейронов).

Радиально-базисная сеть аппроксимирует производные функции $u(x)$. Из (3) производные функции $u(x)$ рассчитываются следующим образом

$$u_{j\dots l}(x)=\frac{\partial^k u}{\partial x_j\dots\partial x_l}=\sum_{i=1}^{m} w^{(i)}\frac{\partial^k\phi^{(i)}}{\partial x_j\dots\partial x_l},$$

$$u_j(x)=\sum_{i=1}^{m} w^{(i)}\frac{\left(x_j-c_j^{(i)}\right)}{\left(r^2+a^{(i)2}\right)^{0.5}},\qquad u_{jj}(x)=\sum_{i=1}^{m} w^{(i)}\frac{r^2+a^{(i)2}-\left(x_j-c_j^{(i)}\right)^2}{\left(r^2+a^{(i)2}\right)^{1.5}}. \tag{4}$$

Отсюда $\Delta u=u_{11}(x)+u_{22}(x)=\sum_{i=1}^{m} w^{(i)}\frac{r^2+2a^{(i)2}}{\left(r^2+a^{(i)2}\right)^{1.5}}$.

Для нахождения u, определяем функционал ошибки как сумму квадратов невязок, получаемых при подстановке u (3) и производных (4) в уравнение (1) и в граничные условия (2)

$$I(w,c,a)=\sum_{i=1}^{n}\left[u_{11}\left(x^{(i)}\right)+u_{22}\left(x^{(i)}\right)-p\left(x^{(i)}\right)\right]^2+$$
$$+\sum_{j=1}^{k}\left[u\left(x^{(j)}\right)-p_1\left(x^{(j)}\right)\right]^2+\sum_{j=1}^{k}\left[n_1u_1\left(x^{(j)}\right)+n_2u_2\left(x^{(j)}\right)-p_2\left(x^{(j)}\right)\right]^2, \tag{5}$$

где $w=\left(w^{(1)}, w^{(2)},..., w^{(m)}\right)$, $c=\left(c^{(1)}, c^{(2)},..., c^{(m)}\right)$, $a=\left(a^{(1)}, a^{(2)},..., a^{(m)}\right)$ и

$x^{(i)} \in \Omega, \quad x^{(j)} \in \partial\Omega$, $i=1,2,...,n, \quad j=1,2,...,k$ – некоторые фиксированные дискретные точки.

Для обучения сети используем градиентный алгоритм обучения [6], минимизирующий функционал I путем настройки весов $w^{(i)}$, центров $c^{(i)}$ и ширины $a^{(i)}$:

$$w_t^{(i)} = w_{t-1}^{(i)} - \eta \frac{\partial I\left(c_{t-1}^{(i)}, a_{t-1}^{(i)}, w_{t-1}^{(i)}\right)}{\partial w_{t-1}^{(i)}}, \qquad c_t^{(i)} = c_{t-1}^{(i)} - \beta \frac{\partial I\left(c_{t-1}^{(i)}, a_{t-1}^{(i)}, w_t^{(i)}\right)}{\partial c_{t-1}^{(i)}}, \quad a_t^{(i)} = a_{t-1}^{(i)} - \alpha \frac{\partial I\left(c_t^{(i)}, a_{t-1}^{(i)}, w_t^{(i)}\right)}{\partial a_{t-1}^{(i)}},$$

где $\eta, \quad \beta, \quad \alpha$ – скорости (коэффициенты) обучения. Процесс уточнения параметров продолжается до достижения определенной погрешности.

Рассмотренный алгоритм экспериментально исследовался при решении следующей задачи:

$$\Delta u = \sin\left(\pi x_1\right) \cdot \sin\left(\pi x_2\right), \quad x \in \Omega$$

$$u = 0, \quad x \in \partial\Omega \qquad \Omega: \; x_1 \in [0;3] \; x_2 \in [0;3]$$

Для решения использовалась нейронная сеть, состоящая из 25 нейронов, с функциями активации типа MQ. Обучение сети происходило с оптимизацией всех трех параметров (весов, ширины, центров), контроль решения производится в 100 точках равномерно расположенных на всей области по сетке 10×10, с шагом 0,3. Коэффициенты обучения для весов порядка 10^{-4}, для ширины и центров RBF порядка 10^{-2}.

В процессе решения находится значение функции в некотором наборе точек, подаваемых на вход нейронной сети (рис.3). После обучения возможно найти решение в любой точке области, подставив ее координаты в выражение (3), на рис. 1 значения функции вычислены в 900 точках.

При решении дифференциальных уравнений в частных производных возникает вопрос о расстановке c_k – центров RBF. Вариант корректировки расположения центров во время обучения сети дает лучшие результаты.

Вопрос о количестве нейронов можно решать экспериментально, но лучше производить постепенное наращивание сети (неоднократно вставляя узел в той точке области, компонент функционала ошибки которой наибольший).

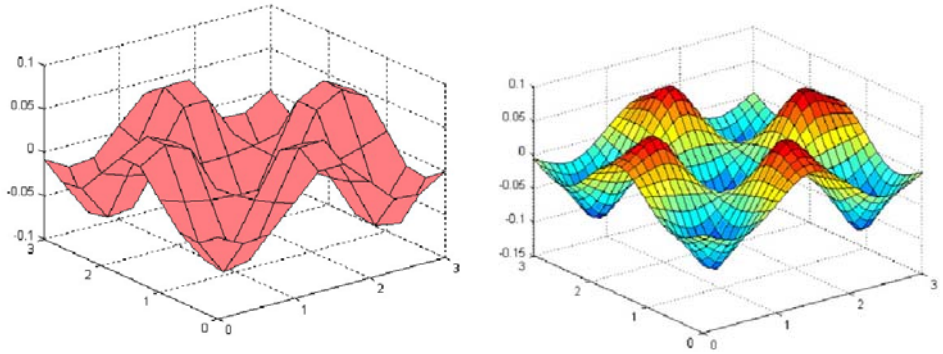


Рис.1 Значение искомой функции с шагом 0,3 и с шагом 0,1

Лучше использовать RBF-сети с различными типами функций активации нейронов. Следует указать отсутствие математической теории обеспечивающей выбор типа RBF при добавлении нового узла. Надлежащий выбор таких функций требует некоторого навыка и опыта пользователя. При этом следует в максимальной степени учитывать особенности данной задачи, моделирующей конкретный физический процесс.

Достоинством применения радиально-базисных нейронных сетей является возможность эффективной реализации не только на специальных параллельных вычислительных структурах, но и на компьютерах традиционной архитектуры.

Список литературы

1. Горбаченко В. И. Нейрокомпьютеры в решении краевых задач теории поля. – М.: Радиотехника, 2003. – 336 с.

2. Флетчер К. Численные методы на основе метода Галёркина. – М.: Мир, 1988. – 352 с.

3. Kansa E.J. Motivation for using radial basis functions to solve PDEs // http://uahtitan.uah.edu/kansaweb.html

4. Васильев А. Н., Тархов Д. А. Новые подходы на основе RBF-сетей к решению краевых задач для уравнения Лапласа на плоскости // Нейрокомпьютеры: разработка, применение. – 2004. – № 7-8. – С. 119 – 126.

5. Nam Mai-Duy, Thanh Tran-Cong. Numerical solution of differential equations using multiquadric radial basis function networks // Neural Networks 14 (2001) 185-199

6. Numerical solution of elliptic partial differential equation using radial basis function neural networks / Li Jianyu, Luo Siwei, Qi Yingjiana, Huang Yapinga // Neural Networks. – 2003. – 16(5/6. – P. 729 – 734.

Формирование и оптимизация базы данных и знаний для диагностики состояний коммуникативного стресса[2]

Янковская А. Е.,
профессор Томского государственного архитектурно-строительного университета,
634003, г. Томск, пл. Соляная, д. 2;
e-mail: yank@tsuab.ru
Рождественская Е. А.,
доцент Томского государственного университета,
634050, г. Томск, ул. Ленина, 36
Муратова Е. А.,
доцент Томского политехнического университета,
634050, г. Томск, ул. Ленина, 30;
e-mail: muratova@tpu.ru

Введение

Проблема ранней диагностики коммуникативного стресса (К-стресса) в учебном процессе является одной из актуальных сфер исследования в социально-психологической науке и практике [1-3]. В исследованиях В.И. Кабрина [4] коммуникативный стресс (частный случай социально-психологического стресса) рассмотрен в качестве сложного феномена, выступающего в роли механизма транскоммуникативного развития личности в процессе социальной адаптации, который выражается в виде эмоционально-отрицательного состояния личности на различных этапах коммуникативного процесса.

Целью проводимых авторами исследований является выявление личностной предрасположенности к переживанию К-стресса в учебно-коммуникативных ситуациях, как у студентов, так и преподавателей. Ранняя диагностика состояний К-стресса, своевременная психопрофилактика и коррекция негативных эмоциональных состояний позволит снизить риск возникновения психологических и психосоматических заболеваний, а также повысить продуктивность деятельности и степень умственной работоспособности индивида.

Выявление различного рода закономерностей в возникновении К-стресса предлагается осуществлять в интеллектуальной информационной системе (ИИС) ДИАКОР-КС (**ДИА**гностика и **КОР**рекция **К**оммуникативного **С**тресса), сконструированной на основе интеллектуального инструментального средства ИМСЛОГ [5], базирующегося на оригинальном матричном представлении данных и знаний [6], логико-комбинаторных тестовых методах распознавания образов [6] и методах графической визуализации с использованием когнитивной графики [6]. Учитывая сложную природу объекта исследования, этап формирования и оптимизации базы данных и знаний является наиболее важным, поскольку формирование признакового пространства и способ представления данных и знаний существенно влияют не только на возможности выявления тех или иных закономерностей и результаты принятия решений, но и на используемые методы их обработки.

[2] Работа выполнена при финансовой поддержке РГНФ (проект № 06-06-12603в) и частично РФФИ (проект № 04-01-00144).

В настоящей статье описываются факторы, обуславливающие возникновение К-стресса, формируемое признаковое пространство, способ представления данных и знаний в ИИС ДИАКОР-КС, способ оптимизации базы данных и знаний, приводятся предварительные результаты формирования и оптимизации базы данных и знаний для диагностики состояний коммуникативного стресса.

Факторы, обуславливающие возникновение К-стресса.
Формирование признакового пространства

При исследовании К-стресса необходимо учитывать роль внешних и внутренних стресс-факторов, проявляющих свое действие в учебно-коммуникативном процессе, их степень активности и продолжительность влияния.

Внешние стресс-факторы представляются объективно фиксируемыми условиями, способствующими возникновению и развитию К-стресса. Для студентов возможны следующие стрессовые ситуации: вынужденное общение с незнакомыми людьми, выступления перед незнакомой аудиторией, ответы на семинарах и экзаменах, большой объем информационной нагрузки, изложение теоретического материала с изобилием научных терминов, авторитарный стиль общения преподавателя, конфликты с преподавателями, конфликты в учебной группе, статусная позиция студента в группе. Для преподавателя, выступающего в роли коммуникативного партнера студента, стрессовыми ситуациями могут быть: неудовлетворенность ответами на семинарах и экзаменах, несовпадение по темпу общения, отсутствие взаимопонимания со студентами, конфликты с коллегами, неудовлетворенность своей работой, возрастание учебной нагрузки, неудовлетворенность оплатой труда.

Внутренние стресс-факторы представляются индивидуально-личностными особенностями человека: личностная тревожность, ригидность, фрустрированность, напряженность, неадекватная самооценка, особенности характерологических черт и темперамента, агрессивность, эмоциональная неустойчивость.

Для изучения закономерностей возникновения К-стресса в учебном процессе у студентов и преподавателей в ИИС ДИАКОР-КС необходимо создать обучающую выборку и представить ее в виде, используемом в системе. Для этого авторами сформировано признаковое пространство, условно разбитое на 5 групп.

В **группу A** включены признаки, соответствующие различным внешним стресс-факторам, оценка которых осуществляется путем анкетирования испытуемых.

В **группу B** включены признаки, диагностируемые с использованием психологических тестов, представляющие собой внутренние личностные факторы:

– уровень тревожности, определяемый по методикам: шкала реактивной-личностной тревожности Ч.Д. Спилбергера - Ю.Л. Ханина, опросник Дж. Тейлора; опросник состояния нервно-психической напряженности Т.А.Немчина;

– уровень ригидности – цветовой тест М. Люшера, Томский опросник ригидности (ТОР);

– тип акцентуации характера – опросник Х. Шмишека, характерологический диагностический опросник (ХДО);

– уровень невротизации и психопатизации – УНП-опросник;

– степень фрустрированности и уровень стрессоустойчивости – тест рисуночной фрустрации С. Розенцвейга.

Использование нескольких методик для определения личностных характеристик позволит повысить достоверность итогового значения, определяемого по специальной процедуре голосования.

В **группу C** включены признаки, определяемые по данным анкетирования и психологического тестирования, характеризующие уровень адаптации к окружающим людям и к окружающему миру, уровень продуктивности деятельности и степень умственной работоспособности. На основе признаков данной группы будет диагностирован К-стресс и возможная его выраженность: отсутствует, слабо выражен, средне выражен, сильно выражен, очень сильно выражен.

В **группу D** включены признаки, формируемые по данным медицинского осмотра (профосмотра), результатам психологического тестирования, интервьюирования и анкетирования, характеризующие

риски возникновения сердечно-сосудистых, желудочно-кишечных, аллергических заболеваний, бронхиальной астмой, гормональных расстройств и пограничных нервно-психических расстройств.

В **группу E** включены признаки, отражающие различные мероприятия по психопрофилактике и коррекции выявленного состояния К-стресса, а именно, необходимость проведения психопрофилактических, психогигиенических, психо-коррекционных, медикаментозных и психотерапевтических мероприятий.

Представление данных и знаний в ИИС ДИАКОР-КС

Для представления обучающей выборки и последующего поиска закономерностей в данных и знаниях и принятия решений в ИИС ДИАКОР-КС используется матричная модель [5,6], включающая матрицу описаний (Q), матрицу различений (R) и матрицу переходов (P).

Строкам матрицы Q сопоставляются обучающие объекты (испытуемые), столбцам – характеристические признаки из групп A и B. Элементы *i*-й строки матрицы Q задают значения характеристических признаков для *i*-го объекта.

Строки матрицы R сопоставляются одноименным строкам матрицы Q, столбцы – классификационным признакам из групп C и D. Множество всех неповторяющихся строк матрицы R соответствует множеству выделяемых образов (состояний К-стресса). С содержательной точки зрения матрица различений R может представлять решения различного рода. Матрица R1 является матрицей диагностического типа, где первый столбец представлен интегральным механизмом классификации, формируемым последующими столбцами, которые сопоставлены признакам из групп C и D. Матрица R2 задает последовательность мероприятий, которые необходимо выполнить для исследуемого объекта (испытуемого) в данной ситуации (признаки группы E). Матрица R3, представляющая независимые механизмы классификации (например, мнения различных экспертов) в ИИС ДИАКОР-КС, не используется.

Строкам матрицы P взаимно однозначно сопоставляются строки матрицы Q, столбцам – интервалы (моменты) времени или управляющие воздействия, которые, в частности, могут задаваться строками матрицы R2. Элемент p_{ij} матрицы P задает состояние, в которое должен переходить *i*-й объект матрицы Q на *j*-м интервале (в *j*-й момент) времени или в результате *j*-го управляющего воздействия. Иначе говоря, матрица P отражает динамику исследуемого состояния.

Отметим, что данная модель позволяет представлять не только фактические данные, но и знания экспертов, поскольку одной строкой матрицы Q можно задавать в интервальной форме подмножество объектов, для которых характерно одно и то же итоговое решение, формируемое соответствующими строками матриц R1, R2 и R3.

Поскольку ИИС ДИАКОР-КС основано на тестовых методах распознавания образов, то описание объекта должно быть представлено в виде бинарных или *k*-значных признаков (*k*=3,4,…), причем, значения части из них могут быть либо неизвестны, либо заданы с некоторой степенью уверенности (вероятностью), либо в некотором диапазоне (парой вероятностей).

Для унификации признакового пространства используется эффективный алгоритм адаптивного кодирования разнотипной информации [7], позволяющий определить оптимальное конечное число информативных интервалов и сформировать новое признаковое пространство, используемое в ИИС ДИАКОР-КС.

Оптимизация базы данных и знаний

Реализованные в ИИС ДИАКОР-КС логико-комбинаторные и генетические алгоритмы позволят выявить закономерности по констатации состояния К-стресса, тем самым оптимизировать выбор характеристических признаков путем учета выявленных закономерностей. Под закономерностями в данных и знаниях понимаются подмножества признаков с определенными легко интерпретируемыми свойствами, влияющими на различимость объектов из разных образов, устойчиво наблюдаемыми для объектов из обучающей выборки и проявляющимися на других объектах той же природы, а также весовые коэффициенты признаков, характеризующие их индивидуальный вклад в различимость объектов из разных образов. К упомянутым подмножествам относятся константные, устойчивые, неинформативные, альтернативные, зависимые, несущественные, обязательные, псевдообязательные признаки, минимальные и безызбыточные логические тесты [6].

Оптимизация базы данных и знаний позволит при минимально достаточной информации об испытуемом принимать достоверные диагностические, организационно-управленческие и психотерапевтические решения.

Заключение

В соответствии с выделенными стресс-факторами, оказывающими влияние на возникновение коммуникативного стресса в учебном процессе у студентов и преподавателей, сформировано признаковое пространство, включающее 77 характеристических и 16 классификационных признаков, позволяющих диагностировать состояние К-стресса. Обучающая выборка включает результаты исследования 169 испытуемых. В результате оптимизации базы данных и знаний для выделенных 123 образов (различных состояний К-стресса) выявлены следующие закономерности: константных признаков – 8; неинформативных признаков – 25; групп альтернативных признаков (по два признака в каждой группе) – 3; зависимых признаков – 0; минимальных различающих подмножеств признаков (минимальных тестов) – 1; мощность минимального различающего подмножества признаков – 41.

Полученная информация будет использована для последующего уточнения признакового пространства при диагностировании различных состояний К-стресса в учебно-коммуникативных ситуациях у студентов и преподавателей.

Список литературы

1. Бодров В. А. Психологический стресс: развитие учения и современное состояние проблемы. – М.: Изд-во ИП РАН, 1995. -136 с.

2. Рождественская Е. А. Особенности проявления тревожности и ригидности в актуальном состоянии преподавателей, переживающих коммуникативный стресс// Вестник Томского государственного университета. -1999. - № 268. – С. 120-122.

3. Коврова М. В. Исследование коммуникативного стресса у студентов в условиях процесса обучения в высшей школе. Дисс. на соискание уч. ст. канд. псих. наук. – Кострома, 2002. – 234 с.

4. Кабрин В. И. Психология коммуникативного развития человека как личности. Дисс. на соискание уч. ст. докт. псих. наук. – Санкт-Петербург, 1993. – С. 382-412.

5. Аметов Р. В., Гедике А. И., Янковская А. Е. Интеллектуальное инструментальное средство ИМСЛОГ (версия 2004 года)// IX Национальная конференция по искусственному интеллекту с международным участием (КИИ-2004). Сб. научных трудов. Том 2. – Москва: Физматлит, 2004. – С. 582-590.

6. Янковская А. Е. Логические тесты и средства когнитивной графики в интеллектуальной системе// Новые информационный технологии в исследовании дискретных структур. Доклады 3-ей Всероссийской конференции с международным участием. – Томск: Изд-во СО РАН, 2000. – С. 163-168.

7. Берестнева О. Г., Муратова А. Е., Янковская А. Е. Эффективный алгоритм адаптивного кодирования разнотипной информации// Искусственный интеллект в XXI веке. Труды Международного конгресса. Том 1. – М.: Физматлит, 2001. – С. 155-166.

Интеллект и ЭВМ

Яппаров Руслан Вагизович

Владимирский государственный университет,
факультет информатики и прикладной математики.
600021 г. Владимир ул. Мира 4б кв. 103.
E-mail: molevich@mail.ru

С самых первых попыток наделить машину разумом исследователи разбились на 2 лагеря – сторонников «восходящего» и «нисходящего» подходов. Приверженцы первого считали, что единственно верный путь для достижения цели – это строительство нейронных сетей мозга, а сторонники нисходящего подхода считают, что работу мозга следует имитировать с помощью сложных программ.

Мыслительные процессы, которые должны выполнять электронные системы крайне сложны и малоизученны. Девизом же исследователей ИИ является фраза агностиков «что известно, то не интересно».

Все процессы, связанные и с мышлением компьютера с человеческим интеллектом опираются на обратную связь.

Технологии электронных машин продвинулись настолько далеко, что из-за низкой компьютерной культуры многие люди готовы очеловечить машины и принести в жертву виртуальному миру свое внимание, эмоции и здоровье.

В настоящее время стала актуальной проблема создания искусственного интеллекта. Это весьма обширная проблема. Сейчас очень стремительно развивается новая отрасль психологии, изучающая взаимоотношения между человеческим разумом и компьютером. Скоро наступят времена, когда по важности она не будет уступать проблеме взаимоотношения человека с человеком или даже человека с обществом.

Человек имеет многолетний опыт работы с компьютерными системами – имитаторами работы человеческого мозга может задать множество вопросов. А разум, умеющий грамотно задавать вопросы, уже является искусственным интеллектом.

Машина, обладающая достаточным объемом фактических данных из истории, могла бы, например, разработать более эффективные экономические теории, чем те явно несовершенные концепции, которыми ныне руководствуется правительство. Вообще говоря, цель создания искусственного интеллекта – снизить интеллектуальную нагрузку, лежащую на современном человеке. Однако первые же исследователи, работающие в области ИИ столкнулись с весьма запутанными проблемами, далеко выходящими за пределы традиционной информатики. Оказалось, что прежде всего необходимо понять механизмы процесса обучения, природу языка и чувственного восприятия. Выяснилось, что для создания машин, имитирующих работу человеческого мозга, требуется разобраться в том, как действуют миллиарды его взаимосвязанных нейронов, а самой трудной проблемой, стоящей перед современной наукой, является познание процессов функционирования человеческого разума, а не просто имитация его работы. Ученным даже трудно придти к единой точке зрения относительно самого объекта их исследования – интеллекта. Некоторые считают, что интеллект – это умение решать сложные задачи, другие рассматривают его как способность к обучению, обобщениям и аналогиям; третьи – как возможность взаимодействия с внешним миром путем общения, восприятия и осознания воспринятого посредством обратной связи. Но, несмотря на эту разность подходов, все исследователи сходны во мнении: не надо давать компьютеру типовой набор задачек и считать сумму баллов. Это годится только для человека. Лучший тест для компьютера, в результате которого можно будет придти к выводу, что он разумный – это если компьютер заставит нас поверить, что мы имеем дело не с машиной, а с человеком. Безусловно, довести машину до такого уровня чрезвычайно трудно, но слишком привлекательна идея создания мыслящих машин «человеческого типа». Эта идея уходит корнями в далекое прошлое. Еще древние египтяне и римляне испытывали благоговейный ужас перед культовыми статуями, которые жестикулировали и изрекали пророчества (разумеется с помощью жрецов). А средневековые летописи полны рассказов об автоматах, способных говорить и двигаться почти так же как и их хозяева – люди.

Много лет спустя первые программисты, развлекаясь, составляли программы для сочинения музыки, решения головоломок, игр, шахмат и т. д. Некоторые, романтично настроенные программисты даже заставляли свои машины писать любовные письма. Они предполагали, что, преодолев период «электронного детства» и обучившись в библиотеках всего мира, хитроумные компьютеры благодаря быстродействию, точности и безотказности памяти, постепенно превзойдут своих создателей – людей. Однако находились и пессимисты, которые считали, что истинный разум невозможно отделить от его психологической основы, заключенной в человеческом организме, «цифровой компьютер не человек, у компьютера нет ни тела, ни эмоций, ни потребностей. Он лишен социальной ориентации, которая приобретается жизнью в обществе, а именно она делает поведение разумным». Создатели же систем искусственного интеллекта парируют, говоря о принципе обратной связи. Принцип обратной связи заключается в использовании информации, поступающей из окружающего мира для изменения поведения машины (подобным образом работает обычная термопара, регулирующая зимой температуру в нашем доме). Принцип обратной связи прост и одновременно позволяет объяснить многие закономерности, так что на его основе строятся теории машинного и человеческого разума. Еще в 1948 г. Винер доказал, что именно благодаря обратной связи все живое приспосабливается к окружающей среде и добивается своей цели. «Все машины, претендующие на разумность, – писал он, – должны обладать способностью преследовать определенные цели и приспосабливаться, т. е. обучаться».

Способность к обучению как критерий интеллекта так же высоко ценится и в обществе людей. В эту сферу ежегодно вкладываются миллиарды долларов.

Однако благодаря той же обратной связи, появляются отрицательные эффекты – в процессе обучения отношения человека с компьютером становятся настолько тесными, что реальный мир заменяется ложным, виртуальным. Скорее всего это связано с недоразвитостью технической культуры человека, технологии двоичной логики ушли так далеко по сравнению с темпами эволюции человека, что ему приходится сутками засиживаться пред голубым монитором, получая объем информации, созданный за последние 20 лет.

Безусловно, то что создано сейчас в области компьютерных технологий, невозможно полностью охватить даже значительному коллективу профессионалов.

Однако дымка надежд и интереса обволакивает и дурманит разум миллиардов молодых людей. Они проводят большую часть дня во всемирной глобальной сети Интернет. Они хотят стать хакерами, заработать, найти вторую половину, просто хорошо провести время и т.д. В общем они повышают свой духовный уровень. Все это, как они утверждают, дают «умные машины» и их мир. То есть люди уходят своим физическим телом, разрушая его, в виртуальный мир, говоря, что он лучше реального. Впервые эта тенденция захвата ЭВМ своей крупной ниши в человеческом обществе проявилась в середине 60-х годов. В это время была создана первая диалоговая программа «Элиза», имевшая бешенную популярность. Программа была построена как универсальная система, способная поддерживать диалог, на английском языке на любую, заложенную в нее тему. Наибольшую известность получил вариант «Доктор». В этом обличии программа Элиза имитировала поведение врача-психотерапевта при первичном опросе пациента. Многие психотерапевты сами стараются говорить как можно меньше, ограничиваясь лишь краткими вопросами и общими наблюдениями, что позволяет пациенту самому рассказывать о своих чувствах, поэтому в Элизе были заложены минимальные познания – информация, необходимую для построения замечаний, имитирующих врача, программа извлекала из ответов пациента.

Вот пример диалога между пациенткой и Элизой:
П. Все мужчины одинаковы!
Э. В каком смысле?
П. Они всегда пристают к тебе с чем-нибудь!
Э. Вы имеете в виду что-нибудь конкретное?
П. Ну да, это мой парень заставил меня прийти сюда
Э. Ваш парень заставил вас прийти сюда?
П. Он говорит, что я почти все время подавлена.
Э. Прискорбно слышать, что вы все время подавлены.
П. Это правда, я несчастна.
Э. Вы думаете, что ваш приход сюда поможет вам не быть несчастной?
П. Это точно, мне нужна чья-нибудь помощь.
Э. Что значит для вас получить помощь?
П. Может быть я научусь ладить с моей мамой.
Э. Расскажите подробнее о вашей семье.

Впечатление, что Элиза понимает пациента в значительной степени иллюзорно, программа была составлена так, чтобы распознать ключевые слова и сочетания в набираемом на клавиатуре тексте. Например если пациент упоминал мать, Элиза отвечала вопросом о семье. Вообще, Элиза понимала речь в самом примитивном смысле – не больше домашней собаки, которая определенным образом реагирует на команды «обед», «гулять» и т. д. Однако ответы Элизы выглядели столь завораживающими и правдоподобными, что многих приводили в восторг. Все это привело создателя Элизы в изумление: «Я был поражен, увидев насколько быстро и глубоко люди вступают в эмоциональный контакт с компьютером, с какой готовностью они очеловечивают его».

Но еще более обеспокоило автора то, что некоторые психиатры стали рассматривать Элизу как многообещающее средство для преодоления дефицита врачей. Посыпавшиеся предложения повергли программиста в ужас (он-то как создатель понимал что к чему). Он охладел и даже стал ярым противником всяческих исследований в области ИИ.

Думаю, в обозримом будущем компьютеры скорее всего останутся тем, чем всегда были для человека машины – помощниками в конкретных делах. Еще по крайней мере 30 лет обобщать, мечтать, планировать, ставить и решать проблемы, писать стихи будут по прежнему люди с их несравненными биологическими мозгами.

Оглавление

Предисловие 1

Donchenko V. S. Fuzzy sets in the plural model of uncertainties: statistical interpretation 3

Gavrilova T., Puuronen S. Ontologies for computer science research 6

Liu Yong . Development of an ITS for self-instructing of Chinese hieroglyph 17

Mahmudova Shafaqat Jabrayil qizi. Application recognition of the bases of images 19

Okulovskiy Yuri. A model and implementation of universal engine for neural systems 21

Shermenev A. Two-waves interaction in cylinder 25

Worawit Meanrach and Suphamit Chittayasothorn. Bitemporal Multi-level Secured Database for Intelligence Applications 26

Айдаров Ю. Р. Применение генетических алгоритмов для анализа криптографических протоколов 29

Альханагтах О. Ж. Программное обеспечение для инвестиционного бизнес-проектирования 30

Андрианов Д. Е. Методы обработки информации о пространственно распределенных объектах 39

Антонова Е. И., Рябцев Т. В., Бенгер Р. В. Модель интеллектуальной системы управления деятельностью коллективов 41

Артемьева И. Л., Рештаненко Н. В., Цветников В. А. Расширяемая интеллектуальная интернет-система по химии 44

Артюхин В. В. Экспертная система диагностики вирусного гепатита 47

Арьков П. А. Построение модели подсистемы комплексной защиты информации как системы случайных величин 49

Ахремчик О. Л. Построение таксономии «элементы» для интеллектуальных систем автоматизированного проектирования 50

Баклан А. Н. Комбинаторная модель и методы оптимизации сетей с технологией АТМ 53

Баклановский М., Корнев Д. Искусственный интеллект в Ghostgrid 54

Бандурин Н. Г., Игнатьев В. А. Комплекс программ для решения систем нелинейных интегро-дифференциальных уравнений (ИДУ), содержащих производные и интегралы дробного порядка 56

Бендерская Е. Н., Ахин М. Х. Нейросетевое обучение с подкреплением для моделирования поведения конкурентов в экономических играх 58

Бендерская Е. Н., Жукова С. В. Сравнительный анализ методов обработки выходных сигналов хаотической нейронной сети 60

Богомолов С. А. О структурной идентификации автоматов 62

Брейман А. Д., Аникин М. А. Пространство интероперабельности в одноранговых средах 64

Варфоломеев А. Г., Марахтанов А. Г. Применение многоагентных систем для моделирования процесса обучения студентов 67

Васильев В. И., Лохматов М. В. Методика оценки безопасности интеллектуального здания 70

Веснин Е. Н., Царев В. А. Оптимизация процесса обработки данных в системах распознавания буквенно-цифровых меток движущихся объектов 73

Гадиатулин Р. А. Оболочка экспертных систем XG# 1.0: Подход к автоматизированному извлечению онтологий и их применению в компоненте объяснения 79

Гай В. Е. Обработка многомерных сигналов на основе стохастических моделей вейвлет коэффициентов 81

Гайваронская Г. С., Павлов С. В. Общие подходы к классификации инфокоммуникационных услуг....84

Гедике А. И. Поиск всех безызбыточных столбцовых покрытий булевых матриц в интеллектуальных распознающих системах 89

Гращенко Л. А. Моделирование системы обеспечения информационной безопасности виртуальных интерфейсов пользователя 92

Гречкин В. А. О вычислительной сложности двукритериальной задачи инвестора 94

Грибова В. В. Система автоматического оценивания пользовательского интерфейса, управляемая базой знаний 97

Джашеева Ф. М., Мохамед-Боташева З. А. Использование агрегирования в методах нелинейной динамики для анализа и прогнозирования временных рядов 100

Дорошенко А. Н. Аппаратная поддержка процессором выполнения шифрованного кода 103

Дурденко В. А., Ильичев М. А.,. Разработка многофункциональной автоматизированной системы распознавания графических изображений 104

Ермаков В. В. Имитационное моделирование и оптимизация дискретного дозирования многокомпонентных смесей 107

Ермаков С. А., Остапенко А. Г., Ермаков А. П. Использование нечеткой математики для анализа защищенности информационных систем 108

Ермолаев Б. А., Плаксин М. А. Управление интерфейсом приложений Microsoft Office на основе объектной модели 110

Жизняков А. Л. Вейвлет-преобразование изображений с нецелым параметром сжатия 112

Жизняков А. Л., Фомин А. А., . Исследование алгоритмов измерения шума на основе вейвлет преобразования 115

Зайцев Е. И. О разработке и использовании программных агентов в распределенных интеллектуальных системах 117

Зиновьев А. А. Рост эффективности идентификации по модели Вольтерра при увеличении количества гармоник во входном сигнале 120

Иващенко М. В. Формирование экстенсионала базы знаний и выработка решений в интеллектуальных автоматизированных обучающих комплексах для подготовки операторов АСУТП..... 123

Исмагилов Н. А. Объектно-структурный подход в проектировании программных систем..... 130

Исматова Х. Р. Распознавание и интерпретация изображений в условиях априорной неопределенности 139

Карпук А. А., Шейнкман Е. С. Математические модели и алгоритмы клиринга межбанковских платежей 142

Климов В. Г., Яковлев В. И.,. Особенности информационных технологий в сфере образования российского общества..... 145

Князева М. А., Бердник А. Н., Волков Д. А., Жеравин М. В., Зотов И. Ю., Маевский М. С., Плохих С. А., Тимченко В. А. Интернет-система, моделирующая процесс преобразования программ, управляемый знаниями..... 147

Колупаева С. Н., Семенов М. Е., Вихорь Н. А. Математическая модель и комплекс программ для исследования процессов пластической деформации в ГЦК материалах 150

Королёв С. Ю. Проектирование иерархии классов для построения реалистичных изображений с использованием алгоритма трассировки луча 153

Котиков В. И. Разработка классификационного куба знаний для нового класса информационных систем – электронных информотек..... 156

Крехов Е. В., Нурматова Е. В. Нейросетевые агенты в аппаратно-программной системе контроля роторного оборудования 160

Кудрявцев А. А., Промыслов В. Г., Поскребышев И. А. Исследование характеристик периодического потока сигналов в сложных вычислительных сетях..... 163

Кузичев А. С. Программа Колмогорова и секвенциальные интеллектуальные системы 164

Кузьмин Д. М., Буслаев А. П. Основы технологии по мониторингу и анализу состояния дорог, характеристик автотранспортных потоков (АТП) и правил дорожного движения (ПДД), дисциплины водителей..... 166

Куливец С. Г. Проектирование и разработка интеллектуальной системы управления финансами предприятия с применением инструментов нечеткой логики и искусственного интеллекта..... 168

Куликова Е. Н. Информационные процессы в системах управления обучением 171

Кульчицкий К. А. Прогнозирование по трендовой модели временного ряда, не подчиняющегося нормальному распределению..... 173

Курбанов Ф. И. А-тестовая система и оценка знания с помощью метрических функций в пространстве F-чисел..... 176

Латыпов Р. Х., Крамин А., Васис О. Автоматизация информационных потоков кролиководческой фермы на базе технологии RFID..... 181

Макатов В. З., Фураева И. И. Автоматизация и распределение учебной нагрузки на кафедре..... 184

Макатов В. З., Фураева И. И. Ротация и объединение потоков в задаче автоматизированного управления рабочим учебным планом 187

Малинина К. О., Жаров А. В., Шамаев В. Г. База данных и программный комплекс электронной энциклопедии по экологии.....189

Малиновская О. Архитектура системы VITA-II для прототипирования учебных курсов on-line на основе онтологий 192

Мамедова М. Г., Джабраилова З. Г. Применение нечеткой логики в системе оценки кадрового потенциала.....195

Марценюк М. А. Матричное представление нечеткой логики 197

Мельников Б. Ф., Мельникова Е. А.,. Некоторые вопросы кластеризации ситуаций в алгоритмах реального времени 200

Мельников Б. Ф., Тарасова Е. Г. Специальные методы принятия решений для разработки экспертных систем в области микробиологии 202

Миронов А. М. Верификация функциональных программ на основе построения их графовых моделей 207

Морнев М. Л. О приближении неприрывных функций нейронными сетями.....212

Низамова Г. Ф., Ковтуненко А. С. Функциональная модель интеллектуальной системы составления расписания учебных занятий 213

Окороченко Г. Е., Бородакий В. Ю. Исследование работы системы передачи сообщений с помощью имитационной модели.....215

Орлов А. А. Выделение полосовых образов заданного профиля на цифровых изображениях.....218

Павловская А. И., Ревяко Г. М., Фисюк М. М. Компьютерная модель идентификации портрета Франциска Скорины.....221

Парамонов А. И. Об одном подходе к построению модели контекста 224

Перевалов Д. С. Использование матриц сравнений в задаче поиска по эталону.....226

Плаксин М. А. Принятие решений методом анализа иерархий: как сократить экспертоемкость? 229

Плескач В. Л., Рогушина Ю. В. Онтологический анализ как средство интеллектуализации поисковых механизмов на рынке информационных услуг.....232

Поленов М. Ю. Проблемы преобразования программных моделей для моделирующих сред 234

Пономарев Д. Ю. Исследование некоторых характеристик сетей массового обслуживания 237

Попов В. Ю., Баклановский М. В., Морнев М. Л. Язык описания нейронных сетей NNML 239

Попов С. В., Трифонова Е. Е. О логическом подходе к синтезу технологических процессов.....242

Портнов К. В. Использование модифицированных генетических алгоритмов в задаче распределения ресурсов.....244

Примаков Д. А. Концепция конструирования и реализации диагностической системы в клинической иммунологии на основе экспертных знаний 250

Пучков А. Ю., Окунев Б. В. Оптимизация поэтапного распределения средств в слабо структурированных задачах.....253

Рабчевский Е. А., Архипов Е. С. Проектирование экспертных систем технической поддержки на основе онтологий 257

Рогов А. А. Применение методов теории распознавания для анализа петроглифов Карелии 260

Рогова К. А. Информационно – поисковая система «Петроглифы Карелии» 262

Рухая Х. М., Тибуа Л. М. Об одном варианте дедуктивного построения базы знаний 264

Самигулина Г. А. Интеллектуальная система прогнозирования асимптотической устойчивости в среднем квадратическом стохастической системы управления 265

Самонов В. Е. Моделирование избирательной кампании методами нечетких игр 269

Семенов А. В., Чугайнов Е. С. Применение факторного анализа для систем биометрической идентификации 272

Скробов А. Генетические алгоритмы создания CoreWars-бойцов 275

Спиридонов. Применение мультифрактального анализа при изучении петроглифов Карелии 278

Татауров П. Дифференциальные операторы Шимуры На модулярных формах Зигеля в трехмерном случае 281

Фищенков А. Н. Особенности моделирования и оптимизация системы обслуживания цеха вулканизации 283

Фофанов В. Б. Случайное расстояние и его применения в дешифрировании изображений 286

Фофанов В. Б., Сабиров Р. Г. Байесовская сегментация изображений 288

Фофанов В. Б., Демченко А. В. . Об одном подходе к поиску зон интереса 291

Фофанов В. Б., Кулеев Р. Ф. . Сегментация зон интереса на основе случайного расстояния 293

Хлызов А. В., Чичагова М. В. Динамически настраиваемые WEB-ориентированные информационные системы 295

Худаяров Б. А., Бандурин Н. Г. Моделирование флаттера вязкоупругих пластин и цилиндрических панелей 297

Чижова И. А. Интеллектуальные системы для прогнозно-металлогенического анализа перспективных площадей 301

Шалыто А. А., Паращенко Д. А., Царев Ф. Н. Применение автоматного программирования при моделировании одного класса мультиагентных систем 305

Шарапов А. А., Коротков С. В. . GRID-агенты самоконфигурации системы обмена сообщениями 308

Швецов А. Н., Сергушичева М. А., Котов С. С., Сорокин С. И. Инструментальный программный комплекс для проектирования мультиагентных систем 309

Шестаков М. А. Заметка об организации решения задач в Интеллсист 312

Шкурко А. В. Социологические проблемы дескриптивной теории принятия решений 314

Шкурко Ю. С. Перспективы алгоритмизации процессов творческого мышления в форме искусственного интеллекта 318

Шматко А. Д., Шаповалова Н. А. Анализ бизнес - процессов компании на основе использования инструментальных средств 320

Якимов В. Н., Дьяконов Г. Н., Машков А. В. Вербальные компоненты NFL-континуума как структура и система 322

Яничкина Е. В. Особенности применения радиально-базисных нейронных сетей для решения краевых задач математической физики 326

Янковская А. Е., Рождественская Е. А., Муратова Е. А. Формирование и оптимизация базы данных и знаний для диагностики состояний коммуникативного стресса.....329

Яппаров Р. В. Интеллект и ЭВМ.....332

Оглавление 335

МАТЕРИАЛЫ IX МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ "ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И КОМПЬЮТЕРНЫЕ НАУКИ"(23-27 октября 2006 г.), том 2, часть 2. Под общей редакцией академика Садовничего В. А., проф. Кудрявцева В. Б., проф. Михалева А. В., 2006 г., 172 с.

Подписано в печать 29.09.2006.
Формат 60×90 1/16. Объем 11.5 п.л.
Заказ 18. Тираж 200 экз.

Издательство ЦПИ при механико–математическом факультете МГУ
г. Москва, Воробьевы горы.
Лицензия на издательскую деятельность ИД № 04059 от 20.02.2001 г.

Отпечатано на типографском оборудовании механико-математического факультета