

БИОНИКА ИНТЕЛЛЕКТА

ИНФОРМАЦИЯ, ЯЗЫК, ИНТЕЛЛЕКТ

№ 1 (78)

2012

НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Основан в октябре 1967 г.

Учредитель и издатель
Харьковский национальный университет радиоэлектроники

Периодичность издания — *3 раза в год*

СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И КИБЕРНЕТИКИ. ТЕОРИЯ ИНТЕЛЛЕКТА

<i>Бондаренко М.Ф., Шабанов-Кушнаренко Ю.П., Ситников Д.Э., Ситникова П.Э., Коваленко А.И.</i> Применение подстановочных операций к предикатам, представленным формулами алгебры конечных предикатов.....	3
<i>Король О.И., Шаронова Н.В.</i> Интеллектуальная обработка данных при формировании патентно-конъюнктурных баз знаний	12

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ. СИСТЕМНЫЙ АНАЛИЗ. ПРИНЯТИЕ РЕШЕНИЙ

<i>Петров Э.Г., Губаренко Е.В.</i> Цели и устойчивость социально-экономических систем при реализации концепции устойчивого развития	17
<i>Пастушенко В.Й., Матус С.К., Стеценко А.М.</i> Ідентифікація коефіцієнта вологопровідності ґрунту за допомогою нейронних мереж.....	23
<i>Гофман Е.А., Олейник А.А., Субботин С.А.</i> Синтез деревьев решений на основе теории приближенных множеств....	29
<i>Литвин О.М., Ярмош О.В.</i> Про похибку апроксимації функції двох змінних білінійними сплайнами МНК в інтегральній формі	33
<i>Литвин О.М., Першина Ю.І.</i> Відновлення об'єктів, що описуються розривними функціями, з використанням криволінійних трапецій	37

ИНТЕЛЛЕКТУАЛЬНАЯ ОБРАБОТКА ИНФОРМАЦИИ. РАСПОЗНАВАНИЕ ОБРАЗОВ

<i>Руденко О.Г., Сныткин М.С.</i> Сжатие изображений с помощью нейронной сети Fuzzy ART.....	45
<i>Ващук Ф.Г., Мельник О.О.</i> Дослідження ефективності алгоритму складання розкладів сумарного випередження і запізнення із налагодженнями, що залежать від послідовності	49
<i>Довбиш А.С., Романюк А.М., Руденко М.С.</i> Ідентифікація кадрів в задачах розпізнавання зображень медичних і біологічних об'єктів.....	53
<i>Мальков Ю.А., Радченко В.А., Таянский С.С.</i> Организация хранения дизъюнктов дедуктивных баз данных на основе реляционного подхода	59
<i>Шкловец А.В., Аксак Н.Г.</i> Определение расстояний между точками на кусочно-гладких картах Кохонена	63
<i>Бых А.И., Жемчужкина Т.В., Носова Т.В., Сухенький В.А.</i> Поиск информативных количественных показателей электромиографического сигнала. Сообщение 3	68
<i>Авраменко В.В., Волков Р.С.</i> Распознавание контурных изображений и их фрагментов при пересечении.....	72

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММНО-ТЕХНИЧЕСКИЕ КОМПЛЕКСЫ

<i>Гришко А.А., Удовенко С.Г., Чалая Л.Э.</i> Гибридные методы машинного обучения в системах управления динамическими объектами	78
<i>Дяченко В.А., Михаль О.Ф.</i> Адаптивное параллельное обучение модифицированной самоорганизующейся карты Кохонена	85
<i>Осотов И.А., Михаль О.Ф.</i> Оценка эффективности проведения локально-параллельных вычислений на многоядерных процессорах	91
<i>Бельчева А.В., Манакова Н.О.</i> Аналитический выбор исходных данных на основе метода иерархий с учетом тематики геоинформационного проекта.....	97

СТРУКТУРНАЯ, ПРИКЛАДНАЯ И МАТЕМАТИЧЕСКАЯ ЛИНГВИСТИКА

<i>Глушаускайте И.В., Заболотная Т.Н.</i> Комбинированный метод автоматизированного определения авторства текстов.....	102
<i>Федорова Т.М.</i> Побудова логічної мережі для флективної обробки дієслів української мови	111
<i>Каменева И.В.</i> Анализ флексии письменной и устной моделей полных непряжательных имен прилагательных русского языка	116
Об авторах	119
Правила оформления рукописів для авторів науково-технічного журналу «Біоніка інтелекту».....	122

УДК 519.7



М.Ф. Бондаренко¹, Ю.П. Шабанов-Кушнаренко¹, Д.Э. Ситников²,
П.Э. Ситникова³, А.И. Коваленко²

¹Украина, г. Харьков, ХНУРЭ, ²Украина, ХДАК,
³Украина, г. Харьков, ХГУ «НУА»

ПРИМЕНЕНИЕ ПОДСТАНОВОЧНЫХ ОПЕРАЦИЙ К ПРЕДИКАТАМ, ПРЕДСТАВЛЕННЫМ ФОРМУЛАМИ АЛГЕБРЫ КОНЕЧНЫХ ПРЕДИКАТОВ

Вводится определение подстановочной операции, а также сужающего, расширяющего и смещающего оператора. Рассмотрено действие операторов подстановки на предикаты, представленные в виде дизъюнктивной и конъюнктивной нормальной формы, найдены необходимые и достаточные условия существования сужающих и расширяющих операторов подстановки. Доказаны необходимые и достаточные условия получения последовательности сужающих (расширяющих) операторов, сводящей предикат к нулю (единице).

ПОДСТАНОВОЧНАЯ ОПЕРАЦИЯ, СУЖАЮЩИЙ ОПЕРАТОР, РАСШИРЯЮЩИЙ ОПЕРАТОР, СМЕЩАЮЩИЙ ОПЕРАТОР, ПРЕДИКАТ

Введение

Появление новых эффективных средств и методов обработки и хранения информации обеспечило базу разработки и внедрения в практику информационных систем различного уровня и назначения. Прогресс в области информационных технологий и методов проектирования программного обеспечения способствовал существенному пересмотру принятых ранее подходов к созданию информационных систем.

Представление знаний в такой системе связано с решением задач формализации знаний. Для этого разрабатываются специальные модели представления и языка описания знаний, выделяются их различные типы.

Одним из средств, используемых в качестве аппарата формализации знаний, является алгебра конечных предикатов – современный универсальный математический аппарат, позволяющий моделировать многоместные конечные отношения на рассматриваемых множествах, представляя их в виде системы логических уравнений.

Одной из задач, требующих решения в данной связи, является разработка алгебры подстановочных операций как отдельного случая алгебры конечных предикатов.

Целью данной статьи является получение зависимости между исходными предикатами, имеющими различные формы, и предикатами, полученными в результате действия подстановочных операций.

1. Действие подстановочных операций на предикат, представленный в виде дизъюнктивной нормальной формы

Рассмотрим действие оператора подстановки на предикат P , представленный в виде дизъюнктивной нормальной форма (ДНФ) и конъюнктивной нормальной формы (КНФ).

Пусть P – n -арный предикат, зависящий от переменных x_1, x_2, \dots, x_n , каждая из которых имеет области допустимых значений, $\{a_{1j}\}_{j=1}^{n_1}$, $\{a_{2j}\}_{j=1}^{n_2}$, ...,

$\{a_{ij}\}_{j=1}^{n_i}$ соответственно. Оператором подстановки $a_{ij}(P)$ назовем оператор, преобразующий предикат P в предикат P_1 путем подстановки значения a_{ij} переменной x_j в предикат P :

$$a_{ij}(P(x_1, x_2, \dots, x_n)) = P(x_1, x_2, \dots, a_{ij}, \dots, x_n) = P(x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n). \quad (1)$$

Приведем несколько примеров действия оператора подстановки.

Пусть $P(x_1, x_2, x_3) = x_1^{a_{12}} \cdot x_2^{a_{21}} \cdot x_3^{a_{33}} \vee x_2^{a_{22}} \cdot x_3^{a_{31}} \vee x_3^{a_{32}}$. Тогда $a_{11}(P) = x_2^{a_{22}} \cdot x_3^{a_{31}} \vee x_3^{a_{32}}$. Очевидно, P и $a_{11}(P) = P_1$ связаны соотношением

$$P_1 \supset P. \quad (2)$$

Пусть $P(x_1, x_2) = x_1^{a_{11}} \cdot x_2^{a_{21}} \vee x_2^{a_{12}} \cdot x_2^{a_{21}} \vee x_2^{a_{22}} \vee \dots \vee x_2^{a_{23}}$. Тогда

$$a_{11}(P) = x_2^{a_{21}} \vee x_2^{a_{22}} \vee x_2^{a_{23}} = x_2^{a_{21}} (x_1^{a_{11}} \vee x_1^{a_{12}} \vee x_1^{a_{13}} \vee x_1^{a_{14}}) \vee x_2^{a_{22}} \vee x_2^{a_{23}}$$

В этом случае P и $a_{11}(P) = P_1$ связаны соотношением

$$P_1 \subset P. \quad (3)$$

Пусть

$$P(x_1, x_2, x_3) = x_1^{a_{11}} \cdot x_2^{a_{21}} \cdot x_3^{a_{32}} \vee x_1^{a_{12}} \cdot x_2^{a_{22}} \cdot x_3^{a_{32}} \vee x_2^{a_{23}} \vee x_3^{a_{31}}.$$

Тогда $a_{11}(P) = P_1 = x_2^{a_{21}} \cdot x_3^{a_{32}} \vee x_2^{a_{23}} \vee x_3^{a_{31}}$. В данном примере не выполняется ни одно из соотношений (2), (3).

Назовем оператор $a_{ij}(P) = P_1$ *сужающим*, если для P и P_1 выполняется соотношение (2).

Назовем оператор $a_{ij}(P) = P_1$ *расширяющим*, если для P и P_1 выполняется соотношение (3).

Назовем оператор $a_{ij}(P) = P_1$ *смещающим*, если для P и P_1 не выполняются соотношения (2), (3).

Исследуем вопрос о зависимости общего вида предиката $P(x_1, x_2, \dots, x_n)$ и сужающего, расширяющего или смещающего действия оператора подстановки.

Рассмотрим унарный предикат $P(x)$, имеющий вид:

$$P(x) = x^{a_1} \vee x^{a_2} \vee \dots \vee x^{a_n}.$$

Очевидно, действие оператора подстановки a_i на данный предикат приводит к получению либо нуля (в случае, если узнавание x^{a_i} отсутствует), либо единицы (в случае, если такое узнавание присутствует). В первом случае оператор подстановки действует сужающее, во втором – расширяющее.

Предположим, что $n = 2$ и рассмотрим бинарный предикат $P(x_1, x_2)$ с областями допустимых значений переменных $\{a_1, a_2, \dots, a_n\}$ и $\{b_1, b_2, \dots, b_m\}$ соответственно. Рассмотрим действие оператора подстановки $a_1(P(x_1, x_2)) = P_1(x_2)$.

Предположим также, что предикат P имеет вид ДНФ. Все слагаемые в P разобьем на пять групп, каждая из которых содержит слагаемые следующего вида:

$$x_1^{a_1} x_2^{b_{j_1}}, j_1 \in \{1, \dots, m\};$$

$$x_1^{a_{i_1}} x_2^{b_{j_1}}, i_1 \in \{2, \dots, n\}, j_1 \in \{1, \dots, m\},$$

причем второй сомножитель в слагаемых этого типа обязательно присутствует среди аналогичных сомножителей слагаемых первого типа;

$$x_1^{a_{i_2}} x_2^{b_{j_2}}, i_2 \in \{2, \dots, n\}, j_2 \in \{1, \dots, m\};$$

$$x_1^{a_{k_1}}, k_1 \in \{2, \dots, n\};$$

$$x_1^{a_{k_2}}, k_2 \in \{1, \dots, m\}.$$

Таким образом, предикат P можно записать в следующем виде:

$$P(x_1, x_2) = \bigvee_{j_1 \in \{1, \dots, m\}} x_1^{a_1} x_2^{b_{j_1}} \vee \bigvee_{\substack{i_1 \in \{2, \dots, n\} \\ j_1 \in \{1, \dots, m\}}} x_1^{a_{i_1}} x_2^{b_{j_1}} \vee \bigvee_{\substack{i_2 \in \{2, \dots, n\} \\ j_2 \in \{1, \dots, m\}}} x_1^{a_{i_2}} x_2^{b_{j_2}} \vee \bigvee_{k_1 \in \{2, \dots, n\}} x_1^{a_{k_1}} \vee \bigvee_{k_2 \in \{1, \dots, m\}} x_2^{b_{k_2}}. \quad (4)$$

Так, например, если $P(x_1, x_2) = x_1^{a_1} x_2^{b_1} \vee x_1^{a_3} \vee x_2^{b_4} \vee x_1^{a_1} x_2^{b_3} \vee x_1^{a_2} x_2^{b_1} \vee x_1^{a_2} x_2^{b_2}$, то, заключая последовательно каждую группу разбиения, соответствующего оператору a_1 в скобки, запишем:

$$P(x_1, x_2) = (x_1^{a_1} x_2^{b_1} \vee x_1^{a_1} x_2^{b_3}) \vee (x_1^{a_2} x_2^{b_1}) \vee (x_1^{a_2} x_2^{b_2}) \vee (x_1^{a_3}) \vee (x_2^{b_4}). \quad (5)$$

Заметим, что второй сомножитель слагаемых второй группы обязательно повторит один из сомножителей слагаемых первой группы.

Итак, при действии оператора подстановки a_1 на предикат P , очевидно, что первое слагаемое в (4) преобразуется в дизъюнкцию вторых сомножителей каждой пары, второе, третье и четвертое слагаемые обратятся в нуль, а последнее останется без изменения:

$$a_1(P(x_1, x_2)) = P_1(x_2) = \bigvee_{j_1 \in \{1, \dots, m\}} x_2^{b_{j_1}} \vee \bigvee_{k_2 \in \{1, \dots, m\}} x_2^{b_{k_2}}. \quad (6)$$

Например, применяя операторы a_1 к предикату P (формула (5) из вышеприведенного примера), получаем

$$a_1(P) = P_1 = (x_2^{b_1} \vee x_2^{b_3}) \vee (x_2^{b_4}). \quad (7)$$

Представим теперь предикат P (формула (6)) в следующем виде:

$$P_1(x_2) = \bigvee_{j_1 \in \{1, \dots, m\}} x_2^{b_{j_1}} \wedge \bigvee_{i=1, n} x_1^{a_i} \vee \bigvee_{k_2 \in \{1, \dots, m\}} x_2^{b_{k_2}}.$$

Тогда

$$P_1 = \bigvee_{\substack{j_1 \in \{1, \dots, m\} \\ i=1, n}} x_1^{a_i} x_2^{b_{j_1}} \vee \bigvee_{k_2 \in \{1, \dots, m\}} x_2^{b_{k_2}}. \quad (8)$$

Очевидно, так как второй сомножитель каждой пары первого слагаемого уравнения (8) повторяет аналогичный в парах первого и второго слагаемого уравнения (4), а множество первых сомножителей первого слагаемого уравнения (8) содержит все возможные узнавания переменной x_1 , то первое слагаемое в (8) полностью включает первое и второе слагаемые в (4). Последние слагаемые уравнений (4) и (8) одинаковы. Таким образом, верны следующие утверждения:

Утверждение 1. Если в предикате $P(x_1, x_2)$ слагаемые из первой группы отсутствуют, то для P и P_1 выполнится соотношение (2), и оператор подстановки $a_1(P(x_1, x_2)) = P_1(x_2)$ действует сужающе.

Доказательство следует из того факта, что в данном случае предикат, представленный формулами (6) и (8), будет состоять только из второго слагаемого.

Пример 1. Пусть $P(x_1, x_2) = x_1^{a_2} x_2^{b_1} \vee x_1^{a_3} x_2^{b_2} \vee x_1^{a_1} \vee x_2^{b_3}$. Тогда $a(P(x_1, x_2)) = P_1(x_2) = x_2^{b_3}$ и, следовательно, $P_1 \supset P$.

Утверждение 2. Если в предикате $P(x_1, x_2)$ слагаемые из первой группы присутствуют, а слагаемые третьей и четвертой группы отсутствуют, то для P и P_1 выполнится соотношение (3), и оператор подстановки $a_1(P(x_1, x_2)) = P_1(x_2)$ действуют расширяюще.

Доказательство следует из представления предиката P_1 формулой (8).

Пример 2. Пусть $P(x_1, x_2) = x_1^{a_1} x_2^{b_1} \vee x_1^{a_1} x_2^{b_2} \vee x_1^{a_2} x_2^{b_1} \vee x_2^{b_3}$. Тогда $a(P(x_1, x_2)) = P_1(x_2) = x_2^{b_1} \vee x_2^{b_2} \vee x_2^{b_3}$ и, следовательно, $P_1 \subset P$.

Утверждение 3. В случае, если предикат P содержит слагаемые первой, третьей и четвертой группы, то не выполняются соотношения (2), (3), т.е.

$$(P_1 \supset P) \wedge (P_1 \subset P). \quad (9)$$

Пример 3. Пусть $P(x_1, x_2) = x_1^{a_1} x_2^{b_2} \vee x_1^{a_2} \vee x_2^{b_3}$. Тогда $a_1(P(x_1, x_2)) = P_1(x_2) = x_2^{b_2} \vee x_2^{b_3}$ и, следовательно, оператор a_1 действует смещающе.

Пусть теперь $P(x_1, x_2) = x_1^{a_1} x_2^{b_2} \vee x_1^{a_2} x_2^{b_3} \vee x_2^{b_1}$. Тогда $a_1(P(x_1, x_2)) = P_1(x_2) = x_2^{b_1} \vee x_2^{b_2}$ и, следовательно, оператор подстановки действует также смещающе.

Рассмотрим теперь n -арный предикат $P(x_1, x_2, \dots, x_n)$ с областью допустимых значений (ОДЗ) переменных $\{a_{11}, a_{12}, \dots, a_{1n}\}$, $\{a_{21}, a_{22}, \dots, a_{2n}\}$, \dots , $\{a_{n1}, a_{n2}, \dots, a_{nn}\}$ соответственно. Исследуем действие оператора подстановки $a_{ij}(P) = P_1$. Разобьем все слагаемые предиката P , представленного в виде ДНФ, на следующие пять классов.

1. Содержит слагаемые вида

$$x_1^{a_{1j_1}} x_2^{a_{2j_2}} \dots x_i^{a_{ij}} \dots x_n^{a_{nj_n}}, j_k \in \{1, \dots, n_k\}, k = \overline{1, n}.$$

2. Содержит слагаемые вида

$$x_1^{a_{1j_1}} x_2^{a_{2j_2}} \dots x_i^{a_{ij}} \dots x_n^{a_{nj_n}}, j_k \in \{1, \dots, n_k\}, k = \overline{1, n},$$

причем произведение всех переменных данного сомножителя, кроме узнавания переменной x_i , $(x_1^{a_{1j_1}} x_2^{a_{2j_2}} \dots x_{i-1}^{a_{i-1, j_{i-1}}} x_{i+1}^{a_{i+1, j_{i+1}}} \dots x_n^{a_{nj_n}})$ содержится среди аналогичных в слагаемых первого класса.

3. Содержит слагаемые вида

$$x_1^{a_{1j_1}} x_2^{a_{2j_2}} \dots x_i^{a_{ij}} \dots x_n^{a_{nj_n}}, l_k \in \{1, \dots, n_k\}, k = \overline{1, n},$$

причем сомножитель $x_1^{a_{1j_1}} x_2^{a_{2j_2}} \dots x_{i-1}^{a_{i-1, j_{i-1}}} x_{i+1}^{a_{i+1, j_{i+1}}} \dots x_n^{a_{nj_n}}$ отличается от аналогичных в слагаемых первого класса.

4. Содержит слагаемые вида $x_i^{a_{ij}}, l_i \in \{1, \dots, n_i\}$.

5. Содержит слагаемые вида

$$x_1^{a_{1l_1}} x_2^{a_{2l_2}} \dots x_{i-1}^{a_{i-1, l_{i-1}}} x_{i+1}^{a_{i+1, l_{i+1}}} \dots x_n^{a_{nl_n}}, l_k \in \{1, \dots, n_k\}, k = \overline{1, n}.$$

Таким образом, предикат P можно записать в виде:

$$\begin{aligned} P(x_1, x_2, \dots, x_n) = & \bigvee_{\substack{j_1 \in \{1, \dots, n_1\} \\ \dots \\ j_n \in \{1, \dots, n_n\}}} x_1^{a_{1j_1}} x_2^{a_{2j_2}} \dots x_i^{a_{ij}} \dots x_n^{a_{nj_n}} \vee \\ & \bigvee_{\substack{j_1 \in \{1, \dots, n_1\} \\ \dots \\ j_n \in \{1, \dots, n_n\}}} x_1^{a_{1j_1}} x_2^{a_{2j_2}} \dots x_i^{a_{ij}} \dots x_n^{a_{nj_n}} \vee \\ & \bigvee_{\substack{l_1 \in \{1, \dots, n_1\} \\ \dots \\ l_n \in \{1, \dots, n_n\}}} x_1^{a_{1l_1}} x_2^{a_{2l_2}} \dots x_i^{a_{il_i}} \dots x_n^{a_{nl_n}} \vee \bigvee_{l_i \in \{1, \dots, n_i\}} x_i^{a_{il_i}} \vee \\ & \bigvee_{\substack{l_1 \in \{1, \dots, n_1\} \\ \dots \\ l_n \in \{1, \dots, n_n\}}} x_1^{a_{1l_1}} x_2^{a_{2l_2}} \dots x_{i-1}^{a_{i-1, l_{i-1}}} x_{i+1}^{a_{i+1, l_{i+1}}} \dots x_n^{a_{nl_n}}. \end{aligned} \quad (10)$$

На примере

$$P(x_1, x_2, x_3) = x_1^{a_{11}} x_2^{a_{21}} \vee x_1^{a_{11}} x_2^{a_{22}} x_3^{a_{31}} \vee x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{31}} \vee x_1^{a_{12}} x_2^{a_{23}} x_3^{a_{33}} \vee x_1^{a_{13}} \vee x_2^{a_{23}} x_3^{a_{32}}$$

проведём разбиение по указанным выше пяти классам, заключая каждый из них в скобки

$$\begin{aligned} P(x_1, x_2, x_3) = & (x_1^{a_{11}} x_2^{a_{21}} \vee x_1^{a_{11}} x_2^{a_{22}} x_3^{a_{31}}) \vee \\ & \vee (x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{31}}) \vee (x_1^{a_{12}} x_2^{a_{23}} x_3^{a_{33}}) \vee \\ & \vee (x_1^{a_{13}}) \vee (x_2^{a_{23}} x_3^{a_{32}}). \end{aligned} \quad (11)$$

Применяя оператор подстановки к предикату P , заданному уравнением (10), получим

$$\begin{aligned} a_{ij}(P) = P_1 = & \bigvee_{\substack{j_1 \in \{1, \dots, n_1\} \\ \dots \\ j_n \in \{1, \dots, n_n\}}} x_1^{a_{1j_1}} x_2^{a_{2j_2}} \dots x_n^{a_{nj_n}} \vee \\ & \bigvee_{\substack{l_1 \in \{1, \dots, n_1\} \\ \dots \\ l_n \in \{1, \dots, n_n\}}} x_1^{a_{1l_1}} x_2^{a_{2l_2}} \dots x_n^{a_{nl_n}} \vee \\ & \bigvee_{\substack{l_1 \in \{1, \dots, n_1\} \\ \dots \\ l_n \in \{1, \dots, n_n\}}} x_1^{a_{1l_1}} x_2^{a_{2l_2}} \dots x_{i-1}^{a_{i-1, l_{i-1}}} x_{i+1}^{a_{i+1, l_{i+1}}} \dots x_n^{a_{nl_n}}, \end{aligned} \quad (12)$$

где первое слагаемое данного уравнения образовано из первого слагаемого уравнения (10) вследствие обращения в единицу узнавания $x_i^{a_{ij}}$, а последнее слагаемое повторяет аналогичное в уравнении (10). Второе, третье и четвёртое слагаемые уравнения (10) обращаются в нуль. На примере предиката, представленного уравнением (11), действие оператора подстановки выглядит следующим образом:

$$\begin{aligned} a_{11}(P(x_1, x_2, x_3)) = & \\ = & (x_2^{a_{21}} \vee x_2^{a_{22}} x_3^{a_{31}}) \vee (x_2^{a_{23}} x_3^{a_{32}}). \end{aligned} \quad (13)$$

Из приведенных рассуждений можно вывести утверждения:

Утверждение 4. Если предикат $P(x_1, x_2, \dots, x_n)$ не содержит слагаемых первого класса, то предикат P_1 , общий вид которого представлен уравнением (12), будет содержать только второе и третье слагаемые, а следовательно, P и P_1 связаны соотношением (2), то есть оператор подстановки действует сужающе.

Утверждение 5. Если предикат $P(x_1, x_2, \dots, x_n)$ содержит слагаемые первого класса, а слагаемые третьего и четвёртого классов отсутствуют, то, предикаты P и P_1 связаны соотношением (3), то есть оператор подстановки действует смещающе.

Утверждение 6. Если предикат $P(x_1, x_2, \dots, x_n)$ содержит слагаемое первого класса, а также слагаемые третьего или четвёртого классов, то оператор подстановки действует смещающе.

Приведём примеры, иллюстрирующие утверждение 4-утверждение 6.

Пример 4. Пусть

$$P(x_1, x_2, x_3) = x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{32}} \vee x_1^{a_{13}} x_2^{a_{21}} x_3^{a_{32}} \vee x_4^{a_{14}} \vee x_2^{a_{22}} x_3^{a_{33}}.$$

Тогда $a_{11}(P(x_1, x_2, x_3)) = P_1 = x_2^{a_{22}} x_3^{a_{33}}$. Очевидно, $P_1 \supset P$, т.е. оператор подстановки a_{11} действует сужающе.

Пример 5. Пусть

$$P(x_1, x_2, x_3) = x_1^{a_{11}} x_2^{a_{21}} \vee x_1^{a_{12}} x_2^{a_{21}} \vee x_2^{a_{22}} x_3^{a_{33}} \vee x_2^{a_{23}} x_3^{a_{32}}.$$

Тогда

$$\begin{aligned} a_{11}(P(x_1, x_2, x_3)) = P_1 = & x_2^{a_{21}} \vee x_2^{a_{22}} x_3^{a_{31}} \vee x_2^{a_{23}} x_3^{a_{32}} = \\ = & x_2^{a_{21}} (x_1^{a_{11}} \vee x_1^{a_{12}} \vee x_1^{a_{13}}) \vee x_2^{a_{22}} x_3^{a_{31}} \vee x_2^{a_{23}} x_3^{a_{32}}. \end{aligned}$$

Следовательно, $P \supset P_1$, т.е. оператор подстановки a_{11} действует расширяюще.

Пример 6. Пусть

$$P(x_1, x_2, x_3) = x_1^{a_{11}} x_2^{a_{22}} x_3^{a_{31}} \vee x_1^{a_{12}} x_2^{a_{23}} x_3^{a_{31}} \vee x_2^{a_{21}} x_3^{a_{32}}.$$

Тогда $a_{11}(P(x_1, x_2, x_3)) = P_1 = x_2^{a_{22}} x_3^{a_{31}} \vee x_2^{a_{21}} x_3^{a_{32}}$. В этом случае оператор подстановки a_{11} действует сдвигающе, так же как и в следующем примере:

$$P(x_1, x_2, x_3) = x_1^{a_{11}} x_2^{a_{22}} x_3^{a_{31}} \vee x_1^{a_{12}} \vee x_1^{a_{13}} \vee x_2^{a_{21}} x_3^{a_{32}}.$$

Тогда

$$a_{11}(P(x_1, x_2, x_3)) = P_1 = x_2^{a_{22}} x_3^{a_{31}} \vee x_2^{a_{21}} x_3^{a_{32}}.$$

2. Действие подстановочных операций на предикат, представленный в виде конъюнктивной нормальной формы

Допустим, что n -арный предикат $P(x_1, x_2, \dots, x_n)$ представлен в виде КНФ таким образом, что каждая элементарная дизъюнкция содержит узнавания только одной переменной. Например,

$$P(x_1, x_2, x_3, x_4) = (x_1^{a_{11}} \vee x_1^{a_{12}} \vee x_1^{a_{14}}) \wedge (x_2^{a_{21}} \vee x_2^{a_{22}}) \wedge (x_3^{a_{31}} \vee x_3^{a_{33}}) \wedge (x_4^{a_{41}} \vee x_4^{a_{42}}).$$

В этом случае предикат P в общем виде можно представить как

$$P(x_1, x_2, \dots, x_n) = \bigvee_{i_1 \in \{1, \dots, n_1\}} x_1^{a_{1i_1}} \wedge \bigvee_{i_2 \in \{1, \dots, n_2\}} x_2^{a_{2i_2}} \wedge \dots \wedge \bigvee_{i_n \in \{1, \dots, n_n\}} x_n^{a_{ni_n}}.$$

Рассмотрим действие оператора подстановки $a_{ij}(P) = P_1$. Возможны два случая:

Элементарная дизъюнкция, содержащая узнавания переменной x_i содержит также узнавание $x_i^{a_{ij}}$. Эта элементарная дизъюнкция не содержит такого узнавания.

В первом случае слагаемое $x_i^{a_{ij}}$ обращается в единицу, а следовательно, в единицу обращается вся скобка. Тогда, очевидно, $P \supset P_1$, т.е. оператор подстановки действует расширяюще. Во втором случае вся элементарная дизъюнкция, содержащая узнавания переменной x_i , обращается в нуль, а следовательно, весь предикат также обращается в нуль. Таким образом, в этом случае оператор действует сужающе.

Рассмотрим теперь предикат $P(x_1, x_2, \dots, x_n)$, представленный в виде произвольной КНФ. Пусть ОДЗ переменных составляют множества $\{a_{1j}\}_{j=1}^{n_1}$, $\{a_{2j}\}_{j=1}^{n_2}$, ..., $\{a_{nj}\}_{j=1}^{n_n}$ соответственно.

Разобьем все элементарные дизъюнкции на следующие три класса.

1. Сомножители, в которых присутствуют узнавание $x_i^{a_{ij}}$.

2. Сомножители, в которых отсутствует узнавание $x_i^{a_{ij}}$, но присутствуют другие узнавания переменной x_i .

3. Сомножители, не содержащие вовсе узнаваний переменной x_i .

Например, пусть предикат $P(x_1, x_2, x_3)$ имеет следующий вид:

$$P(x_1, x_2, x_3) = (x_1^{a_{11}} \vee x_1^{a_{12}} \vee x_2^{a_{21}})(x_1^{a_{11}} \vee x_2^{a_{22}} \vee x_3^{a_{31}} \vee x_3^{a_{32}}) \wedge (x_1^{a_{12}} \vee x_1^{a_{13}} \vee x_2^{a_{21}})(x_1^{a_{12}} \vee x_3^{a_{31}})(x_2^{a_{22}} \vee x_3^{a_{33}}) \wedge (x_2^{a_{21}} \vee x_3^{a_{31}} \vee x_3^{a_{32}}).$$

Тогда, исследуя оператор подстановки $a_{11}(P(x_1, x_2, x_3))$, разобьем все множители на три класса:

$$(x_1^{a_{11}} \vee x_1^{a_{12}} \vee x_2^{a_{21}}), (x_1^{a_{11}} \vee x_2^{a_{22}} \vee x_3^{a_{31}} \vee x_3^{a_{32}});$$

$$(x_1^{a_{12}} \vee x_1^{a_{13}} \vee x_2^{a_{21}}), (x_1^{a_{12}} \vee x_3^{a_{31}});$$

$$(x_2^{a_{22}} \vee x_3^{a_{33}}), (x_2^{a_{21}} \vee x_3^{a_{31}} \vee x_3^{a_{32}}).$$

Обозначим произведение всех сомножителей первого класса через A , т.е.

$$A = \bigwedge_k (x_i^{a_{ij}} \vee \bigvee_{i_1 \in \{1, \dots, n_1\}} x_1^{a_{1i_1}} \wedge \dots \wedge \bigvee_{i_n \in \{1, \dots, n_n\}} x_n^{a_{ni_n}})_k. \quad (14)$$

Во втором классе в каждой элементарной дизъюнкции можно выделить слагаемое двух типов: к первому отнесем слагаемые, являющиеся узнаваниями переменной x_i , и обозначим их сумму в каждой j -й скобке через B_j :

$$B_j = \left(\bigvee_{k_j \in \{1, \dots, n_j\}} x_i^{a_{ik_j}} \right)_j; \quad (15)$$

ко второму типу отнесем остальные слагаемые в этой элементарной дизъюнкции и обозначим их сумму через C_j в каждой j -й скобке:

$$C_j = \left(\bigvee_{\substack{k_j \in \{1, \dots, n_j\} \\ l \neq i \\ l \in \{1, \dots, n\}}} x_l^{a_{lk_j}} \right)_j. \quad (16)$$

Так, в вышеприведенном примере можно обозначить

$$B_1 = x_1^{a_{12}} \vee x_1^{a_{13}}; B_2 = x_1^{a_{12}}; C_1 = x_2^{a_{21}}; C_2 = x_3^{a_{31}}.$$

Произведение сомножителей третьей группы обозначим через D :

$$D = \bigwedge_j \left(\bigvee_{i_2 \in \{1, \dots, n_2\}} x_2^{a_{2i_2}} \wedge \dots \wedge \bigvee_{i_n \in \{1, \dots, n_n\}} x_n^{a_{ni_n}} \right)_j, \quad (17)$$

Итак, предикат P можно представить в виде

$$P(x_1, x_2, \dots, x_n) = A \wedge \bigwedge_j (B_j \vee C_j) \wedge D. \quad (18)$$

При действии оператора подстановки $a_{ij}(P)$ первый сомножитель — A обращается в единицу, во втором сомножителе слагаемые B_j обращаются в нуль, третий сомножитель D останется без изменения. Таким образом,

$$a_{ij}(P(x_1, x_2, \dots, x_n)) = \bigwedge_j C_j \wedge D. \quad (19)$$

Из вышеприведенных объяснений можно вывести следующие утверждения.

Утверждение 7. Если предикат P не содержит элементарных дизъюнкций первого класса, то оператор подстановки действует сужающее.

Доказательство. В этом случае

$$P(x_1, x_2, \dots, x_n) = \bigwedge_j (B_j \vee C_j) \wedge D, \quad (20)$$

а

$$P_1 = \bigwedge_j C_j \wedge D.$$

Так как $\bigwedge_j (B_j \vee C_j) \wedge D = (\bigwedge_j B_j \wedge \bigwedge_j C_j) \wedge D$, то, очевидно, выполняется соотношение (2).

Следствие. Если в дополнение к указанному в утверждении условию предикат P не содержит сомножителей третьей группы, то оператор подстановки действует также сужающее.

Утверждение 8. Если предикат P содержит элементарные дизъюнкции первого класса, но не содержит таковых второго класса, то оператор подстановки действует расширяюще.

Доказательство. В этом случае

$$P(x_1, x_2, \dots, x_n) = A \wedge D, \quad (21)$$

а следовательно, $P_1 = D$.

Очевидно, P и P_1 связаны соотношением (3).

Следствие. Если в дополнение к указанному в утверждении условию предикат P не содержит сомножителей третьей группы, то оператор подстановки действует также расширяюще.

Утверждение 9. Если предикат P содержит сомножители первого и второго класса, то оператор подстановки действует смешивающе.

Доказательство. В этом случае предикат P представлен формулой (18), предикат P_1 представлен формулой (19) и они не связаны, очевидно, ни одним из соотношений (2), (3).

Следствие. Если в дополнение к указанному в утверждении условию предикат P не содержит сомножителей третьей группы, то оператор подстановки действует смешивающе.

Приведем примеры, иллюстрирующие приведенные утверждения.

Пример 7. Пусть $P(x_1, x_2, x_3) = (x_1^{a_{12}} \vee x_2^{a_{21}} \vee x_3^{a_{31}}) \wedge (x_2^{a_{22}} \vee x_3^{a_{32}}) \wedge (x_2^{a_{23}} \vee x_3^{a_{33}})$. Тогда

$$a_{11}(P) = (x_2^{a_{21}} \vee x_3^{a_{31}}) \wedge (x_2^{a_{22}} \vee x_3^{a_{32}}) \wedge (x_2^{a_{23}} \vee x_3^{a_{33}}) = P_1.$$

Очевидно, $P_1 \supset P$, т.е. оператор $a_{11}(P)$ действует сужающе.

Пример 8. Пусть $P(x_1, x_2, x_3) = (x_1^{a_{11}} \vee x_2^{a_{21}}) \wedge (x_2^{a_{23}} \vee x_3^{a_{31}}) \wedge (x_3^{a_{32}} \vee x_2^{a_{22}})$. Тогда

$$a_{11}(P) = (x_2^{a_{23}} \vee x_3^{a_{31}}) \wedge (x_3^{a_{32}} \vee x_2^{a_{22}}) = P_1. \text{ Очевидно,}$$

$P \supset P_1$, т.е. оператор $a_{11}(P)$ действует расширяюще.

Пример 9. $P(x_1, x_2, x_3) = (x_1^{a_{11}} \vee x_3^{a_{31}})(x_1^{a_{22}} \vee x_2^{a_{21}} \vee x_3^{a_{32}})(x_2^{a_{22}} \vee x_3^{a_{33}})$.

Тогда $a_{11}(P) = x_1^{a_{11}} \vee x_3^{a_{31}}(x_2^{a_{22}} \vee x_3^{a_{33}}) = P_1$.

Очевидно, не выполняется (2), (3) т.е. оператор $a_{11}(P)$ действует смешивающе.

3. Применение подстановочных операций к конъюнкции дизъюнктивных нормальных форм

Рассмотрим действие оператора подстановки на конъюнкцию предикатов $Q = P_1 \wedge P_2$, где P_1 и P_2 – предикаты, представленные в виде ДНФ.

Пусть множества переменных, на которых определены предикаты P_1 и P_2 , не пересекаются. Тогда

$$\begin{aligned} a_{i,j,k}(Q) &= a_{i,j}(P_1(x_{i_1}, x_{i_2}, \dots, x_{i_n})) \wedge P_2(x_{j_1}, x_{j_2}, \dots, x_{j_n}) = \\ &= P_2 \wedge a_{i,j,k}(P_1). \end{aligned}$$

Очевидно, оператор подстановки $a_{i,j}$ действует на предикат Q так же, как и на предикат P_1 :

если $a(P_1) \supset P_1$, то $a(Q) = a(P_1) \wedge P_2 \supset P_1 \wedge P_2$;

если $a(P_1) \subset P_1$, то $a(Q) = a(P_1) \wedge P_2 \subset P_1 \wedge P_2$.

Предположим, что предикаты P_1 и P_2 имеют общие переменные x_{k_1}, x_{k_2}, \dots . Рассмотрим действие оператора подстановки $a_{k,m}(Q)$.

Утверждение 10. Если оператор подстановки $a_{k,m}$ является сужающим для предикатов P_1 и P_2 , то он является также сужающим и для их конъюнкции – предиката $Q = P_1 \wedge P_2$.

Доказательство.

$$a_{k,m}(Q) = a_{k,m}(P_1) \wedge a_{k,m}(P_2) \supset P_1 \wedge P_2,$$

т.к. $a_{k,m}(P_1) \supset P_1 \wedge a_{k,m}(P_2) \supset P_2$.

Пример 10. $P_1 = x_1^{a_{11}} x_2^{a_{22}} x_3^{a_{33}} \vee x_1^{a_{11}} x_2^{a_{23}} x_3^{a_{31}} \vee x_3^{a_{32}}$, $P_2 = x_2^{a_{22}} x_3^{a_{33}} x_4^{a_{42}} \vee x_4^{a_{41}}$. Оператор a_{21} является сужающим для предикатов P_1 , P_2 и Q : $a_{21}(Q) = x_3^{a_{32}} \vee x_4^{a_{41}}$.

Утверждение 11. Если оператор подстановки $a_{k,m}$ является расширяющим для предикатов P_1 и P_2 , то он является также расширяющим и для их конъюнкции – предиката $Q = P_1 \wedge P_2$.

Доказательство.

$$a_{k,m}(Q) = a_{k,m}(P_1) \wedge a_{k,m}(P_2) \subset P_1 \wedge P_2,$$

т.к. $a_{k,m}(P_1) \subset P_1 \wedge a_{k,m}(P_2) \subset P_2$.

Пример 11.

$$P_1 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} \vee x_1^{a_{12}} x_3^{a_{31}}, \quad P_2 = x_2^{a_{21}} x_3^{a_{31}} \vee x_3^{a_{33}} x_4^{a_{41}}.$$

Оператор a_{21} является расширяющим для предикатов P_1 , P_2 и Q :

$$a_{21}(Q) = (x_1^{a_{11}} x_3^{a_{33}} \vee x_1^{a_{12}} x_3^{a_{31}}) \wedge (x_3^{a_{31}} \vee x_3^{a_{33}} x_4^{a_{41}}).$$

Но так как $Q = x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{31}} \vee x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} x_4^{a_{41}}$, то $a_{21}(Q) \supset P_1 \wedge P_2$.

Утверждение 12. Если оператор подстановки $a_{k,m}$ является смешивающим для предикатов P_1 и P_2 , то на предикат $Q = P_1 \wedge P_2$ он может действовать любым образом.

Докажем утверждение путем приведения примеров.

Пример 12. $P_1 = x_1^{a_{11}} x_2^{a_{21}} \vee x_2^{a_{22}} x_3^{a_{31}}$,
 $P_2 = x_2^{a_{21}} x_3^{a_{33}} \vee x_2^{a_{22}}$, $a_{21}(P_1) = x_1^{a_{11}}$, $a_{21}(P_2) = x_3^{a_{33}}$;
 $P_1 \wedge P_2 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} \vee x_2^{a_{22}} x_3^{a_{31}}$, $a_{21}(Q) = x_1^{a_{11}} x_3^{a_{33}}$.

В данном случае для предиката Q оператор a_{21} является сдвигающим.

Пример 13. $P_1 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{31}} \vee x_2^{a_{22}} x_3^{a_{33}} \vee x_3^{a_{32}}$,
 $P_2 = x_2^{a_{21}} x_3^{a_{33}} \vee x_2^{a_{22}} x_4^{a_{41}} \vee x_3^{a_{32}} x_4^{a_{41}}$.

Тогда $P_1 \wedge P_2 = x_2^{a_{22}} x_3^{a_{33}} x_4^{a_{41}} \vee x_2^{a_{22}} x_3^{a_{32}} x_4^{a_{41}}$,

$a_{21}(P_1) = x_1^{a_{11}} x_3^{a_{31}} \vee x_3^{a_{32}}$, $a_{21}(P_2) = x_3^{a_{33}} \vee x_3^{a_{32}} x_4^{a_{41}}$,

то есть этот оператор является сдвигающим для предикатов P_1 и P_2 . $a_{21}(Q) = x_3^{a_{32}} x_4^{a_{41}}$. В данном случае для предиката Q оператор a_{21} является сужающим.

Пример 14. $P_1 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{31}} \vee x_2^{a_{22}} x_3^{a_{31}} \vee x_3^{a_{33}}$,

$P_2 = x_2^{a_{21}} x_3^{a_{31}} x_4^{a_{42}} \vee x_2^{a_{22}} x_3^{a_{32}} \vee x_3^{a_{33}}$.

Тогда $Q = P_1 P_2 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{31}} x_4^{a_{42}} \vee x_3^{a_{33}}$.

$a_{21}(P_1) = x_1^{a_{11}} x_3^{a_{31}} \vee x_3^{a_{33}}$, $a_{21}(P_2) = x_3^{a_{31}} x_4^{a_{42}} \vee x_3^{a_{33}}$,

то есть этот оператор является сдвигающим для предикатов P_1 и P_2 . $a_{21}(Q) = x_1^{a_{11}} x_3^{a_{31}} x_4^{a_{42}} \vee x_3^{a_{33}}$. В данном случае для предиката Q оператор a_{21} является расширяющим.

Утверждение 13. Если оператор подстановки $a_{k_{lm}}$ является сужающим для предиката P_1 и расширяющим для предиката P_2 , то на предикат $Q = P_1 \wedge P_2$ он может действовать любым образом.

Докажем утверждение путем приведения примеров.

Пример 15. $P_1 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} \vee x_3^{a_{31}}$,

$P_2 = x_2^{a_{22}} x_3^{a_{31}} \vee x_4^{a_{42}}$. $Q = P_1 P_2 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} x_4^{a_{42}} \vee$

$\vee x_2^{a_{22}} x_3^{a_{31}} \vee x_3^{a_{31}} x_4^{a_{42}}$, $a_{21}(Q) = x_1^{a_{11}} x_3^{a_{33}} x_4^{a_{42}} \vee \vee x_3^{a_{31}} x_4^{a_{42}}$.

В данном случае для предиката Q оператор a_{21} является сдвигающим.

Пример 16. $P_1 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} \vee x_3^{a_{32}}$,

$P_2 = x_2^{a_{22}} x_3^{a_{31}} \vee x_4^{a_{42}}$.

Тогда $Q = P_1 P_2 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} x_4^{a_{42}} \vee x_3^{a_{32}} x_4^{a_{42}}$.

$a_{21}(Q) = x_1^{a_{11}} x_3^{a_{33}} x_4^{a_{42}} \vee x_3^{a_{32}} x_4^{a_{42}}$.

В данном случае для предиката Q оператор a_{21} является расширяющим.

Утверждение 14. Если оператор подстановки $a_{k_{lm}}$ является сужающим для предиката P_1 и сдвигающим для предиката P_2 , то на предикат $Q = P_1 \wedge P_2$ он может действовать любым образом.

Доказательство проведем аналогично предыдущим утверждениям.

Пример 17. $P_1 = x_1^{a_{11}} x_2^{a_{22}} x_3^{a_{31}} \vee x_1^{a_{13}} x_2^{a_{21}} x_3^{a_{33}} \vee x_3^{a_{32}}$,

$P_2 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{32}} \vee x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{33}}$. $Q = P_1 P_2 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{32}}$.

В данном случае для предиката Q оператор a_{11} является расширяющим.

Пример 18. $P_1 = x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{31}} \vee x_1^{a_{13}} x_2^{a_{21}} x_3^{a_{33}} \vee x_3^{a_{32}}$,

$P_2 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{32}} \vee x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{32}} \vee x_2^{a_{23}}$.

$Q = P_1 P_2 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{32}} \vee x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{32}} \vee x_2^{a_{23}} x_3^{a_{32}}$.

В данном случае для предиката Q оператор a_{11} является сдвигающим.

Утверждение 15. Если оператор подстановки $a_{k_{lm}}$ является расширяющим для предиката P_1 и сдвигающим для предиката P_2 , то на предикат $Q = P_1 \wedge P_2$ он может действовать любым образом.

Приведем примеры, иллюстрирующие доказательство утверждения.

Пример 19. $P_1 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} \vee x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{33}} \vee x_2^{a_{22}} x_3^{a_{33}}$,

$P_2 = x_1^{a_{11}} x_2^{a_{22}} \vee x_1^{a_{12}} x_2^{a_{23}}$. $Q = P_1 P_2 = x_1^{a_{11}} x_2^{a_{22}} x_3^{a_{33}}$.

В данном случае для предиката Q оператор a_{11} является расширяющим.

Пример 20. $P_1 = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{32}} \vee x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{32}} \vee$

$\vee x_2^{a_{23}} x_3^{a_{33}} \vee x_2^{a_{21}} x_3^{a_{31}}$, $P_2 = x_1^{a_{11}} x_2^{a_{22}} \vee x_1^{a_{12}} x_2^{a_{23}} \vee \vee x_2^{a_{21}}$.

$Q = P_1 P_2 = x_1^{a_{12}} x_2^{a_{23}} x_3^{a_{33}} \vee x_2^{a_{21}} x_3^{a_{31}}$.

В данном случае для предиката Q оператор a_{11} является сужающим.

Пример 21. $P_1 = x_1^{a_{11}} x_2^{a_{22}} x_3^{a_{31}} \vee x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{31}} \vee$

$\vee x_2^{a_{23}} \vee x_3^{a_{32}}$, $P_2 = x_1^{a_{11}} x_2^{a_{22}} x_3^{a_{31}} \vee x_1^{a_{12}} x_2^{a_{23}} x_3^{a_{32}} \vee \vee x_2^{a_{23}} x_3^{a_{31}}$.

Тогда $Q = P_1 P_2 = x_1^{a_{11}} x_2^{a_{22}} x_3^{a_{31}} \vee \vee x_1^{a_{12}} x_2^{a_{23}} x_3^{a_{32}} \vee x_2^{a_{23}} x_3^{a_{31}}$.

В данном случае для предиката Q оператор a_{11} является сдвигающим.

4. Условия существования сужающих и расширяющих подстановочных операций

Утверждение 16. Пусть предикат $P(x_1, x_2, \dots, x_n)$ представлен в виде ДНФ. Для того чтобы существовал сужающий оператор подстановки по переменной x_i , необходимо и достаточно, чтобы существовало такое значение a_{ij} из области допустимых значений (ОДЗ) переменной x_i , что при разбиении слагаемых предиката P , соответствующему этому значению, элементарные конъюнкции из первой группы отсутствовали бы.

Доказательство. Необходимость. Пусть существует такое значение a_{ij} , что выполняется условие, сформулированное в утверждении. Тогда, из утверждения 1 следует, что оператор подстановки $a_{ij}(P)$ действует сужающе, что и требовалось доказать.

Достаточность. Пусть существует сужающий оператор подстановки для предиката P . Обозначим его $a_{ij}(P)$. Предположим, что предикат P содержит слагаемые вида

$$x_i^{a_{ij}} \wedge \bigwedge_{\substack{q \in \{1, \dots, n\} \\ n \neq q}} \bigwedge_{g \in \{1, \dots, n\}} \quad (22)$$

т.е. слагаемое из первой группы. Тогда, из утверждений 2 и 3, приведенных ранее, следует, что условие

$P_1 \supset P$ не выполняется, то есть оператор действует не сужающе, что противоречит условию утверждения. Следовательно, наше предположение не верно, а значит, слагаемые типа (22) отсутствуют.

Приведем пример. Пусть

$$P(x_1, x_2, x_3) = x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{33}} \vee x_1^{a_{13}} x_2^{a_{22}} x_3^{a_{31}} \vee x_3^{a_{32}} \vee x_2^{a_{23}}.$$

Найдём операторы подстановки по переменным x_1 и x_3 , действующие сужающе при условии, что ОДЗ каждой переменной состоит из трёх значений. Итак, для переменной x_1 отсутствует узнавание $x_1^{a_{11}}$, значит, оператор $a_{11}(P)$ действует сужающе:

$$a_{11}(P) = x_3^{a_{32}} x_2^{a_{23}} = P_1 \rightarrow P_1 \supset P.$$

Для переменной x_3 не существует сужающего оператора подстановки, так как все узнавания данной переменной представлены в данном предикате.

Утверждение 17. Пусть предикат $P(x_1, x_2, \dots, x_n)$ представлен в виде КНФ. Для того чтобы существовал сужающий оператор подстановки по переменной x_i , необходимо и достаточно, чтобы существовало такое значение a_{ij} из ОДЗ переменной x_i , что при разбиении сомножителей предиката P , соответствующего этому значению, элементарные дизъюнкции первого типа отсутствовали бы.

Доказательство. Необходимость. Пусть существует такое значение a_{ij} , что выполняется условие, сформулированное в утверждении. Тогда, из утверждения 7, приведенного ранее, следует, что оператор подстановки $a_{ij}(P)$ действует сужающе, что и требовалось доказать.

Достаточность. Пусть существует сужающий оператор подстановки для предиката P ; обозначим его $a_{ij}(P)$. Предположим, что сомножители первого типа при разбиении, соответствующем значению a_{ij} , присутствуют, т.е. предикат P содержит элементарные дизъюнкции вида

$$x_i^{a_{ij}} \vee \bigvee_{\substack{q \in \{1, \dots, n\} \\ n \neq q \\ g \in \{1, \dots, n\}}} x_g^{a_{gq}} \quad (23)$$

Тогда, из утверждений 8 и 9, приведенных ранее, следует, что условие $P_1 \supset P$ не выполняется, то есть оператор действует не сужающе, что противоречит условию утверждения. Следовательно, наше предположение неверно, а значит, слагаемые типа (23) отсутствуют.

Например, пусть $P(x_1, x_2, x_3) = (x_1^{a_{11}} \vee x_2^{a_{21}} \vee x_3^{a_{33}})(x_1^{a_{12}} \vee x_2^{a_{21}} \vee x_3^{a_{31}}) \wedge (x_1^{a_{11}} \vee x_3^{a_{32}})$. Найдём для переменной x_2 оператор подстановки, действующий сужающе. Так как предикат P не содержит узнаваний $x_2^{a_{22}}$ и $x_2^{a_{23}}$, то операторы $a_{22}(P)$ и $a_{23}(P)$ действуют сужающее. $a_{22}(P) = a_{23}(P) = (x_1^{a_{11}} \vee x_3^{a_{33}})(x_2^{a_{21}} \vee x_3^{a_{31}})(x_1^{a_{11}} \vee x_3^{a_{32}}) = P_1$. Очевидно, $P_1 \supset P$.

Пусть теперь $P(x_1, x_2, x_3) = (x_1^{a_{11}} \vee x_1^{a_{12}})(x_1^{a_{12}} \vee x_2^{a_{21}} \vee x_3^{a_{33}})(x_2^{a_{23}} \vee x_3^{a_{32}})$. Тогда, т.к. узнавание $x_1^{a_{13}}$ отсутствует, то оператор подстановки $a_{13}(P)$ действует сужающе. $a_{13}(P) = 0$, следовательно $P_1 \supset P$.

Утверждение 18. Пусть предикат $P(x_1, x_2, \dots, x_n)$ представлен в виде ДНФ. Для того чтобы существовал расширяющий оператор подстановки по переменной x_i , необходимо и достаточно, чтобы существовало такое значение a_{ij} из ОДЗ переменной x_i , что при разбиении слагаемых предиката P , соответствующего этому значению, слагаемые первого класса присутствовали бы, в то время как слагаемые третьего и четвертого классов отсутствовали бы.

Доказательство этого и ниже приведенных утверждений 19, 20 и 21 полностью аналогичны приведенным для утверждений 16 и 17.

Пример 22. Пусть

$$P(x_1, x_2, x_3) = x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{31}} \vee x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{31}} \vee x_1^{a_{13}} \vee x_1^{a_{11}} x_3^{a_{32}}.$$

Тогда $a_{22}(P) = a_{21}(P) = x_1^{a_{12}} x_3^{a_{31}} \vee x_1^{a_{13}} \vee x_1^{a_{11}} x_3^{a_{32}} = P_1$, т.е. $P \supset P_1$, следовательно, операторы $a_{22}(P)$ и $a_{21}(P)$ действуют расширяюще.

Утверждение 19. Пусть предикат $P(x_1, x_2, \dots, x_n)$ представлен в виде КНФ. Для того чтобы существовал расширяющий оператор подстановки по переменной x_i , необходимо и достаточно, чтобы существовало такое значение a_{ij} из ОДЗ переменной x_i , что при разбиении сомножителей предиката P , соответствующему значению, элементарные дизъюнкции первого типа присутствовали бы, а элементарные дизъюнкции второго типа отсутствовали бы.

Пример 23. $P(x_1, x_2, x_3) = (x_1^{a_{11}} \vee x_2^{a_{21}} \vee x_3^{a_{31}}) \wedge$

$$\wedge (x_1^{a_{11}} \vee x_3^{a_{32}}) \wedge (x_1^{a_{12}} \vee x_2^{a_{21}} \vee x_2^{a_{23}} \vee x_3^{a_{32}}).$$

Тогда оператор подстановки $a_{21}(P)$ действует расширяюще, т.к. $a_{21}(P) = x_1^{a_{11}} \vee x_3^{a_{32}}$ и, очевидно, $P \supset P_1$.

Утверждение 20. Пусть предикат $P(x_1, x_2, \dots, x_n)$ представлен в виде ДНФ. Для того чтобы существовал смешивающий оператор подстановки по переменной x_i , необходимо и достаточно, чтобы существовало такое значение a_{ij} из ОДЗ переменной x_i , что при разбиении слагаемых предиката P , соответствующему этому значению, слагаемые первого класса, а также третьего или четвертого класса присутствовали бы.

Пример 24.

$$P(x_1, x_2, x_3) = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} \vee x_1^{a_{12}} x_2^{a_{22}} x_3^{a_{33}} \vee x_3^{a_{31}}.$$

Тогда $a_{21}(P) = x_1^{a_{11}} x_3^{a_{33}} \vee x_3^{a_{31}} = P_1$, и, очевидно, оператор действует смешивающе.

Или пусть

$$P(x_1, x_2, x_3) = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} \vee x_1^{a_{12}} x_2^{a_{21}} x_3^{a_{33}} \vee x_3^{a_{31}}.$$

Тогда $a_{11}(P) = x_2^{a_{21}} x_3^{a_{33}}$ и очевидно, $a_{11}(P)$ также является смешивающим оператором подстановки.

Утверждение 21. Пусть предикат $P(x_1, x_2, \dots, x_n)$ представлен в виде КНФ. Для того чтобы существовал сдвигающий оператор подстановки по переменной x_i , необходимо и достаточно, чтобы существовало такое значение a_{ij} из ОДЗ переменной x_i , что при разбиении сомножителя предиката P , соответствующего этому значению, элементарные дизъюнкции первого и второго типа присутствовали бы.

Пример 25. Пусть $P(x_1, x_2, x_3) = (x_1^{a_{11}} \vee x_2^{a_{21}} \vee$

$$\vee x_3^{a_{31}}) \wedge (x_1^{a_{11}} \vee x_3^{a_{32}})(x_1^{a_{12}} \vee x_2^{a_{21}} \vee x_2^{a_{23}} \vee x_3^{a_{32}}).$$

Тогда оператор подстановки $a_{11}(P)$ действует сдвигающе, т.к. $a_{11}(P) = x_2^{a_{21}} \vee x_2^{a_{23}} \vee x_3^{a_{32}}$.

5. Применение последовательности сужающих и расширяющих подстановочных операций к одному предикату

Для построения последовательности сужающих (расширяющих, сдвигающих) операторов для предиката P необходимо последовательно находить переменные, по которым такие операторы существуют, и действовать одним из них на предикат P . Из утверждений 16-21 следует способ нахождения сужающих операторов. Например, $P(x_1, x_2, x_3) = x_1^{a_{11}} x_2^{a_{21}} \vee x_2^{a_{22}} x_3^{a_{31}} \vee x_2^{a_{22}} x_3^{a_{32}} \vee x_1^{a_{12}} x_2^{a_{23}}$. Если область допустимых значений каждой переменной состоит из трех значений, то по переменной x_1 существует сужающий оператор a_{13} . Тогда $a_{13}(P) = x_2^{a_{21}} x_3^{a_{31}} \vee x_2^{a_{22}} x_3^{a_{32}}$. Заметим, что если для исходного предиката не существовало сужающего оператора по переменной x_2 , то для полученного такой оператор существует: $a_{23}(P) = 0$.

Очевидно, последовательное применение сужающих (расширяющих) операторов приводит либо к нулю (единице), либо к предикату P_k , для которого на k -м шаге не найдется ни одного сужающего (расширяющего) оператора.

Правомерен вопрос: когда для данного предиката P существует последовательность сужающих (расширяющих) операторов, поочередное действие которых приводит к нулю (единице).

Утверждение 22. Для того чтобы для предиката P , представленного в виде ДНФ, существовала последовательность сужающих операторов подстановки, сводящих данный предикат к нулю, необходимо и достаточно, чтобы предикат P нельзя было представить в виде

$$P(x_1, x_2, \dots, x_n) = P_1(x_1, x_2, \dots, x_n) \vee P_2(x_i, x_{i_2}, \dots, x_{i_k}), \quad (24)$$

где $x_i, x_{i_2}, \dots, x_{i_k}$ – переменные, по которым не существует сужающих операторов подстановки.

Доказательство. Необходимость.

Предположим, существует последовательность сужающих операторов $a_{i_1 j_1}, a_{i_2 j_2}, \dots, a_{i_n j_n}$ такая, что $a_{i_n j_n}(a_{i_{n-1} j_{n-1}}(\dots(a_{i_1 j_1}(P)))) = 0$. Пусть также предикат

P можно представить в виде (22). Очевидно, применяя операторы $a_{i_1 j_1}, a_{i_2 j_2}, \dots, a_{i_n j_n}$ к предикату P , изменяться будет лишь слагаемое, представленное предикатом P_1 в то время, как P_2 останется без изменения. Следовательно, на определенном шаге не найдется ни одного сужающего оператора подстановки, а значит, допущение неверно, и предикат P не может быть представлен в виде (22).

Достаточность. Пусть предикат P не может быть представлен в виде (22). Это означает, что хоть одно узнавание каждой из переменных, по которым первоначально не существует сужающих операторов, присутствуют в одной элементарной конъюнкции с переменными, для которых такие операторы существуют. Таким образом, при действии ими на предикат соответствующие слагаемые обратятся в нуль, и, следовательно, появятся новые сужающие операторы.

Например, пусть $P(x_1, x_2, x_3) = x_1^{a_{11}} x_2^{a_{21}} x_3^{a_{33}} \vee x_2^{a_{22}} x_3^{a_{31}} \vee x_2^{a_{23}} x_3^{a_{32}}$. По переменной x_2 нет сужающих операторов, но узнавание $x_2^{a_{21}}$ находится в одной элементарной конъюнкции с переменной x_2 , по которой существуют сужающие операторы a_{12} и a_{13} . Итак, $a_{12}(P) = P_1 = x_2^{a_{22}} x_3^{a_{31}} \vee x_2^{a_{23}} x_3^{a_{32}}$. Тогда $a_{21}(P_1) = 0$.

Пусть $P(x_1, x_2, x_3) = x_1^{a_{12}} x_2^{a_{21}} \vee x_2^{a_{21}} x_3^{a_{31}} \vee x_2^{a_{22}} x_3^{a_{32}} \vee x_2^{a_{23}} x_3^{a_{33}}$. Тогда P можно представить в виде (22), где $P_1 = x_1^{a_{12}} x_2^{a_{21}}$, $P_2 = x_2^{a_{21}} x_3^{a_{31}} \vee x_2^{a_{22}} x_3^{a_{32}} \vee x_2^{a_{23}} x_3^{a_{33}}$, значит, не существует последовательности операторов, сводящих данный предикат к нулю.

Утверждение 23. Для того чтобы для предиката P , представленного в виде КНФ, существовала последовательность сужающих операторов подстановки, сводящих данный предикат к нулю, необходимо и достаточно, чтобы предикат P нельзя было представить в виде

$$P(x_1, x_2, \dots, x_n) = P_1(x_1, x_2, \dots, x_n) \wedge P_2(x_{i_1}, x_{i_2}, \dots, x_{i_k}), \quad (25)$$

где $x_{i_1}, x_{i_2}, \dots, x_{i_k}$ – переменные, по которым существуют сужающие операторы подстановки.

Доказательство. Необходимость.

Предположим, существует последовательность сужающих операторов $a_{i_1 j_1}, a_{i_2 j_2}, \dots, a_{i_n j_n}$ такая, что $a_{i_n j_n}(a_{i_{n-1} j_{n-1}}(\dots(a_{i_1 j_1}(P)))) = 0$. Пусть также предикат P нельзя представить в виде (23), т.е. все элементарные дизъюнкции содержат переменные, для которых не существует сужающих операторов. Очевидно, в этом случае при действии сужающих операторов по переменным, кроме $x_{i_1}, x_{i_2}, \dots, x_{i_k}$, эти переменные останутся без изменения. Таким образом, действие последовательности сужающих операторов не приводит к нулю, а значит, предположение неверно.

Достаточность. Пусть предикат P может быть представлен в виде (23). Это означает, что действуя последовательно сужающими операторами на

предикат P , второй сомножитель в (23) обратится в нуль, тем самым обращая в нуль весь предикат, что и требовалось доказать.

Утверждение 24. Для того чтобы для предиката P , представленного в виде ДНФ, существовала последовательность расширяющих операторов подстановки, сводящих данный предикат к единице, необходимо и достаточно, чтобы предикат P нельзя было представить в виде

$$P(x_1, x_2, \dots, x_n) = P_1(x_1, x_2, \dots, x_n) \vee \vee P_2(x_{i_1}, x_{i_2}, \dots, x_{i_k}), \quad (26)$$

где $x_{i_1}, x_{i_2}, \dots, x_{i_k}$ – переменные, по которым существуют расширяющие операторы подстановки.

Утверждение 25. Для того чтобы для предиката P , представленного в виде КНФ, существовала последовательность расширяющих операторов подстановки, сводящих данный предикат к единице, необходимо и достаточно, чтобы предикат P можно было представить в виде

$$P(x_1, x_2, \dots, x_n) = P_1(x_1, x_2, \dots, x_n) \wedge \wedge P_2(x_{i_1}, x_{i_2}, \dots, x_{i_k}), \quad (27)$$

где $x_{i_1}, x_{i_2}, \dots, x_{i_k}$ – переменные, по которым не существует расширяющих операторов подстановки.

Доказательство утверждений 18 и 19 аналогично доказательству утверждений 16 и 17.

Выводы

Таким образом, в статье введено определение операции подстановки, а также исследованы результаты ее действия на предикат, представленный в виде дизъюнктивной и конъюнктивной нормальной формы. Введены также определения сужающих, расширяющих и смещающих операторов,

а также исследованы условия их существования. При этом найдены необходимые и достаточные условия существования различных операций, введенных в алгебре подстановочных операций, и построения последовательности таких операций для получения специальных видов предикатов.

Список литературы: 1. Шабанов-Кушнарченко, Ю.П. Теория интеллекта. Математические средства [Текст] / Ю.П. Шабанов-Кушнарченко. – Харьков, – 1984. – 144 с. 2. Шабанов-Кушнарченко, Ю.П. Теория интеллекта. Технические средства средства [Текст] / Ю.П. Шабанов-Кушнарченко. – Харьков, – 1986. – 176 с. 3. Шабанов-Кушнарченко, Ю.П. Теория интеллекта. Проблемы и перспективы средства [Текст] / Ю.П. Шабанов-Кушнарченко. – Харьков, – 1987. – 210 с.

Поступила в редколлегию 9.11.2011

УДК 519.7

Застосування підстановочних операцій до предикатів, що представлені формулами алгебри предикатів / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнарченко, Д.Е. Ситніков, П.Е. Ситнікова, А.І. Коваленко // Біоніка інтелекту: наук.-техн. журнал. – 2012. – № 1 (78). – С. 3-11.

Отримано залежності між вихідними предикатами, що мають різноманітні форми, та предикатами, отриманими у результаті дій підстановочних операцій.

Бібліогр.: 3 найм.

UDC 519.7

Application of substitution operations to predicates represented with the help of finite predicates algebra / M.F. Bondarenko, U.P. Shabanov-Kushnarenko, D.E. Sitnikov, P.E. Sitnikova, A.I. Kovalenko // Bionics of Intelligence: Sci. Mag. – 2012. – № 1 (78). – P. 3-11.

Dependencies between original predicates in various forms and predicates obtained as a result of the application of substitution operations have been obtained.

Ref.: 3 items.

УДК 347.77

О.И. Король¹, Н.В. Шаронова²¹ ХПИ, г. Харьков, Украина, korolka@bk.ru² ХПИ, г. Харьков, Украина, nvsharonova@mail.ru

ИНТЕЛЛЕКТУАЛЬНАЯ ОБРАБОТКА ДАННЫХ ПРИ ФОРМИРОВАНИИ ПАТЕНТНО-КОНЪЮНКТУРНЫХ БАЗ ЗНАНИЙ

Создание модели интеллектуальной системы, ориентированной на описание закономерностей в текстовой патентно-конъюнктурной информации, представленных на естественном языке, и моделирование всех уровней лингвистической обработки в процессе формирования баз знаний с помощью универсального математического аппарата, основанного на алгебре конечных предикатов.

ИНТЕЛЛЕКТУАЛЬНАЯ ОБРАБОТКА ДАННЫХ, ПАТЕНТНО-КОНЪЮНКТУРНАЯ ИНФОРМАЦИЯ, АЛГЕБРА КОНЕЧНЫХ ПРЕДИКАТОВ

Введение

Современные справочно-информационные поисковые программы и патентные базы данных сохраняют детали того или иного объекта интеллектуальной собственности с довольно высокой точностью. Однако объемы данных растут, они легкодоступны в сети Интернет, но при этом не обладают точной и понятной структурой и не являются знанием в полном смысле слова. При обработке информации нужно получить закономерности, а не потоки и списки данных. Таким образом, стоит задача интеллектуальной обработки патентно-конъюнктурной информации (ПКИ) для извлечения знаний, существующих в хранилищах данных, а также применение лингвистических технологий для более полной обработки.

Рассмотрим основные особенности патентно-конъюнктурных данных, представленных в системах хранения информации, с точки зрения поиска закономерностей, существующих в них:

- Как правило, описание объектов содержит не менее 25-50 характерных признаков или полей базы данных, где каждый признак может быть дискретным (и иметь 5-10 и более значений) или непрерывным.
- Множество значений и признаков не является окончательным и может измениться.
- Критерии, определяющие качество объектов, носят как формальный, так и экспертный характер, т.е. не всегда могут быть выражены явной зависимостью.
- Объем данных достаточно велик и растет, причем некоторые существенные в прошлом по отношению к данной задаче объекты теряют свое качество от времени из-за постоянного развития технологий.
- Опытный эксперт обладает интуитивным знанием закономерности, определяющей качественный объект, и способен отделить существенные по отношению к данной задаче объекты от несущественных, но не может выразить свои знания в явном виде.
- Процесс оценки и ранжирования объектов занимает достаточно существенное время.

Анализ последних исследований показал, что наиболее известной технологией интеллектуальной

обработки данных является интеллектуальный анализ данных (Data Mining) – исследовательский анализ данных, имеющий целью отыскание интересных взаимосвязей между данными, скрытых закономерностей, которые могут использоваться при принятии решений [1]. Основная особенность Data Mining – объединение широкого математического инструментария и последних достижений в сфере информационных технологий, разработанных на основе искусственного интеллекта к организации процесса извлечения знаний из потока данных. К наиболее известным подходам относят системы на основе нейронных сетей [2], статистических методов [3, 8], нечеткой логики, методов обобщения по примерам объектов (KAD [4], АТ-ТЕХНОЛОГИЯ [5], INDUCE [6] и др.), которые обеспечивают работу в средах с разными типами данных и могут работать с экспертом, не являющимся программистом. Рассматриваемые системы реализуют процесс обобщения и некоторый уровень обработки входной информации для подготовки исследуемых данных. Например, АТ-ТЕХНОЛОГИЯ позволяет преобразовывать структурированную информацию базы данных в базу знаний экспертной системы на основе применения алгоритмов класса ID3 [5].

Из вышесказанного видно, что поиск закономерностей вручную задача трудоемкая и требует применения современных технологий автоматизации обработки данных и интеллектуальных систем, ориентированных на описание закономерностей и моделирование всех уровней лингвистической обработки текстовой ПКИ. На сегодняшний день подобные технологии не применяются, а поиск ПКИ осуществляется путем обращения к государственным патентным базам данных и с помощью универсальных интернет поисковиков (Google, Yandex, Yahoo и др.).

1. Интеллектуальные системы обобщения

Рассмотрим интеллектуальную систему «Трейд» [7], обеспечивающую пользователю поддержку при обработке потоков данных, поступающих из различных источников, и попробуем применить ее для нашей прикладной области. Архитектура интеллектуальной системы «Трейд» приведена на рис. 1.

Основой системы является алгоритм обобщения по примерам, подробно рассмотренный в ряде работ [5, 6]. Алгоритм требует представления описания рассматриваемой области как многомерного дискретного пространства D_n , где n – число координат-характеристик объекта $X = \{x_1, x_2, \dots, x_n\}$, а описания объектов как точки пространства D_n в виде вектора, содержащего значения признаков $\phi = \{a_1, a_2, \dots, a_n\}$, где a_i – значение i -го параметра. Пространство содержит неизвестные, т.е. не получившие оценку эксперта факты, и известные, составляющие базу фактов. База фактов разделена на две части: T – существенные объекты по отношению к данной задаче и F – несущественные объекты. Разделение осуществляется либо экспертом, либо с помощью набора критериев $C = \{f_1, \dots, f_k\}$, где C_i определяет i -ю локальную закономерность на данном наборе фактов, а f_k – реализация k -го критерия для C_i .

На множествах T и F строится разделяющая их функция выбора $y(\phi_i)$ такая, что $y(\phi) \geq 0$, если $\phi \in T$ и $y(\phi) < 0$ если $\phi \in F$. Для неизвестной части пространства D_n функция выбора будет разделять объекты в соответствии с законом, полученным на базе фактов. В работе [3] рассмотрен процесс построения многоуровневой функции выбора для работы в больших пространствах.

Система «Трейд», использующая рассмотренный алгоритм обобщения, должна выделить признаки и их описания для каждой предметной области, определить базы фактов для каждой из локальных закономерностей C_i , построить функцию

выбора и сохранить ее в базу знаний. Для получения результата необходимо осуществить следующие действия.

- Выделить поля БД, которые являются входными характеристиками (множество X).
- Выделить поля, являющиеся оценками ситуаций, или ввести эти оценки с помощью эксперта.
- Обратиться к блоку выделения подпространств для определения множества целей $\{C_1, C_2, \dots, C_s\}$ и построить множество Φ_i (множество существенных примеров для i -й цели) и его дополнение $1 - \Phi_i$ (множество несущественных примеров для i -й цели) по каждой цели C_i .

Для каждой цели C_i необходимо породить многоуровневую функцию выбора [7].

Видно, что система «Трейд» легко работает, когда пользователю надо сформировать запрос из одного-двух критериев. Ситуация усложняется, когда цель определена множеством критериев (например, не только поиск существующего технического решения, но и учет степени тождественности или схожести до степени смешения и т.п.). Это свидетельствует о существовании локальных закономерностей внутри общего поля знания. Причем чем сложнее описание объекта, тем больше будет локальных областей (при условии существования большой выборки соответствующих им фактов). Следовательно, нам необходим универсальный математический аппарат, который был бы ориентирован на моделирование всех уровней лингвистической обработки текстовой патентно-конъюнктурной документации (ПКД), а также поддерживал процесс формирования

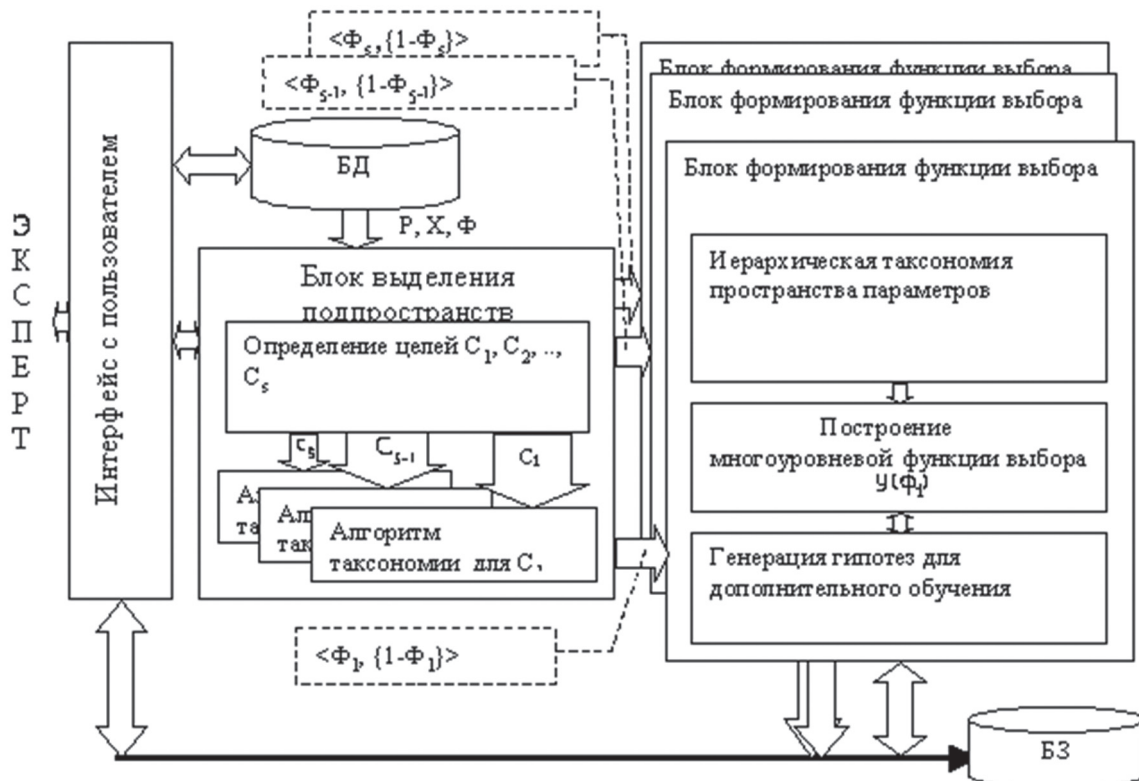


Рис. 1. Архитектура интеллектуальной системы «Трейд»

исходных наборов фактов для работы алгоритмов обобщения с учетом следующих функциональных возможностей [9]:

- Выделение множества целей на основе критериев качества объектов.
- Выделение существенных признаков, коррекция.
- Формализация описания предикатов, которые реализуются при любом виде интеллектуальной обработки ПКИ, для формирования уровней, описывающих свойства этих предикатов.
- Поддержка процесса извлечения закономерностей на основе алгоритма обобщения.
- Поддержка процесса немонотонного развития системы во времени.

2. Алгебра конечных предикатов — универсальный аппарат для описания закономерностей ПКИ

Вышеуказанные функции можно осуществить с помощью алгебры конечных предикатов (АКП) [10]. АКП полна в том смысле, что любой ее предикат можно представить в виде суперпозиции базисных операций, примененных к базисным элементам.

АКП характеризуется алфавитом A , состоящим из k символов $\{a_1, a_2, \dots, a_k\}$ и алфавитом переменных B , состоящим из n символов $\{x_1, x_2, \dots, x_n\}$ [11]. В логике высказываний «атомом» является высказывание, которое далее не разделяется — предикат. Предикат P , заданный на множестве U^n , представляет собой любую функцию $\varepsilon = P(x_1, x_2, \dots, x_n)$, отображающую данное множество U^n в $\Sigma = \{0, 1\}$.

Алгебра предикатов при любом значении n является разновидностью булевой алгебры, в ней выполняются все основные тождества булевой алгебры [10]. Базисными предикатами для АКП являются предикаты вида:

$$x_i^a = \begin{cases} 1, & \text{если } x_i = a \\ 0, & \text{если } x_i \neq a \end{cases} \quad (1 \leq i \leq n), \quad (1)$$

где $i = \{1, 2, \dots, n\}$, a — любой элемент универсума.

Прежде, чем представить системы предикатов ПКИ в виде, понятном АКП, предлагается создать онтологии прикладной области. Построение онтологии предполагает определение классов объектов и описание их отношений с помощью одного из формальных языков, например, дескриптивной логики, что позволяет отвечать на запросы или так

называемые компетентные вопросы (competency questions), которые изначально составляются на естественном языке и затем «переводятся» на используемый формальный язык. Процесс построения онтологии проиллюстрирован на рис. 2 и состоит из следующих этапов [12-13]:

- Извлечение терминов: этот этап состоит в том, чтобы обнаружить и извлечь из входного корпуса термины и их свойства. Для этого используются специализированные информационные ресурсы — глоссарий патентных терминов и терминов по интеллектуальной собственности, синтаксические шаблоны. С помощью синтаксического анализатора извлекаются пары (объект, свойство) и триплеты (объект, свойство, объект), относящиеся к общим смысловым блокам.

- Извлечение внешних отношений: на этом этапе применяется реляционный анализ понятий для извлечения внешних отношений.

- В завершение, результаты выполнения двух предыдущих этапов объединяются для получения более полной онтологии.

Таким образом, часть системы онтологий ПКИ может иметь вид, представленный на рис. 3, а система предикатов описана формулами (2) – (7).

Промышленная собственность $X_1^{ПС}$:

$$\begin{cases} X_1^{ИПМ} \vee X_1^{ПО} \vee X_1^{ТИМС} = 1, \\ X_1^{ИПМ} \wedge X_1^{\overline{ИПМ}} = 0, \\ X_1^{ПО} \wedge X_1^{\overline{ПО}} = 0, \\ X_1^{ТИМС} \wedge X_1^{\overline{ТИМС}} = 0, \end{cases} \quad (2)$$

где $X_1^{ИПМ}$ — изобретения и полезные модели, $X_1^{ПО}$ — промышленные образцы, $X_1^{ТИМС}$ — топографии интегральных микросхем.

Объекты изобретения и полезной модели $X_{11}^{ИПМ}$:

$$\begin{cases} X_{11}^H \vee X_{11}^{ПН} = 1, \\ X_{11}^H \wedge X_{11}^{\overline{ПН}} = 0, \\ X_{11}^{ПН} \wedge X_{11}^{\overline{ПН}} = 0, \end{cases} \quad (3)$$

где X_{11}^H ($X_{11}^{\overline{ПН}}$) — новое (не новое) изобретение, $X_{11}^{ПН}$ ($X_{11}^{\overline{ПН}}$) — применение (не применение) ранее известного изобретения по новому назначению.

Топографии интегральных микросхем $X_{12}^{ТИМС}$:

$$\begin{cases} X_{12}^{ТИМС} = X_1^{ПС} \wedge X_0^{ИС}, \\ X_{12}^{ТИМС} \wedge X_{12}^{\overline{ТИМС}} = 0, \end{cases} \quad (4)$$



Рис. 2. Методика построения онтологии

где $X_0^{ИС}$ – интеллектуальная собственность, $X_1^{ПС}$ – промышленная собственность, $X_{12}^{ТИМС}$ – топографии интегральных микросхем, $X_{12}^{ТИМС}$ – не топографии интегральных микросхем.

Объекты промышленного образца (ПО) $X_{13}^{ПО}$:

$$\begin{cases} X_{13}^{K1} \vee X_{13}^{K2} \vee \dots \vee X_{13}^{K32} = 1, \\ X_{13}^{K1} \wedge X_{13}^{K1} = 0, \\ X_{13}^{K2} \wedge X_{13}^{K2} = 0, \\ \dots \dots \dots \\ X_{13}^{K32} \wedge X_{13}^{K32} = 0, \end{cases} \quad (5)$$

где $X_{13}^{K1}, X_{13}^{K2}, \dots, X_{13}^{K32}$ – классы ПО, $X_{13}^{K1}, X_{13}^{K2}, \dots, X_{13}^{K32}$ – не классы ПО.

Объекты нового изобретения X_{111}^H :

$$\begin{cases} X_{111}^{ПЦ} \vee X_{111}^Y \vee X_{111}^{Be} \vee X_{111}^{ШТ} = 1, \\ X_{111}^{ПЦ} \wedge X_{111}^{ПЦ} = 0, \\ X_{111}^Y \wedge X_{111}^Y = 0, \\ X_{111}^{Be} \wedge X_{111}^{Be} = 0, \\ X_{111}^{ШТ} \wedge X_{111}^{ШТ} = 0, \end{cases} \quad (6)$$

где $X_{111}^{ПЦ}$ – процесс, X_{111}^Y – устройство, X_{111}^{Be} – вещество, $X_{111}^{ШТ}$ – штамм микроорганизма, $X_{111}^{ПЦ}$ – не процесс, X_{111}^Y – не устройство, X_{111}^{Be} – не вещество, $X_{111}^{ШТ}$ – не штамм микроорганизма.

Классификация новых устройств X_{111}^Y :

$$\begin{cases} X_{1112}^A \vee X_{1112}^B \vee X_{1112}^C \vee X_{1112}^D \vee X_{1112}^E \vee X_{1112}^F \vee \\ \vee X_{1112}^G \vee X_{1112}^H = 1, \\ X_{1112}^A \wedge X_{1112}^A = 0, \\ X_{1112}^B \wedge X_{1112}^B = 0, \\ X_{1112}^C \wedge X_{1112}^C = 0, \\ X_{1112}^D \wedge X_{1112}^D = 0, \\ X_{1112}^E \wedge X_{1112}^E = 0, \\ X_{1112}^F \wedge X_{1112}^F = 0, \\ X_{1112}^G \wedge X_{1112}^G = 0, \\ X_{1112}^H \wedge X_{1112}^H = 0, \end{cases} \quad (7)$$

где $X_{1112}^A, X_{1112}^B, X_{1112}^C, X_{1112}^D, X_{1112}^E, X_{1112}^F, X_{1112}^G, X_{1112}^H$ – классы новых изобретений в зависимости от сферы применения, $X_{1112}^A, X_{1112}^B, X_{1112}^C, X_{1112}^D,$

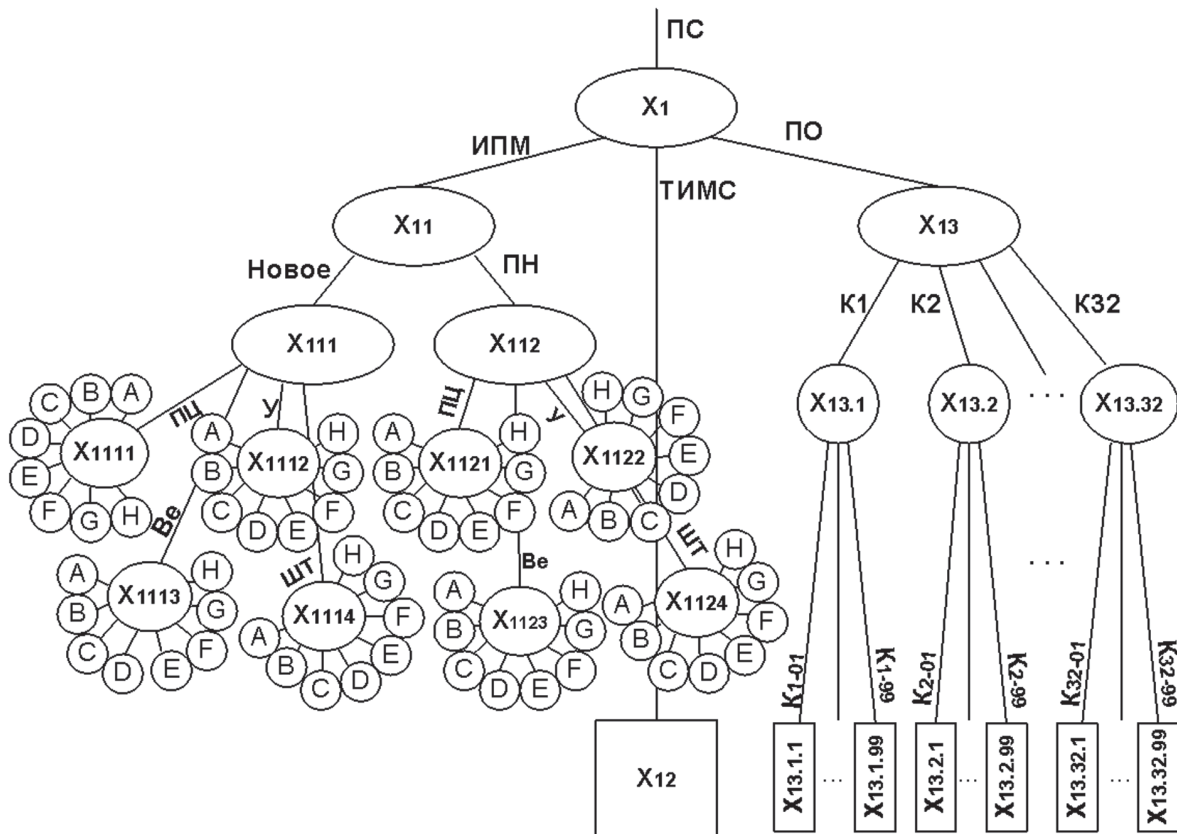


Рис. 3. Часть системы онтологий патентно-конъюнктурной информации.

Ветвь X_1 «Промышленная собственность», где ПС – промышленная собственность,

ИПМ – изобретения и полезные модели, ТИМС – топографии интегральных микросхем,

ПО – промышленные образцы, Новое – новое изобретение, ПН – применение ранее известного изобретения в новом качестве или новым способом, ПЦ – процесс, Y – устройство, Be – вещество, ШТ – штамм микроорганизма, A, B, C, D, E, F, G, H – классы изобретений в зависимости от сферы применения, K1, K2, ..., K32 – классы промышленных образцов в зависимости от сферы применения, K1-01, ..., K1-99, K2-01, ..., K2-99, ..., K32-01, ..., K32-99 – подклассы промышленных образцов

X_{1112}^E , X_{1112}^F , X_{1112}^G , X_{1112}^H – данные не относятся к заданным классам изобретений.

Выводы

В работе были рассмотрены существующие на сегодняшний момент технологии интеллектуальной обработки данных, и выбран для применения в нашей предметной области универсальный математический аппарат, основанный на алгебре конечных предикатов.

На языке АКП могут быть описаны любые конечные отношения, поэтому для нас это наиболее подходящий математический аппарат, т.к. позволяет легко обнаружить и извлечь из входного корпуса термины и их свойства, отображать многоместные отношения, связывающее текстовую ПКИ.

Была построена модель, представляющая ПКД на языке АКП, которая каждый новый признак разбивает на непересекаемые классы эквивалентности. Она является полной, несократимой и непротиворечивой. Часть данной модели была приведена в работе.

Путь к каждому предикату можно описать формулами, примеры которых приведены в работе.

Список литературы: 1. Оперативная аналитическая обработка данных: концепции и технологии [Электронный ресурс] / Ивановский гос. энергетический ун-т. – Режим доступа URL: http://citforum.ru/seminars/cis99/sch_03.shtml – 2009. – Загл. с экрана. 2. *Шапот, М.* Интеллектуальный анализ данных в системах поддержки принятия решений [Текст] / М. Шапот // Журн. открытые системы. – 2008. – №1. С. 30-35. 3. *Гаврилова, Т.А.* Базы знаний интеллектуальных систем [Текст]: учеб. / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб: Питер, 2000. – 384 с. 4. *Загорюцкий, И.М.* Выбор алгоритма обучения в системах приобретения знаний из данных [Текст]: материалы 12-ой национал. конф. по искусственному интеллекту с междунар. участием (КИИ 2010), – М.: Физматлит, 2005. – Т. 1. – С. 131-135. 5. *Калинина, Е.А.* Применение технологии Data Mining для автоматизированного построения баз знаний интегрированных экспертных систем [Текст] / Е.А. Калинина, Г.В. Рыбина.: материалы 8-ой национал. конф. по искусственному интеллекту с междунар. участием (КИИ 2002), – М.: Физматлит, 2002. – Т. 1. – С. 119-127. 6. *Чубукова, И.А.* Data Mining [Текст] / И.А. Чубукова. – М.: БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий – ИНТУИТ.ру, 2008. – 384 с. 7. *Корлякова, М.О.* Многоуровневая экспертная система на основе обобщения примеров по признакам [Текст]: материалы 7-ой национал. конф.

по искусственному интеллекту с междунар. участием (КИИ 2000), – М.: Физматлит, 2000. – Т. 1. – С.103-112. 8. *Туманов, В.Е.* Хранилища данных: Жизненный цикл разработки. [Текст] / В.Е. Туманов // Машиностроитель. – 2005. – № 8. – С. 22-30. 9. *Король, О.И.* Технології побудови систем інтелектуальної обробки патентно-кон'юнктурної інформації [Текст] / О.И. Король // Вісник Херсонського нац. тех. ун-ту. – Херсон. – 2011. – №2 (41). – С. 163-165. 10. *Бондаренко, М.Ф.* Теория интеллекта [Текст]: учеб. / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнаренко. – Харьков: Компания СМІТ, 2006. – 576 с. 11. *Шаронова, Н.В.* Автоматизированные информационные библиотечные системы: задачи обработки информации [Текст]: монография, Нар. Укр. Акад. / Н.В. Шаронова, Н.Ф. Хайрова; [Каф. информац. технологий и документоведения]. – Х., 2003. – 120 с. 12. *Gomez-Perez, A.* Ontological Engineering: what are ontologies and how can we build them [Текст] / O. Corcho, M. Fernandez-Lopez, A. Gomez-Perez, // In Cardoso (ed) Semantic Web: Theory, Tools and Applications. – IDEA Group. – 2007. – Pages 44-70. 13. *Suárez-Figueroa, A.* How to write and use the Ontology Requirements Specification Document [Текст] / M.C. Su11 ptrez-Figueroa, A. Gómez-Pérez, Boris Villazón-Terrazas // Proceedings of the 8th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2009). – ISBN: 978-3-642-05150-0. LNCS 5871. – Volume: Part II. – 2009. – Pages: 966-982.

Поступила в редколлегию 17.01.2012

УДК 347.77

Інтелектуальна обробка даних при формуванні патентно-кон'юнктурних баз знань / О.И. Король, Н.В. Шаронова // Біоніка інтелекту : наук.-техн. журнал. – 2012. – № 1 (78). – С. 12-16.

В статті розглянуті існуючі методи інтелектуальної обробки інформації. Підтверджено, що найкращим є математичний апарат, заснований на алгебрі кінцевих предикатів. Спроба створити модель інтелектуальної системи, що описує закономірності текстової патентно-кон'юнктурної інформації, представлена на природній мові.

Л. 3. Бібліогр.: 13 найм.

UDK 347.77

Intellectual data treatment in the formation of patent conjunctural knowledge bases / O.I. Korol, N.V. Sharonova // Bionics of Intelligense: Sci. Mag. – 2012. – № 1 (78). – P. 12-16.

The article reviewed existing methods of intellectual treatment. The best is the mathematical instrument based on the algebra of final predicates algebra. Trying to create a model of intellectual system oriented to description of regularities in textural patent conjunctural information presented in its original language.

Fig. 3. Ref.: 13 items.



Э.Г. Петров¹, Е.В. Губаренко²

¹ ХНУРЕ, м. Харків, Україна, ST@kture.kharkov.ua;

² ХНУРЕ, м. Харків, Україна, vergeley@mail.ru

ЦЕЛИ И УСТОЙЧИВОСТЬ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМ ПРИ РЕАЛИЗАЦИИ КОНЦЕПЦИИ УСТОЙЧИВОГО РАЗВИТИЯ

Проведен системный анализ особенности целеобразования социально-экономических систем (СЭС). Описан процесс формирования общих целей системы, а также процедуры усиления (пересечения) и расширения (объединения) при обобщении индивидуальных целей элементов СЭС. Установлена прямая зависимость уровня потребляемых ресурсов и уровня разнородности (разнообразия) индивидуальных целей. Предложена модель устойчивости, опирающаяся на следующие параметры: обобщенное пространство целей, доступные системе ресурсы и удовлетворенность индивидов (элементов системы). Предложен подход к определению количественной оценки устойчивости СЭС.

СОЦИАЛЬНО-ЭКОНОМИЧЕСКАЯ СИСТЕМА, ЦЕЛЕОБРАЗОВАНИЯ, ИНДИВИДУАЛЬНЫЕ ЦЕЛИ, УСТОЙЧИВОСТЬ, УСИЛЕНИЕ И РАСШИРЕНИЕ ЦЕЛЕЙ, РЕСУРСЫ, ЭФФЕКТИВНЫЕ ЦЕЛИ

Введение

Социально-экономические системы (СЭС) являются сложными искусственными активными системами с эндогенным целеобразованием [1]. Перечисленные особенности обусловлены тем, что системообразующим элементом СЭС выступает социальный индивид (социальная группа), целью которого является максимизация степени удовлетворения личных потребностей, а СЭС являются инструментом достижения указанной цели путем обмена личного живого труда на универсальный ресурс (деньги) с последующей трансформацией его в личные потребности [2]. Таким образом, социальный индивид является потребителем целевого эффекта СЭС, а также ее основным системообразующим ресурсом.

В отличие от пассивных (экзогенных) систем, для которых задача формирования целей является внешней, а управление их достижением заключается в определении оптимальной траектории и минимизации затрат на компенсацию возмущающих воздействий, для СЭС, как активных систем, центральной проблемой является формирование эффективной цели.

Эта задача многоаспектна и включает в себя необходимость согласования личных индивидуальных целей членов СЭС и определение компромиссной общей цели [3]. В условиях ограниченности большинства внешних ресурсов возникает проблема согласования целей различных СЭС, которые конкурируют за ресурсы, рынки сбыта и т.д.

Указанные задачи являются достаточно сложными и в настоящее время плохо исследованными. Достаточно отметить, что на процесс формирования индивидуальных целей оказывают влияние социальные (семья, культурные особенности, социальные роли, общение, средства массовой информации (СМИ), общество в целом),

территориальные (климат, флора и фауна), экономические (финансовое положение в обществе, покупательская способность) и государственные факторы (государство через образовательную систему имеет беспрецедентную возможность корректировать индивидуальные цели).

С другой стороны, для СЭС очень важно сформировать адекватную эффективную цель, которая не только бы учитывала все особенности индивидуальных целей и была достаточно конкретизирована, но и могла быть достигнута в рамках тех ресурсов, которыми располагает СЭС. Доступ к ресурсам, необходимым для реализации целей, для современных СЭС становится принципиально важным. На данный момент свободных легкодоступных ресурсов не осталось, и расширение потребления одной системой означает снижение потребления конкурирующих СЭС. Очевидно, что подобная ситуация не может саморегулироваться и нуждается в государственном и межгосударственном контроле. В противном случае в определенных условиях данные конфликты приводят к локальным или региональным кризисным ситуациям, что в свою очередь вызывает экономические, социальные и экологические кризисы, вплоть до вооруженных столкновений

Переход к концепции устойчивого развития [4,5] предусматривает необходимость развития теории и практики управления, обеспечивающего устойчивость и повышение эффективности функционирования СЭС на всех уровнях. Сама же концепция устойчивого развития общества призвана найти альтернативный путь разрешения конфликтных ситуаций с сохранением поступательного развития всех СЭС путем установления жестких международных квот, при этом сохраняя баланс потребления ресурсов между конкурирующими СЭС [6].

Цель авторов статьи заключается в анализе особенностей целеобразования для СЭС и разработке методологии формирования согласованных целей, ориентированных на повышение уровня удовлетворенности индивидов в условиях ограниченности доступных ресурсов, разработке метода и моделей формирования управления устойчивого развития СЭС.

1. Особенности процесса целеобразования и реализации целей

Процесс целеобразования в СЭС может быть представлен следующим образом (рис. 1). Индивиды, которые являются системообразующими элементами системы, формируют личные цели. Затем формируется эффективная цель, на реализацию которой СЭС расходует ресурсы, получая при этом некий результат (благо); далее результат распределяется между индивидами и формируется индивидуальная удовлетворенность (Y_L).

Как видно из рис. 1, устойчивость СЭС может быть нарушена при формировании неверной эффективной цели, при нехватке ресурсов для достижения эффективной цели и при непропорциональном распределении благ.

В момент, когда индивид становится элементом соответствующей СЭС, **личные цели** (C_L) могут формироваться абсолютно независимо от СЭС, могут согласовываться с уже существующей эффективной целью СЭС или же полностью дублировать эффективную цель

Эффективная цель (C_E) формируется на основании индивидуальных целей, но не является их точным отображением. Личная цель (C_L) представляет собой кортеж (1), каждое значения которого отображает некое фиксированное положение индивида в многомерном пространстве $\{x_1, x_2, \dots, x_n\}$.

Другими словами – это список целей, которые имеют свои собственные параметры (2).

$$C_L(x) = \{x_1, x_2, \dots, x_n\}, \tag{1}$$

где n – количество фиксированных состояний индивида

$$x_i = (V, S, \Delta S(t), Z, \Delta Z(m), P), i = \overline{1, n}, \tag{2}$$

где V – важность состояния для индивида; S – величина удовлетворенности от достижения данного состояния; $\Delta S(t)$ – функция изменения S при повторном достижении этого состояния, чаще всего зависит от времени или от количества достигнутых состояний; Z – затраты на достижение данного состояния; $\Delta Z(m)$ – функция, отображающая динамику изменения затрат при повторном достижении этого состояния, чаще всего зависит от количества достигнутых состояний; P – повторяемость (или обратный ему показатель уникальность), показывает насколько однородными являются индивидуальные цели.

Эффективная цель (C_E) формируется как функция от совокупности индивидуальных целей

$$C_E = f\left(\bigcup_{i=1}^n C_{L_i}\right), \tag{3}$$

где n – количество индивидов; f – функция, устанавливающая зависимость между эффективной целью и совокупностью индивидуальных; C_E – эффективная цель СЭС; C_{L_i} – индивидуальная цель i -го элемента СЭС. При этом должно выполняться следующее ограничение: **доход от достижения эффективной цели не должен быть меньше суммы личных доходов**. В противном случае существование такой СЭС не имеет смысла.

$$D(C_E) \geq \sum_{i=1}^n D(C_{L_i}), \tag{4}$$

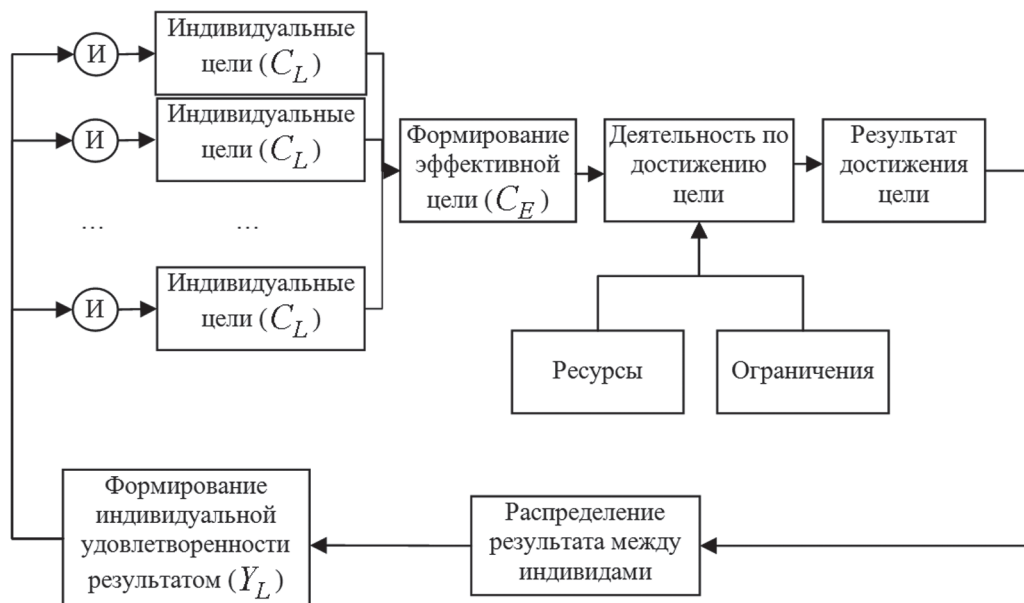


Рис. 1. Схематическое представление процесса формирования и достижения целей СЭС

где D – функция дохода; n – количество индивидов; C_E – эффективная цель СЭС; C_{L_i} – индивидуальная цель i -го элемента СЭС.

Эффективная цель формируется из пространства индивидуальных целей (3), само же пространство образуется из объединения целей каждого индивида, вне зависимости от количества и качества состояний в индивидуальных целях, всегда можно сформировать n -мерное пространство, где каждое состояние будет иметь некое положение и количественные оценки всех установленных параметров (2). При формировании эффективной цели следует выделить две процедуры: расширение (5) и усиление (6) целей.

$$C_{L_1} \cup C_{L_2} \cup \dots \cup C_{L_n}, \quad (5)$$

где n – количество индивидов.

В соответствии с условием формирования пространства целей (5), ряд целей будут накладываться друг на друга, тем самым усиливая однородность целей и важность достижения данного состояния для эффективной цели. Данную особенность можно выявить при пересечении индивидуальных целей (6). Иногда проще осуществлять попарное сравнение (фиксирование пересечение попарно) (7).

$$C_{L_1} \cap C_{L_2} \cap \dots \cap C_{L_n}, \quad (6)$$

$$C_{L_1} \cap C_{L_2}, C_{L_1} \cap C_{L_3}, C_{L_2} \cap C_{L_3}, \dots \quad (7)$$

Состояния с высокой однородностью требуют меньше затрат на реализацию потребностей для каждого отдельного индивида.

2. Формирование индивидуальных целей и общественное целеобразование для государственных СЭС

Целеобразование для общества является одним из важнейших вопросов. Как нити формируют канат, так и цели каждого индивида формируют цель общества. Можно сформулировать закономерность: чем более схожи цели (потребности) элементов системы, тем выше ее устойчивость.

Перед государством, как формой организации общества, стоит дилемма. С одной стороны, для обеспечения устойчивости государственной СЭС необходимо сформировать единообразие целей каждой группы, каждого индивида, что потребует силовых методов для подавления собственных целей индивидов, и вызовет нарастание напряженности в обществе, что снизит устойчивости СЭС. С другой – обеспечить удовлетворение каждой индивидуальной цели элементов системы нет возможности, так как количество ресурсов, необходимых для достижения целей, стремится к бесконечности.

На международном и государственном уровнях проблема целеобразования решается посредством формулирования концепции функционирования системы. Например, концепция экономического

роста и обогащения. Другими словами, люди формируют свои цели на основе критерия: получить максимальную выгоду (чаще всего в денежно выражении). Целью государства является увеличение экономических показателей (ВВП, ВНД и прочее), соответственно, целью групп (предприятий, юридических лиц) – максимизация прибыли, а индивидов – увеличение личного дохода и благосостояния.

Альтернативой этой концепции является концепция устойчивого развития, в основе которой лежит принцип разумной достаточности потребляемых благ и последовательного (устойчивого) повышения их разнообразия, уровня и качества за счет интенсификации и интеллектуализации процесса производства.

Способами привития нравственных норм поведения членов общества, ориентированных на **разумную достаточность** потребления, всегда выступали средства культуры: театральные постановки, различные мероприятия, художественная литература, средства массовой информации (СМИ), кинофильмы, мультипликация. На этом редко акцентируют внимание, но существенное влияние на эти факторы оказывают образование, научные исследования (научная литература), мнение экспертов или уважаемых личностей общества. Они формируют некие причинно-следственные связи, предлагая или расширяя варианты поведения, они создают поведенческие стереотипы (пространство целей и траектории их достижения), которые индивид мог ранее и не рассматривать как возможную альтернативу. Это не означает, что индивид осуществит выбор именно этой альтернативы, но как вариант она уже существует.

Чаще всего его используют производители для позиционирования и продвижения товаров и услуг, формируя новый спрос, новые цели среди индивидов общества, следовательно, изменяя его структуру и общую цель. Если речь идет о транснациональных корпорациях (ТНК), возникает факт воздействия чужеродной культуры и конкурирующих государств. Вот почему, обсуждаемая в последнее время проблема информационной безопасности имеет очень важное значение для сохранения устойчивости государственной СЭС.

Целеобразование – сложный, динамический процесс, который вбирает в себя желания, мечты, мотивы и предрасположенность индивидов, обобщение целей при формировании групп и общества, а также воздействие государства при организации стремлений повысить однообразие целей.

Цели индивида переменчивы, неустойчивы, быстро теряют актуальность вне зависимости, были ли они достигнуты. И все же, такие цели являются образующими элементами процесса целеобразования и формирования цели общества.

Формирование индивидуальных целей происходит под воздействием следующих факторов:

- социального положения, которое человек занимает на момент формирования цели;
- концептуальной цели государства, которая была установлена на момент формирования цели;
- влияния других индивидуальных целей, с которыми человек сталкивался до, либо в момент формирования цели;
- окружающей среды как социальной, так и экологической, т.к. ареал обитания оказывает существенное влияние на стремления, цели и потребности;
- образования и воспитания.

Общественные цели формируются как цели множества социальных групп. Индивидуальная цель формирует социальную группу, но не оказывает прямого воздействия на цели общества. Это происходит из-за того, что цели групп более устойчивы и ограничены, а для достижения цели необходимо время. Каждому индивиду, в зависимости от того, к какой группе он относится, обществом предписывается его социальная роль: цели, поведенческие стереотипы, потребности и траектории их достижения. Это не означает, что человек не имеет права установить для себя иные цели, но такой факт будет воспринят крайне негативно всем обществом, а не только одной социальной группой. Зачастую результатом является исключение человека из социальной группы. На практике это будет означать социальную изоляцию.

Формирование социальных групп и распределение социальных ролей происходит под воздействием следующих факторов:

- концептуальной цели государства, которая была установлена на момент формирования цели;
- существующих социальных групп и социальных ролей;
- индивидуальных целей, которые могут полностью изменить социальные роли, либо сформировать новую социальную группу.

Государственные цели отличаются своей консервативностью. В отличие от индивидуальных целей, которые недолговечны, государственная должна просуществовать не одно поколение. Частая смена приведет к потере устойчивости. Вот почему, когда говорят о государственной цели, употребляют понятие концептуальная цель. Данная цель, хотя и формируется на основе целей общественных групп и косвенно индивидуальных целей, оказывает весьма существенное давление на выбор целей индивидов, на формирование социальных групп и установление требований для социальных ролей. По сути, концептуальная цель – это аккумуляция всех целей общества, всех стремлений индивидов, прошлых и будущих. Она определяет дальнейшее развитие культуры, науки, взаимоотношений.

Достаточно часто общественные группы дублируют друг друга и по целям, и по социальным ролям. Кроме того, их недолговечность усложняет формирование общей цели и распознавание перспективных направлений развития, поэтому государство искусственно объединяет социальные группы в социальные институты или общественные институты. Следует подчеркнуть, что социальные институты – это искусственное обобщение социальных групп, их основная функция заключается в сборе и аккумулировании информации.

В этом заключается дилемма: разнообразие индивидуальных и групповых целей обеспечивает высокую приспособляемость и живучесть по отношению к случайным возмущениям, но обеспечивает низкую сопротивляемость целенаправленному вредоносному воздействию других систем. При столкновении интересов верх одерживает система с более высоким единообразием целей.

3. Формирование количественной оценки устойчивости СЭС

Задание допустимого пространства состояний СЭС означает, что по всем критическим характеристикам СЭС ($\langle h_{i,j} \rangle$, $i = \overline{1,3}$ – номер элемента СЭС, $j = \overline{1,n}$ – номер локальной характеристики i -го элемента) заданы ограничения вида:

$$\begin{aligned} h_{i,j}^H &\leq h_{i,j} \leq h_{i,j}^B; \\ h_{i,j} &= h_{i,j}^D, \end{aligned} \quad (9)$$

где H , B , D – соответственно индексы нижнего, верхнего и допустимого значений локальных характеристик состояния СЭС.

Каждое из неравенств или уравнений (9) определяет локальную линейную или нелинейную границу, а их совокупность – некоторую область в n -мерном пространстве характеристик. Это не исключает возможности объединения нескольких характеристик в функционально связанные группы. В такой интерпретации состояние каждой конкретной СЭС может быть представлено в виде многомерной точки в допустимой области функционирования. Тогда кратчайшее расстояние от точки, представляющей состояние системы, до границы допустимой области может быть интерпретировано как количественный показатель устойчивости системы. Каждая локальная характеристика определяет какую-либо границу допустимого пространства, а их функционально связанная группа – какой-либо фрагмент границы допустимой области. Тогда расстояние до этого фрагмента границы от точки фактического состояния системы характеризует социальную, экономическую, экологическую или любую другую устойчивость системы.

С учетом сказанного, характеристика запаса устойчивости СЭС по любой локальной

характеристике будет определяться в натуральных показателях по формуле

$$\Delta h_{ij} = h_{ij}^F - h_{ij}^H, \quad (10)$$

где H, F – соответственно граничное и фактическое значение характеристики. В относительных показателях –

$$U_{ij} = \frac{\Delta h_{ij}}{h_{ij}^B - h_{ij}^H} \cdot 100\%. \quad (11)$$

Более сложно формируется оценка устойчивости по функционально связанной группе показателей. В этом случае предлагается оценивать значение устойчивости по безразмерному показателю вида

$$U_i = \sum_{j=1}^n a_{ij} h_{ij}^N, \quad (12)$$

где a_{ij} – безразмерные показатели относительной важности i -й характеристики при условии, что $0 \leq a_{ij} \leq 1, \sum_{j=1}^n a_{ij} = 1$; h_{ij}^N – нормализованное значение i -й локальной характеристики СЭС.

Для нормализованных локальных характеристик должны выполняться следующие условия [7]: безразмерность; ограниченный, одинаковый для всех интервал возможных значений $[0,1]$; инвариантность к направлению доминирования (min, max); неотрицательность.

Удовлетворение всех перечисленных требований обеспечивается моделью нормализации вида:

$$h_{ij}^N = \frac{h_{ij}^F - h_{ij}^H}{h_{ij}^B - h_{ij}^H} = \frac{\Delta h_{ij}}{h_{ij}^B - h_{ij}^H}. \quad (13)$$

Значение a_{ij} (12) определяют экспертным путем или методом компараторной идентификации [8]. Показатель устойчивости U изменяется в интервале от $[0,1]$ и непосредственно характеризует запас устойчивости СЭС по любой группе параметров или в целом.

Помимо описанного подхода, устойчивость СЭС более высокого уровня можно оценивать, опираясь на значения устойчивости элементов, которые входят в состав такой СЭС. Далеко не всегда есть возможность достоверно и в полном объеме оценить характеристики сложных объектов, и чем масштабней СЭС, тем выше вероятность возникновения и накопления погрешности в определении как значений характеристик, так и их весовых коэффициентов

Другая особенность критерия (12) заключается в возможности полной компенсации одних показателей другими. Это означает, что если по некоторым характеристикам запас устойчивости $U_{ij} = 0$, общая оценка (12) будет положительной. Во избежание такой ситуации предлагается использовать альтернативную мультипликативную оценку вида

$$\bar{U}_i = \prod_{j=1}^n U_{ij}, \quad (14)$$

однако необходимо учитывать, что мультипликативные оценки в принципе не позволяют учесть различную значимость локальных характеристик. Этот недостаток можно преодолеть, применив аддитивно-мультипликативную оценку устойчивости сложных СЭС вида

$$\bar{U}_0 = \sum_{i=1}^m b_i \bar{U}_i. \quad (15)$$

где b_i – весовые коэффициенты, которые характеризует степень влияния устойчивости i -го элемента на устойчивость вышестоящей СЭС. При этом должны выполняться следующие требования: $0 \leq b_i \leq 1, \sum_{i=1}^m b_i = 1$, где n – число элементов СЭС.

Выводы

Концепция устойчивого развития, которая обобщает показатели социальной, экономической и экологической сферы, направлена на устранение гипертрофированного перевеса приоритетов развития в сторону экономических факторов, что вылилось в концепцию экономического роста. Результатом стало исчерпывание практически всего запаса устойчивости мировой СЭС. Легкодоступные ископаемые ресурсы на грани истощения, восстанавливающиеся ресурсы (атмосферный воздух, грунтовые воды, плодородные почвы) резко снижают свои характеристики, но все же главной потерей за этот период являются потери в социальной сфере. Основным показателем благополучия граждан является их здоровье. Индекс здоровья населения падает даже в экономически развитых странах. Отмечаются рост аллергических реакций, хронических и профессиональных заболеваний, приобретенной и врожденной инвалидности и прочее. На фоне резкого снижения доли экономически активного населения в общей численности необходимо отметить снижение и качественных характеристик: уровня образования, квалификации, преимущества.

Реализация концепции устойчивого развития связана с планомерной, кропотливой воспитательной образовательной работой по изменению общественных, групповых и личных социальных приоритетов и предпочтений.

Список литературы: 1. Петров, Э.Г. Роль, задачи и методы государственного управления при реализации концепции устойчивого развития [Текст] / Э.Г. Петров, Е.В. Губаренко // Бионика интеллекта. – 2011. – №3(77). – С. 60-64. 2. Петров, Э.Г. Методы и инструментальные средства систем поддержки принятия решений при организационном управлении социально-экономическими системами [Текст] / Э.Г. Петров, Е.В. Губаренко // Бионика интеллекта. – 2010. – №3(74). – С. 26-36. 3. Глушков,

В.М. Введение в АСУ [Текст] / В.М. Глушков. — изд.2-е, исправленное и дополненное. — К.: Техника, 1974. — 320 с.

4. Згуровский, М.З. Роль инженерной науки и практики в устойчивом развитии общества [Текст] / М.З. Згуровский, Г.А. Статюха // Системні дослідження та інформаційні технології. — 2007. — №1. — С. 19-38.

5. Згуровский, М.З. Системный подход к оценке и управлению устойчивого развития общества [Текст] / М.З. Згуровский, Г.А. Статюха // Системні дослідження та інформаційні технології. — 2007. — №3. — С. 7 — 27.

6. Петров, Э.Г. Проблемы и перспективы международного управления ресурсами на основе квотирования при реализации концепции устойчивого развития мировой социально-экономической системы [Текст] / Э.Г. Петров, Е.В. Губаренко // Бионика интеллекта. — 2011. — №2(76). — С.114-122.

7. Губаренко, Е.В. Методология формирования количественных оценок уровня развития региона [Текст] / Е.В. Губаренко, Н.В. Подмогильный // Вестник ХНТУ. — 2010. — №2(38). — С.76-80.

8. Овезгельдыев, А.О. Синтез и идентификация моделей многофакторного оценивания и оптимизации [Текст] / А.О. Овезгельдыев, Э.Г. Петров, К.Э. Петров. — Киев: «Наукова думка». — 2002. — 164с.

Поступила в редколлегию 28.11.2011

УДК 519.81

Цілі та стійкість соціально-економічних систем при реалізації концепції сталого розвитку / Е.Г. Петров, Е.В. Губаренко // Біоніка інтелекту: наук.-техн. журнал. — 2012. — № 1 (78). — С. 17-22.

Приведена схематична послідовність появи індивідуальних, групових суспільних цілей, а також можливі шляхи їх досягнення. Наводиться аналіз процесу досягнення цілей в умовах обмежених ресурсів. Наводиться метод розрахунку стійкості за об'ємом використання ресурсів та рівня сприйняття індивідом результатів функціонування соціально-економічної системи. Наведена модель формування кількісних оцінок стійкості системи.

Л. 1. Бібліогр.: 8 найм.

UDK 519.81

The purposes and stability of social and economic systems at realization of the concept of the sustainable development / E.G. Petrov, E.V. Gubarenko // Bionics of Intelligence: Sci. Mag. — 2012. — № 1 (78). — P. 17-22.

The schematic sequence of occurrence of the individual, group and public purposes, and as possible ways of their generalization is presented. Process of achievement of the purposes in the conditions of the limited resources is analysed. The method of definition of stability on volume of consumed resources and level of satisfaction of individuals from achievement of overall aims of social and economic systems is offered. The model of formation of quantitative estimations of stability of systems is offered.

Fig. 1. Ref.: 8 items.

УДК 631.432:626.86:



В.Й. Пастушенко, С.К. Матус, А.М. Стеценко

Національний університет водного господарства
та природокористування, м. Рівне

vteren@gmail.com, matus_sv@ukr.net, s_anastasia@ukr.net

ІДЕНТИФІКАЦІЯ КОЕФІЦІЄНТА ВОЛОГОПРОВІДНОСТІ ГРУНТУ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

Проведено розрахунок коефіцієнта вологопровідності ґрунту за експериментальними аналітичними формулами та за допомогою нейромереж різної архітектури, здійснено порівняльний аналіз розрахунків.

НЕЙРОННА МЕРЕЖА, КОЕФІЦІЄНТ ВОЛОГОПРОВІДНОСТІ ГРУНТУ, ВОЛОГІСТЬ, ВСМОК-
ТУЮЧИЙ ТИСК

Вступ

При застосуванні диференційного рівняння вологопереносу виникає необхідність визначення коефіцієнтів і функцій, які є параметрами цього рівняння та описують фізичні процеси переносу вологи [1]. До таких функцій відносимо функцію водоутримання або основну гідрофізичну характеристику (ОГХ) – залежність всмоктуючого тиску ґрунту від його вологості $\psi(W)$ – і функцію вологопровідності ненасичених ґрунтів. Говорячи про функцію вологопровідності, розрізняють характеристику вологопровідності, що пов'язує коефіцієнт вологопровідності і потенціал ґрунтової вологи $k(\psi)$ та характеристику, яка пов'язує коефіцієнт вологопровідності ґрунту і його вологість $k(W)$ [2].

1. Аналіз досліджень і публікацій

Здатність ґрунту проводити воду є важливим показником, необхідним для різних меліоративних розрахунків. Вивченням кількісних закономірностей переміщення вологи в ґрунті займалися Алпат'єв А.М., Воронін А.Д., Глобус А.М., Муромцев М.О., Роде А.А., Судніцин І.І. та інші. Інформація про вологопровідність може бути використана для математичного (кількісного) аналізу різних випадків переміщення ґрунтової вологи, що відбувається в природних умовах. До таких випадків переміщення слід віднести фільтрацію води через ґрунт, надходження вологи від рівня ґрунтових вод до поверхні, поглинання вологи ґрунтом, надходження ґрунтової вологи до коріння рослин.

Універсальною функцією, що відображає вплив усіх факторів, які можуть впливати на енергетичний стан води в ґрунті, є потенціал вологи. Як відомо, вода переміщується завжди від більшого потенціалу до меншого. Діючою силою при цьому є градієнт вільної енергії або потенціалу ґрунтової вологи, що відображає сили, які викликають ізотермічний потік вологи. В ізотермічних умовах складовими потенціалу вологи є: капілярний тиск, осмотичні і гравітаційні сили. Внаслідок цього,

гідралічна провідність ненасичених вологою ґрунтів (капілярна провідність k , м/добу) характеризується не одним значенням (як при насиченому вологоперенесенні), а функцією від всмоктуючого тиску $k = k(\psi)$ або від вологості ґрунту $k = k(W)$.

З точки зору вологопровідності ґрунтовий профіль ділиться на дві частини: знизу – область повного насичення, де рух вільної вологи відбувається під впливом сили тяжіння, згідно з законами фільтрації, а зверху – область з рухом вологи у ненасиченому ґрунті.

Найбільш розповсюдженими методами визначення $k(\psi)$ є експериментальні методи, які базуються на аналізі даних про рух вологи: методи стаціонарного і нестаціонарного потоків [3]. Реалізувати стаціонарний режим потоку не завжди легко з технічної точки зору, а крім того, такі дослідження вимагають значних затрат часу. Нестационарні методи потребують менших затрат часу експериментатора у порівнянні зі стаціонарними, але достовірність і точність отриманих даних залежить від того, наскільки умови дослідження відповідали цілому ряду теоретичних припущень.

Складна природа залежності ґрунту з водою у вигляді “вологість – потенціал – вологопровідність” найчастіше представляється у вигляді емпіричних формул та графіків. Запропоновано багато емпіричних апроксимаційних формул для функції вологопровідності, які з певною точністю відтворюють цю функцію на окремих інтервалах [3]. Найбільш відомими є формули Ван-Генухтена (1980), Муалема (1976), Гарднера, Уїнстона, К.П. Лудніна, С.Ф. Авер'янова, В.А. Бодрова, А.І. Будаговського.

Спеціальний аналіз цих формул і області їх використання показав, що для мінеральних і торфових ґрунтів використовуються наступні формули:

Формула Ван-Генухтена:

$$k(\psi) = \frac{k_f \cdot \left\{ 1 - (\alpha \cdot \psi)^{m \cdot n} \cdot \left[1 + (\alpha \cdot \psi)^n \right]^{-m} \right\}^2}{\left[1 + (\alpha \cdot \psi)^n \right]^{m \cdot l}}, \quad (1)$$

$$k(W) = k_f \cdot \sqrt{\frac{W - W_r}{W_s - W_r}} \cdot \left[1 - \left(1 - \left(\frac{W - W_r}{W_s - W_r} \right)^{1/n} \right) \right]^2, \quad (2)$$

де m, n, α – параметри апроксимації експериментальних даних ОГХ: $m = 1 - \frac{1}{n}$, $0 < m < 1$; n, α – два основних параметри апроксимації реальних даних ОГХ, які мають фізичний зміст, n – безрозмірна величина, що характеризує кут нахилу кривої ОГХ, її крутизну ($n = 1 \dots 4$); α – величина, яка обернена значенню тиску повітря і тому має розмірність $1/m$; k_f – коефіцієнт фільтрації (м/добу); ψ – капілярно-сорбційний потенціал (м); W – об’ємна вологість, яка відповідає даному потенціалу (%); W_r – мінімально можлива вологість (%); W_s – повна вологостемність (%).

Для дерново-підзолистих ґрунтів: $k_f = 0,15 \frac{m}{\text{добу}}$.

Для торфових ґрунтів: $k_f = 0,35 \frac{m}{\text{добу}}$.

Формула Уїнстона:

$$k(\psi) = k_f \cdot \left(\frac{1}{\psi} \right)^m, \quad (3)$$

де ψ – всмоктуючий тиск (потенціал) ґрунту, k_f – коефіцієнт фільтрації; m – емпіричний коефіцієнт ($m = 2$ для глинистих та дерново-підзолистих ґрунтів, $m = 2,5$ для торфових ґрунтів, $m = 4$ для піску).

Наближені вирази для попередніх розрахунків коефіцієнта вологостемності [4, 96-119 с.]:

$$K_B = 0,001 \cdot e^{10 \cdot \eta} \text{ – для піску і верхніх шарів торфу,} \quad (4)$$

$$K_B = 0,001 \cdot e^{7 \cdot \eta} \text{ – для підорного шару торфу,} \quad (5)$$

$$\eta = \frac{\theta - \theta_0}{\theta_{\text{ПВ}} - \theta_0} \text{ – коефіцієнт активної вологості.} \quad (6)$$

θ – об’ємна вологість (%), $\theta_{\text{ПВ}}$ – повна вологостемність (%), θ_0 – мінімально можлива вологість (%).

Для піску і верхніх шарів торфу: $\theta_{\text{ПВ}} = 83,8\%$, $\theta_0 = 42\%$. Для підорного шару торфу: $\theta_{\text{ПВ}} = 81,5\%$, $\theta_0 = 51\%$.

Формула Авер’янова С.Ф.

$$k(\theta) = k_f \cdot \left(\frac{\theta - \theta_0}{\theta_{\text{ПВ}} - \theta_0} \right)^{3,5}, \quad (7)$$

де k_f – коефіцієнт фільтрації м/добу; θ – об’ємна вологість, %; $\theta_{\text{ПВ}}$ – повна вологостемність, %; θ_0 – мінімально можлива вологість, %.

Результати розрахунку коефіцієнта вологостемності для дерново-підзолистих та торфових ґрунтів за вищенаведеними формулами представлені у табл. 1 і 2. Найвищу точність розрахунків у порівнянні з експериментальними даними показала формула Ван-Генухтена, графіки розрахунків за якою при зміні вологості ґрунту в часі наведені на рис. 1 і 2.

Таблиця 1

Результати розрахунку коефіцієнта вологостемності для дерново-підзолистих ґрунтів на двох вибірках даних

№	Формула	Середньо-квадратичне відхилення	
		вибірка 1	вибірка 2
1	Формула Ван-Генухтена	0,005150023	0,004769792
2	Формула Уїнстона	0,98984064	0,09873856
3	Формула Авер’янова	0,406349011	0,237242401

Таблиця 2

Результати розрахунку коефіцієнта вологостемності для торфових ґрунтів на двох вибірках даних

№	Формула	Середньо-квадратичне відхилення	
		вибірка 1	вибірка 2
1	Формула Ван-Генухтена	0,02542259	0,02664897
2	Формула Уїнстона	0,046249477	0,0402224
3	Формула Лудніна, Гончарука, Пашкевича	0,072336063	0,089954
4	Формула Авер’янова	2,247555649	2,62011920

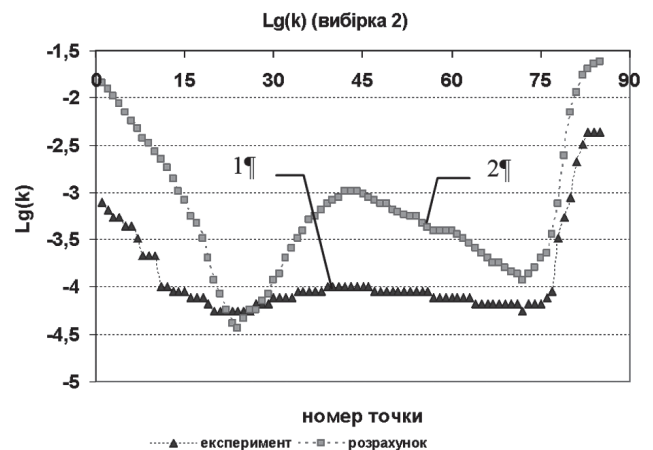


Рис. 1. Результати розрахунку коефіцієнта вологостемності дерново-підзолистої ґрунту за формулою Ван-Генухтена на вибірці даних №2: 1 – експериментальні дані, 2 – результати розрахунків

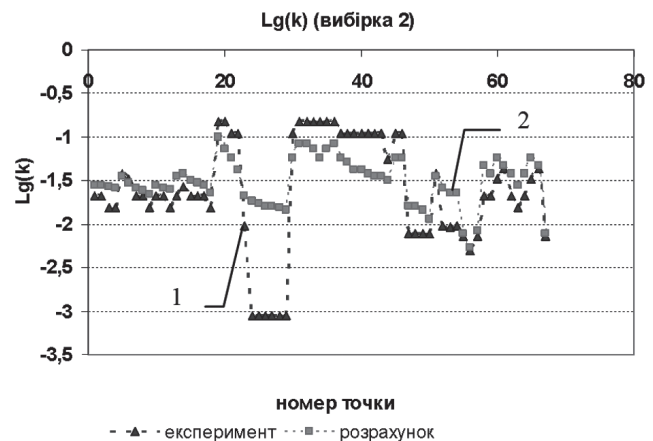


Рис. 2. Результати розрахунку коефіцієнта вологостемності торфового ґрунту за формулою Ван-Генухтена на вибірці даних №2: 1 – експериментальні дані, 2 – результати розрахунків

2. Постановка завдання

У роботі поставлено задачу розробки альтернативної методики розрахунку коефіцієнта вологопровідності ґрунту на базі нейронних мереж, аналізі роботи нейромереж різної архітектури, порівняння їх роботи з результатами визначення коефіцієнта вологопровідності за аналітичними формулами та з експериментальними даними.

3. Розрахунок коефіцієнта вологопровідності ґрунту за допомогою нейронних мереж

Нейронні мережі (НМ) – це обчислювальні структури, які моделюють прості біологічні процеси, що асоціюються з процесами мозку людини. НМ представляють собою системи, здатні до адаптації та навчання шляхом аналізу позитивних і негативних впливів [5]. Основним джерелом знань про об’єкт моделювання (керований модуль меліоративної системи) є архів даних про зміну вхідних і вихідних параметрів. На базі архіву з врахуванням обраної архітектури нейронної мережі створювався набір тренувальних шаблонів $\{X_i, Y_i\}_j, j=1, J$, кожен з яких включав вектор відомих вхідних сигналів X_{ij} і відповідний йому вихід мережі Y_{ij} . Навчання вибраного варіанта моделі зводиться до налаштування параметрів нейромережі (параметрів передаточних функцій нейронів, вагових коефіцієнтів) так, щоб вона найкраще запам’ятовувала набір тренувальних шаблонів. НМ служить своєрідним індикатором правильності вибору нами вхідних параметрів. При включенні до вхідного вектора параметра, який тісно пов’язаний з модельованим процесом, похибка розрахунку зменшиться.

На коефіцієнт вологопровідності ґрунту k впливають потенціал вологи (всмоктуючий тиск) ψ та вологість ґрунту W . Для розрахунку k створено нейронну мережу виду:

$$k = NN(W_1, W_2, \psi_1, \psi_2), \quad (8)$$

де вхідними параметрами є: W_1 і W_2 – значення вологості на середині двох суміжних шарів ґрунту; ψ_1 і ψ_2 – значення всмоктуючого тиску ґрунту у цих шарах; NN – перетворення, яке здійснюється нейронної мережею.

Побудова, навчання і тестування нейронних мереж проводилося у нейроемуляторі NeuroPro та редакторі Neural Network Toolbox (редактор для роботи з нейронними мережами) системи MatLab. При цьому використовувались мережі різної архітектури: багатошарова (NeuroPro), узагальнено-регресійна мережа, лінійна, мережа з радіальними базисними елементами з нульовою помилкою на навчальній вибірці даних, мережа з радіальними базисними елементами, “класична” багатошарова нейронна мережа з навчанням за методом зворотного поширення помилки, мережа Елмана, багатошарова нейронна мережа зі зворотним поширенням

помилки – каскадна НМ, перцептрон, ймовірнісна нейронна мережа (MatLab).

Результати розрахунку коефіцієнта вологопровідності для дерново-підзолистих та торфових ґрунтів представлені в табл. 3 і 4. При цьому вибірка даних №1 використовувалася як навчальна, а вибірка №2 – як тестувальна.

Таблиця 3

Результати розрахунків коефіцієнта вологопровідності для дерново-підзолистих ґрунтів на базі нейронних мереж (шари ґрунту 2,5-7,5 см)

Кількість точок при навчанні і тестуванні: 85

№	Нейромережа	Середньо-квадратичне відхилення	
		навчання	тестування
1	Багатошарова НМ (нейроемулятор NeuroPro): к-сть шарів – 3, к-сть нейронів в 1-му шарі – 10, передаточні функції – сигмоїдні	4,786E-06	6,8E-4
2	НМ з радіально- базисними елементами: к-сть нейронів у 1-му шарі – 2, передаточні ф-ції – радіально-базисні, к-сть нейронів у 2-му шарі – 1, передаточна ф-ція – лінійна	4,4679e-005	1,0676e-004
3	Узагальнено-регресійна мережа: к-сть нейронів у 1-му шарі – 85, передаточні ф-ції – радіально-базисні, к-сть нейронів у 2-му шарі – 1, передаточна ф-ція – лінійна	1,0183e-004	4,7783e-004
4	НМ з радіально- базисними елементами з нульовою помилкою на навчальній вибірці даних: к-сть нейронів у 1-му шарі – 85, передаточні ф-ції – радіально-базисні, к-сть нейронів у 2-му шарі – 1, передаточна ф-ція – лінійна	2,1367e-016	8,7885e-004
5	Перцептрон: к-сть нейронів – 1, одинична ф-ція активації з жорстким обмеженням	0,0021	9,4803e-004
6	Лінійна нейромережа: к-сть нейронів – 1, передаточна ф-ція – лінійна	0,0021	9,4803e-004
7	Багатошарова НМ зі зворотним поширенням помилки – каскадна НМ: к-сть шарів – 10, к-сть нейронів у шарах 1-9 – 10, к-сть нейронів у шарі 10 – 1, передаточні функції – сигмоїдні	0,5E-3	1,1E-3

8	Мережа Елмана: к-сть шарів – 10, к-сть нейронів у шарах 1-9 – 10, к-сть нейронів у шарі 10 – 1, передаточні функції – сигмоїдні	0,3123	0,0436
9	“Класична” багатошарова НМ з навчанням за методом зворотного поширення помилки: к-сть шарів – 10, к-сть нейронів у шарах 1-9 – 10, к-сть нейронів у шарі 10 – 1, передаточні функції – сигмоїдні	0,3091	0,1288
10	Ймовірнісна НМ: к-сть нейронів у 1-му шарі – 85, передаточні ф-ції – радіально-базисні, к-сть нейронів у 2-му шарі – 1, передаточна ф-ція – ф-ція конкуренції	0,9995	0,9996

Таблиця 4

Результати розрахунків коефіцієнта вологопровідності для торфових ґрунтів на базі нейронних мереж (шари ґрунту 12,5-30 см)

Кількість точок при навчанні – 67, при тестуванні – 62

№	Нейромережа_anastasia@ukr.net	Середньо-квадратичне відхилення	
		навчання	тестування
1	Багатошарова НМ (нейроемулятор NeuroPro): к-сть шарів – 3, к-сть нейронів в 1-му шарі – 10, передаточні функції – сигмоїдні	1,21073E-4	5,32E-3
2	Узагальнено-регресійна мережа: мережа: к-сть нейронів у 1-му шарі – 67, передаточні ф-ції – радіально-базисні, к-сть нейронів у 2-му шарі – 1, передаточна ф-ція – лінійна	0,0034	0,0057
3	Лінійна нейромережа: к-сть нейронів – 1, передаточна ф-ція – лінійна	0,0071	0,0075
4	НМ з радіально- базисними елементами з нульовою помилкою на навчальній вибірці даних: к-сть нейронів у 1-му шарі – 67, передаточні ф-ції – радіально-базисні, к-сть нейронів у 2-му шарі – 1, передаточна ф-ція – лінійна	2,6903e-015	0,0112
5	НМ з радіально- базисними елементами: к-сть нейронів у 1-му шарі – 2, передаточні ф-ції – радіально-базисні, к-сть нейронів у 2-му шарі – 1, передаточна ф-ція – лінійна	0,0126	0,0143

6	“Класична” багатошарова НМ з навчанням за методом зворотного поширення помилки: к-сть шарів – 9, к-сть нейронів у шарах 1-8 – 10, к-сть нейронів у шарі 10 – 1, передаточні функції – сигмоїдні	0,1452	0,0670
7	Мережа Елмана: к-сть шарів – 10, к-сть нейронів у шарах 1-9 – 10, к-сть нейронів у шарі 10 – 1, передаточні функції – сигмоїдні	0,0579	0,0723
8	Багатошарова НМ зі зворотним поширенням помилки – каскадна НМ: к-сть шарів – 10, к-сть нейронів у шарах 1-9 – 10, к-сть нейронів у шарі 10 – 1, передаточні функції – сигмоїдні	0,0644	0,0735
9	Перцептрон: к-сть нейронів – 1, одинична ф-ція активації з жорстким обмеженням	0,0644	0,0749
10	Ймовірнісна НМ: к-сть нейронів у 1-му шарі – 67, передаточні ф-ції – радіально-базисні, к-сть нейронів у 2-му шарі – 1, передаточна ф-ція – ф-ція конкуренції	0,9612	0,9474

Найвищу точність показала багатошарова НМ, результати роботи якої на тестовій вибірці даних при зміні вологості ґрунту в часі наведено на рис. 3 і 4.

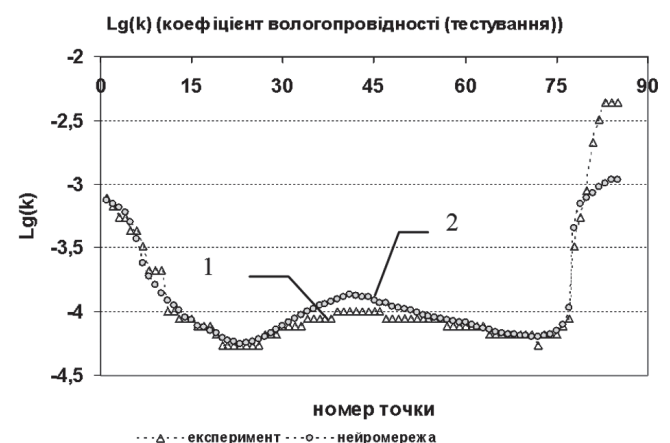


Рис. 3. Результати розрахунку коефіцієнта вологопровідності дерново-підзолистого ґрунту за допомогою багатошарової НМ на тестовій вибірці даних: 1 – тестова вибірка даних, 2 – результати роботи нейромережі

Проведемо порівняльний аналіз розрахунку коефіцієнта вологопереносу за аналітичними формулами та за допомогою багатошарової НМ. Для

цього побудуємо графічні залежності коефіцієнта вологопереносу від вологості та від всмоктуючого тиску ґрунту (рис. 5–8). У першому випадку у якості розрахункової залежності використаємо формулу Ван-Генухтена, у другому – формулу Уїнстона.

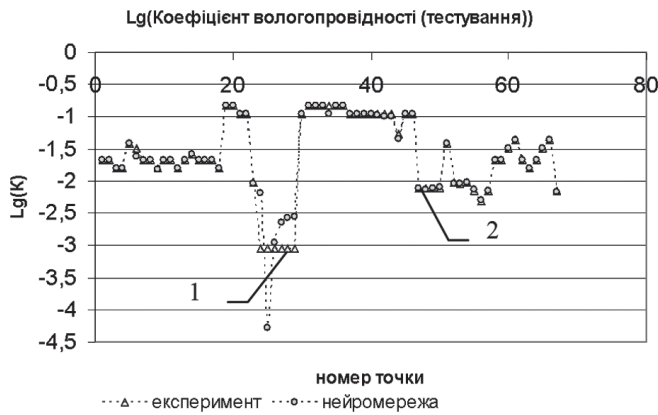


Рис. 4. Результати розрахунку коефіцієнта вологопровідності торфового ґрунту за допомогою багатосарової мережі на тестовій вибірці даних: 1 – тестова вибірка даних, 2 – результати роботи нейромережі

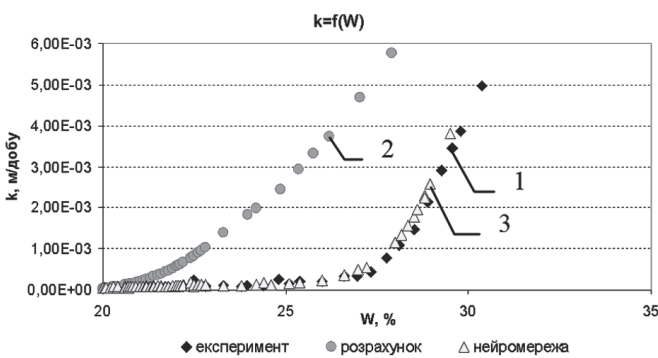


Рис. 5. Результати розрахунку коефіцієнта вологопровідності дерново-підзолистого ґрунту: 1 – експериментальні дані, 2 – результати розрахунку за формулою Ван-Генухтена (СКВ=0,0037), 3 – результати роботи нейромережі (СКВ=0,00045)

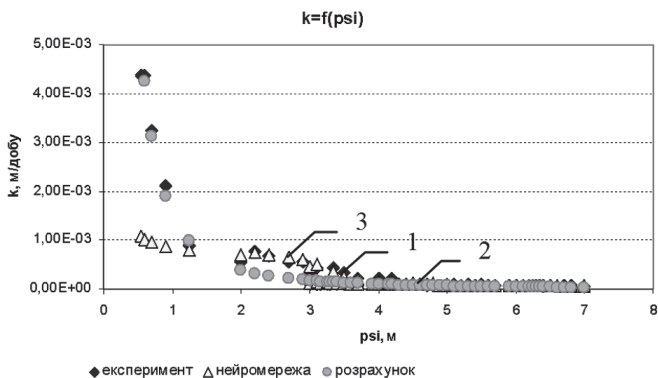


Рис. 6. Результати розрахунку коефіцієнта вологопровідності дерново-підзолистого ґрунту: 1 – експериментальні дані, 2 – результати розрахунку за формулою Уїнстона (СКВ=0,00015), 3 – результати роботи нейромережі (СКВ=0,00069)

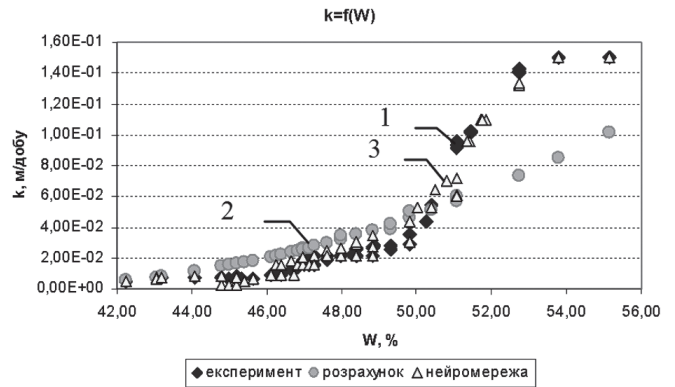


Рис. 7. Результати розрахунку коефіцієнта вологопровідності торфового ґрунту: 1 – експериментальні дані, 2 – результати розрахунку за формулою Ван-Генухтена (СКВ=0,0263), 3 – результати роботи нейромережі (СКВ=0,0107)

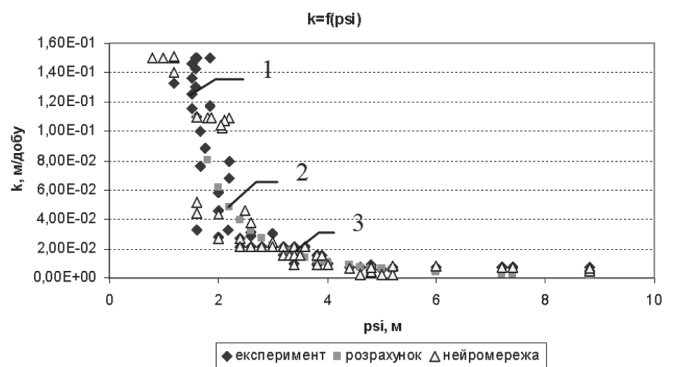


Рис. 8. Результати розрахунку коефіцієнта вологопровідності торфового ґрунту: 1 – експериментальні дані, 2 – результати розрахунку за формулою Уїнстона (СКВ=0,0363), 3 – результати роботи нейромережі (СКВ=0,0109)

Як бачимо із графіків та за даними СКВ нейромережа у більшості випадків показує вищу точність обчислень коефіцієнта вологопереносу, ніж аналітичні формули.

Висновки

Коефіцієнт вологопровідності, який описує фізичні процеси у ґрунті, можна визначити за допомогою ряду емпіричних формул, в які входять експериментальні коефіцієнти. Альтернативною методикою є використання нейронних мереж, за допомогою яких на основі експериментальної вибірки з достатньо високою точністю ідентифікується коефіцієнт вологопровідності або будь-який інший параметр ґрунту, що є функцією від експериментальних даних. Навчену НМ на навчальній вибірці даних можна успішно використовувати на тестових незалежних вибірках для певного типу ґрунту, які не входили у навчальний набір. Серед перспектив подальших досліджень є аналіз роботи інших архітектур НМ для ідентифікації параметрів ґрунту та керування його водно-повітряним режимом.

Список літератури: 1. Ковальчук, П.І. Системна оптимізація водокористування при зрошенні [Текст] / П.І. Ковальчук, Н.В. Пендак, В.П. Ковальчук, М.М. Волошин. — Рівне: НУВГП, 2008. — 204 с. 2. Муромцев, Н.А. Мелиоративная гидрофизика почв. Методы исследования, гидрофизические закономерности, регулирование водного режима почв и растений [Текст] / Н.А. Муромцев. — Л.: «Ленинград гидрометеоиздат», 1991. — 271 с. 3. Глобус, А.М. Почвенно-гидрофизическое обеспечение агроэкологических математических моделей [Текст] / А.М. Глобус. Л.: Гидрометеоиздат, 1987. — 423 с. 4. Луднин, К.П. Исследование влагопроводности ненасыщенной почвы [Текст] / К.П. Луднин, В.М. Гончарик, И.А. Пашкевич // Мелиорация переувлажнённых земель. — Минск, 1973. 5. Круглов, В.В. Нечёткая логика и искусственные нейронные сети [Текст] / В.В. Круглов, М.И. Дли, Р.Ю. Голунов. — Москва, 221 с.

Надійшла до редколегії 14.12.2011

УДК 631.432:626.86

Идентификация коэффициента влагопроводности грунта с помощью нейронных сетей / В.И. Пастушенко, С.К. Матус, А.М. Стеценко // Бионика интеллекта: науч.-техн. журнал. — 2012. — № 1 (78). — С. 23-28.

Статья посвящена анализу существующих методик вычисления коэффициента влагопереноса в ненасыщенную зону грунта и разработке альтернативных методик на базе нейросетей, которые обучаются на экспериментальных данных. При этом использовались нейросетиразной архитектуры. Проведен сравнительный анализ результатов расчетов, выполненных с помощью различных методик.

Табл. 4. Ил. 8. Библиогр.: 5 назв.

UDC 631.432:626.86

The coefficient's of soil's hydraulic conductivity identification by means of neural networks / V. Pastushenko, S. Matus, A. Stetsenko // Bionics of Intelligense: Sci. Mag. — 2012. — № 1 (78). — P. 23-28.

The article is devoted to analysis of existing methods of water transfer coefficient's calculation in non-saturated part of soil and to development of alternative methods on the base of neural networks, which are taught on experimental data. Neural networks of different architecture have being used. It was made a comparative analysis of calculations' results, carried out by different methods.

Tab. 4. Fig. 8. Ref.: 5 items.

УДК 004.93

Е.А. Гофман¹, А.А. Олейник², С.А. Субботин³¹Запорожский национальный технический университет,
г. Запорожье, Украина, gofman_jenek@rambler.ru;²Запорожский национальный технический университет,
г. Запорожье, Украина, olejnikaa@gmail.com;³Запорожский национальный технический университет,
г. Запорожье, Украина, subbotin@zntu.edu.ua

СИНТЕЗ ДЕРЕВЬЕВ РЕШЕНИЙ НА ОСНОВЕ ТЕОРИИ ПРИБЛИЖЕННЫХ МНОЖЕСТВ

Предложен новый метод идентификации деревьев решений с использованием теории приближенных множеств. Созданный метод позволяет сократить время работы и повысить эффективность синтезируемых моделей сложных объектов и систем. Проведены эксперименты по решению тестовых задач.

ДЕРЕВО РЕШЕНИЙ, ИДЕНТИФИКАЦИЯ, ПРИБЛИЖЕННОЕ МНОЖЕСТВО, ОБУЧАЮЩАЯ ВЫБОРКА

Введение

Теория приближенных множеств [1, 2] является важным математическим инструментом в области компьютерных технологий. В частности, теория приближенных множеств используется при принятии решений, интеллектуальном анализе данных, представлении знаний и др. Основным направлением, для которого может применяться теория приближенных множеств, является задача сокращения размерности входных данных, которая часто возникает при разработке различных информационных систем. Это объясняется тем, что практически любая система описывается набором признаков, определение каждого из которых зачастую требует значительных затрат материальных и временных ресурсов. Существуют различные методы, позволяющие находить сокращенный набор признаков для дальнейшей классификации на основании использования теории приближенных множеств [3, 4].

Таким образом, актуальным является применение теории приближенных множеств для синтеза моделей исследуемых объектов и систем при решении задач классификации. В данной работе предлагается выполнять идентификацию деревьев решений с предварительным применением теории приближенных множеств для обеспечения возможности построения распознающих моделей на основе сокращенного набора признаков.

Целью данной статьи является исследование и разработка метода синтеза деревьев решений с использованием теории приближенных множеств, который должен позволить повысить эффективность и скорость синтеза деревьев решений за счет использования сокращенного набора признаков, полученного при помощи теории приближенных множеств.

Для достижения поставленной цели необходимо решить следующие задачи:

- исследование теории приближенных множеств;
- выделение операций и подходов теории приближенных множеств, которые можно использовать для решения поставленной цели;
- рассмотрение теории построения деревьев решений;
- интеграция теории приближенных множеств в теорию синтеза деревьев решений, в результате чего необходимо разработать новый метод синтеза деревьев решений;
- проведение экспериментов и сравнение разработанного метода с существующими методами идентификации деревьев решений.

1. Теория приближенных множеств

Теория приближенных множеств (rough sets) была разработана [1, 2] как математический подход для описания неопределенности, неточности и неуверенности. Эта теория основана на утверждении, что с каждым объектом универсального множества связана некоторая информация (данные, знания). Объекты, характеризующиеся одинаковой информацией, являются неразличимыми (сходными). Отношение неразличимости, порожаемое таким способом, является математической основой теории приближенных (грубых) множеств.

Основой концепции теории приближенных множеств являются операции аппроксимации множеств.

В теории приближенных множеств таблица решений представляется как

$$T = (U, A, C, D),$$

где U – универсальное множество; A – множество всех признаков; C – подмножество признаков-условий; D – подмножество признаков-решений ($C, D \subset A$).

Пусть $a \in A$, $P \subseteq A$, тогда бинарное отношение $IND(P)$ называется отношением неразличимости:

$$IND(P) = \{(x, y) \in U \times U : \forall a \in P, a(x) = a(y)\}.$$

Пусть $U/IND(P)$ описывает множество всех классов эквивалентности отношения $IND(P)$. Классы эквивалентности $U/IND(C)$ и $U/IND(C)$ называются классами условий и решений соответственно.

Пусть $R \subseteq C$ и $X \subseteq U$, тогда

$$\begin{aligned} \underline{R}X &= \cup\{Y \in U / R : Y \subseteq X\}, \\ \overline{R}X &= \cup\{Y \in U / R : Y \cap X\}, \end{aligned}$$

где $\underline{R}X$ и $\overline{R}X$ – R -нижняя и R -верхняя аппроксимации X , соответственно, $(\underline{R}X, \overline{R}X)$ – R -приближённое множество. Если X является R -определяемым, то $\underline{R}X = \overline{R}X$, в противном случае – X является R -приближённым.

Граница $BN_R(X)$ определяется как $BN_R(X) = \overline{R}X - \underline{R}X$. Таким образом, если X является R -определяемым, то $BN_R(X) = \Phi$.

Пусть $c \in C$. Признак c является несущественным в T , если $POS_{(C-(c))}(D) = POS_C(D)$, в противном случае – признак является незаменимым в T . Признак c является независимым, если все $c \in C$ являются незаменимыми.

Множество признаков $R \subseteq C$ является сокращением C , если $T' = \{U, A, R, D\}$ является независимым и выполняется $POS_{R'}(D)$. Другими словами, сокращением является минимальное подмножество признаков, сохраняющее представленное выше условие.

$CORE(C)$ определяет множество всех признаков, которые являются незаменимыми в C :

$$CORE(C) = \cap REDUCT(C),$$

где $REDUCT(C)$ – множество всех сокращений C .

2. Деревья решений

Деревья решений представляют собой нисходящую систему, основанную на подходе “разделяй и властвуй”, основной целью которой является разделение дерева на взаимно непересекающиеся подмножества [5, 6]. Каждое подмножество представляет собой подзадачу классификации.

Дерево решений описывает процедуру принятия решения о принадлежности определённого экземпляра к тому или иному классу.

Дерево решений является древовидной структурой, состоящей из внутренних и внешних узлов, связанных рёбрами [7]. Внутренние узлы – модули, принимающие решение, рассчитывают значение функции решения, на основании чего определяют дочерний узел, который будет посещён далее. Внешние узлы (также называемые конечными узлами), напротив, не имеют дочерних узлов и описывают либо метку класса, либо значение, характеризующее входные данные. В общем случае, деревья решений используются следующим образом. Вначале передаются данные (обычно это вектор значений входных переменных) на корневой узел дерева решений. В зависимости от полученного значения функции решения, используемой во

внутреннем узле, происходит переход к одному из дочерних узлов. Такие переходы продолжаются до тех пор, пока не будет посещён конечный узел, описывающий либо метку класса, либо значение, связанное со входным вектором значений признаков.

Для применения деревьев решений на практике в целях классификации или прогнозирования значений выходных параметров исследуемых объектов по наборам значений входных характеристик необходимо с помощью данных обучающей выборки сформировать дерево решений таким образом, чтобы оно наилучшим образом описывало исследуемый объект.

Пусть задана обучающая выборка

$$S = \langle X, Y \rangle,$$

где $X = \{X_i\}$ – набор значений признаков, характеризующих рассматриваемый объект или процесс; $Y = \{y_p\}$ – массив значений выходного параметра в заданной выборке; $X_i = \{x_{ip}\}$ – i -й признак в выборке, $i = 1, 2, \dots, L$; $x_{ip} \in [x_{\min i}, x_{\max i}]$ – значение i -го признака для p -го экземпляра выборки, $p = 1, 2, \dots, m$; L – общее число признаков в исходном наборе; m – число экземпляров выборки.

Тогда задача построения дерева решений $T = \{t_k\}$ по заданной выборке S заключается в идентификации узлов $t_k = \langle c_k, l_k, r_k \rangle$ так, чтобы значение ошибки прогнозирования или классификации E построенной модели было минимальным: $E \rightarrow \min$, где $t_k = \langle c_k, l_k, r_k \rangle$ – k -й узел дерева T , представляющий собой структуру, в которой c_k – функция принятия решений на основе значений входных переменных (в случае, если узел является внутренним) или значение выходной переменной (для внешних узлов), l_k и r_k – ссылки на левого и правого потомков k -го узла соответственно, представляющих собой структуры, аналогичные t_k .

Построение деревьев решений связано с извлечением правил из обучающих выборок. Каждый путь от корня дерева к одному из его листьев может быть преобразован к логическому высказыванию – правилу типа «если А, то В», где его антецедент получается путем использования всех условий, представленных во внутренних узлах от корня к выходному листу, а правая часть правила получается из соответствующего листа дерева.

Поэтому постановку задачи синтеза дерева решений как логической модели исследуемого объекта, процесса или явления можно также представить в следующем виде. Пусть задана обучающая выборка данных, состоящая из m экземпляров, каждый из которых характеризуется L атрибутами. При этом каждый атрибут может относиться к определённому лингвистическому терму LT . Для каждого i -го экземпляра указаны вхождения к лингвистическим термам для каждого атрибута и указан лингвистический терм выходной переменной. Тогда необходимо построить такое дерево

решений, которое позволяет выполнять отнесение выходного параметра к лингвистическому терму с заданной точностью.

3. Синтез деревьев решений с использованием теории приближённых множеств

Синтез деревьев решений на основе использования теории приближённых множеств состоит из двух фаз: вычисление сокращённых наборов признаков и непосредственно идентификация дерева решений. Таким образом, достигается интеграция двух подходов, за счёт чего обеспечивается повышение эффективности разрабатываемого метода. Входная обучающая выборка может быть как дискретной, так и непрерывной. Однако, при построении дерева решений на основе непрерывных выборок, их необходимо предварительно дискретизировать [8].

Таким образом, предлагаемый метод состоит из этапов, описанных ниже.

Этап 1. Входная обработка данных. На данном этапе задаётся входная выборка данных $T1$. При необходимости выполняется дискретизация непрерывных признаков, в результате чего получается выборка данных $T2$.

Этап 2. Определение сокращённого набора признаков для $T2$. Данный этап предназначен для поиска сокращённого набора признаков, на основе которого будет синтезироваться дерево решений. Данный этап состоит из следующих шагов.

Шаг 1. Входная таблица данных $T2$ разбивается так, что последняя колонка считается колонкой решений.

Шаг 2. Строки входной таблицы данных сортируются таким образом, что экземпляр с наименьшим значением выходного признака оказывается вверху таблицы, т.е. экземпляры перераспределяются в порядке возрастания.

Шаг 3. Генерация булевой матрицы для заданной таблицы входных данных. Для этого выполняется проверка по каждому экземпляру. В случае, если значения первого и второго признаков для всех экземпляров одинаковы, то в булеву матрицу заносится значение «1», в противном случае – «0»:

$$a_{i,j} = \begin{cases} 1, & x_l^j = x_l^i, \forall l = \overline{1,m}; \\ 0, & \exists l: x_l^j \neq x_l^i, \forall l = \overline{1,m}, \end{cases}$$

где i, j – признаки i и j , соответственно; m – количество экземпляров входной таблицы данных; l – индекс текущего экземпляра выборки данных.

Шаг 3 повторяется для каждого признака выборки данных. Это продолжается до тех пор, пока не будет построена полная булева матрица для всех пар признаков.

Шаг 4. После выполнения шага 3 в полученной булевой матрице производится сложение значений каждого столбца:

$$\forall i = \overline{1,n}: A_i = \sum_{j=1, i \neq j}^n a_{i,j},$$

где n – количество столбцов входной таблицы данных.

Столбец с максимальной суммой считается незаменимым признаком:

$$x_{\text{inf}} = x_{\max_{i=1,n} A_i},$$

где $\max_{i=1,n} A_i$ – индекс столбца с максимальной суммой.

Шаг 5. На основании полученной ранее булевой матрицы получается сокращённая булева матрица путём удаления из матрицы выбранного на шаге 4 признака:

$$A' = A / A_{\max_{i=1,n} A_i}.$$

После чего сокращённая булева матрица становится рабочей булевой матрицей:

$$A = A'.$$

Шаг 6. Шаги 4 и 5 продолжают до тех пор, пока не выполняется одно из следующих условий.

1. Суммы столбцов сокращённой булевой матрицы равны нулю, что означает следующее: нет больше никакой информации о том, каково различие во влиянии признаков:

$$\forall i = \overline{1,n}: \sum_{j=1, i \neq j}^n a_{i,j} = 0.$$

2. Сокращённая булева матрица пуста:

$$A = \emptyset.$$

Шаг 7. Отобранные незаменимые признаки группируются и считаются полученным сокращённым набором признаков, который будет использоваться на следующем этапе.

Таким образом, в результате этапа 2 получается выборка данных $T3$.

Этап 3. Идентификация дерева решений. Синтез дерева решений выполняется на основе выборки данных $T3$, при этом в каждый момент времени берётся один признак и для разделения используются все узлы на одном уровне:

$$T3 \Rightarrow DT,$$

где DT – полученное дерево решений, дерево решений получается путём использования любого метода синтеза деревьев решений.

Этап 4. Генерация правил. Правила генерируются путём всевозможных переходов от корневого узла к листьям в полученном дереве решений:

$$DT \Rightarrow RB,$$

где RB – полученная база правил.

Как видно, предложенный метод включает в себя две основные фазы: вычисление сокращённых наборов признаков и идентификация дерева решений. В связи с этим сложность предложенного метода зависит от сложности каждой из этих фаз.

Таким образом, если обучающая выборка данных состоит из m экземпляров и n признаков, то задача вычисления сокращённого набора признаков

минимальной длины является NP-сложной задачей. Вычислительная сложность предобработки входной таблицы данных и её сортировки равна $O(m^2)$. При этом необходимо произвести $C(m, 2)$ сравнений и, если каждый экземпляр характеризуется n признаками, то сложность сравнения равна $O(nm^2)$. Сложность идентификация дерева решений зависит от значений разделяющих признаков, таким образом, построенное дерево решений может быть n -арным деревом.

Предложенный метод синтеза деревьев решений на основе теории приближённых множеств был программно реализован в среде пакета Matlab 7.0.

При помощи разработанного программного обеспечения и встроенных средств пакета Matlab 7.0 проводились эксперименты. Для экспериментов использовались тестовые данные, которые были взяты из общедоступных репозиториях [9]. Экспериментальные исследования проводились на основании выборки, которая содержала информацию об эхокардиограммах пациентов с сердечными приступами. Выборка содержала информацию о 132 пациентах, каждый из которых характеризовался 12 признаками. Кроме того, для каждого пациента указывалось, жив он или умер. Предложенный метод сравнивался с методом построения деревьев решений ID3 [5, 6], а также с мультиагентным [10, 11] и эволюционным [11, 12] методами. На основании проведенных экспериментов были получены базы правил, характеризующиеся следующим качеством классификации пациентов: 95,2%, 89,4%, 92,3% и 91,1% для предложенного, ID3, мультиагентного и эволюционного методов соответственно.

Таким образом, можно отметить, что предложенный метод построения деревьев решений на основании теории приближенных множеств обеспечивает более точные результаты прогнозирования по сравнению с другими известными методами.

Выводы

В работе решена актуальная задача синтеза деревьев решений, которые могут использоваться для классификации, а также для построения базы правил с целью разработки экспертных систем.

Научная новизна работы заключается в том, что разработан новый метод построения деревьев решений с использованием теории приближенных множеств, который позволяет повысить эффективность и скорость синтеза деревьев решений за счёт использования сокращённого набора признаков, получаемого при помощи теории приближенных множеств.

Практическая ценность полученных результатов заключается в том, что на основе предложенного метода разработано программное обеспечение, позволяющее выполнять идентификацию деревьев решений и получение базы правил, на основании которых можно создавать экспертные системы с меньшей ошибкой классификации.

Список литературы: 1. Pawlak Z. Rough sets / Z. Pawlak // International Journal of Computer and Information Sciences. — 1982. — № 11. — P. 341-356. 2. Pawlak Z. Rough Sets-Theoretical Aspects and Reasoning about Data / Z. Pawlak. — Dordrecht : Kluwer Academic Publications, 1991 — 237 p. 3. Ramadevi Y. Knowledge Extraction Using Rough Sets / Y. Ramadevi, C.R. Rao // Classification, International conference on Bioinformatics and diabetes mellitus. — India, 2006. — P. 128-132. 4. Олейник, Ал. А. Модификация метода муравьиных колоний с использованием операций над чёткими множествами [Текст] / Ал. А. Олейник, С. А. Субботин // Автоматика-2008 : п'ятнадцата міжнародна науково-технічна конференція, 23–26 вересня 2008 р. : тези доповідей. — Одесса, 2008. — С. 396–398. 5. Quinlan J. R. Induction of decision trees / J. R. Quinlan // Machine Learning. — 1986. — № 1. — P. 81–106. 6. Rokach L. Data Mining with Decision Trees. Theory and Applications / L. Rokach, O. Maimon. — London : World Scientific Publishing Co, 2008. — 264 p. 7. Classification and regression trees / L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. — California : Wadsworth & Brooks, 1984. — 368 p. 8. Субботин, С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навч. посібник [Текст] / С. О. Субботин. — Запоріжжя : ЗНТУ, 2008. — 341 с. 9. UCI Machine Learning Repository [electronic resource] / Center for Machine Learning and Intelligent Systems. — Access mode : <http://archive.ics.uci.edu/ml/datasets.html>. 10. Олейник, А. А. Мультиагентные методы интеллектуальной оптимизации для моделирования сложных объектов и систем [Текст] / А. А. Олейник, С. А. Субботин // Нейроинформатика, её приложения и анализ данных : XVII Всероссийский семинар, 2–4 октября 2009 г. : материалы семинара. — Красноярск, 2009. — С. 79–82. 11. Субботин, С. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткіологічних і нейромережних моделей: монографія [Текст] / С. О. Субботин, А. О. Олійник, О. О. Олійник ; під заг. ред. С.О. Субботіна. — Запоріжжя : ЗНТУ, 2009. — 375 с. 12. Gen M. Genetic algorithms and engineering design / M. Gen, R. Cheng. — New Jersey : John Wiley & Sons, 1997. — 352 p.

Поступила в редколлегию 11.01.2012

УДК 004.93

Синтез дерев розв'язків на основі теорії наближених множин / Є. О. Гофман, О. О. Олійник, С. О. Субботин // Біоніка інтелекту: наук.-техн. журнал. — 2012. — № 1 (78). — С. 29-32.

Запропоновано новий метод ідентифікації дерев розв'язків з використанням теорії наближених множин. Створений метод дозволяє скоротити час роботи й підвищити ефективність синтезованих моделей складних об'єктів і систем. Проведено експерименти з вирішення тестових завдань.

Бібліогр.: 12 найм.

UDC 004.93

Synthesis of decision trees based on rough set theory / Ye. A. Gofman, O. O. Oliinyk, S. A. Subbotin // Bionics of Intelligense: Sci. Mag. — 2012. — № 1 (78). — P. 29-32.

A new method of identification of decision trees using the theory of rough sets is proposed. Created a method allows to reduce the time and improve the effectiveness of the synthesized models of complex objects and systems. Experiments on the solution of test problems are conducted.

Ref.: 12 items.

УДК 519.6



О.М. Литвин, О.В. Ярмош

Українська інженерно-педагогічна академія,
Харків, Україна, yel_mag@mail.ru

ПРО ПОХИБКУ АПРОКСИМАЦІЇ ФУНКЦІЇ ДВОХ ЗМІННИХ БІЛІНІЙНИМИ СПЛАЙНАМИ МНК В ІНТЕГРАЛЬНІЙ ФОРМІ

Досліджується похибка наближення функцій двох змінних, заданих слідами на системі взаємно перпендикулярних прямих точно або з похибками, операторами інтерлінації з використанням допоміжних функцій у вигляді білінійних сплайнів методом найменших квадратів (МНК) в інтегральній формі. Також запропоновано вигляд оператора наближення функцій двох змінних операторами інтерлінації функцій за допомогою МНК в інтегральній формі.

ПОХИБКА НАБЛИЖЕННЯ, БІЛІНІЙНИЙ СПЛАЙН, МЕТОД НАЙМЕНШИХ КВАДРАТІВ, ОПЕРАТОР ІНТЕРЛІНАЦІЇ ФУНКЦІЙ

Вступ

Практичне отримання найкращих наближень для окремих функцій або для класів функцій за допомогою тригонометричних поліномів (чи інших систем ортогональних функцій) є складною задачею. Тому на практиці, як правило, користуються операторами апроксимації Фур'є, Фейера, Валле Пуссена, Хаара та іншими операторами, що не дають найкращого наближення, але мають ряд переваг (простота обчислення коефіцієнтів, менша кількість арифметичних операцій для обчислення наближеного значення функції в одній точці тощо).

На даний час достатньо добре вивчено наближення інтерполяційними сплайнами функцій однієї та багатьох змінних [1-7]. Слід підкреслити таку особливість одновимірної теорії сплайнів: в одновимірному випадку теорію наближення сплайнами можна створювати на відрізку $I = [0, 1]$, а потім заміною змінних переносити результати на відрізок $[a, b]$. Тому питання вивчення вигляду операторів наближення з використанням допоміжних функцій у вигляді білінійних сплайнів, а також порівняння похибки такого наближення з похибками інших операторів наближення є актуальними.

В роботах [6,7] побудовані оператори сплайн-апроксимації та сплан-інтерлінації функцій, що дозволяють отримати наближення функцій високої точності та встановлені точні оцінки похибки наближення функцій однієї змінної сплайнами першого порядку в нормі $L_2[0, 1]$.

1. Постановка задачі

В даній роботі досліджується похибка наближення функцій двох змінних, заданих слідами на системі взаємно перпендикулярних прямих, операторами інтерлінації з використанням білінійних сплайнів методом найменших квадратів (МНК) в інтегральній формі.

В основу роботи покладено теорему, що наведено в роботі [2, с. 302] без доведення.

Теорема 1. Для функції $f(x, y) \in W_2^{1,1}(I^2)$, $I = [0, 1]$ оператори

$$L_{m,n}(x, y, C) = \sum_{i=0}^m \sum_{j=0}^n C_{i,j} s_{m,i}(x) s_{n,j}(y),$$

коефіцієнти $C_{i,j}$, $i = \overline{0, m}$, $j = \overline{0, n}$ яких знаходяться з умови

$$J_{m,n}(C, L_{m,n}) = \int_0^1 \int_0^1 \sum_{p=0}^1 \sum_{q=0}^1 \left(\frac{\partial^{p+q}}{\partial x^p \partial y^q} [f(x, y) - L_{m,n}(x, y, C)] \right)^2 dx dy \rightarrow (1) \rightarrow \min_C,$$

можна зобразити у вигляді

$$L_{m,n} f(x, y) = A_{1,m} A_{2,n} f(x, y),$$

де $A_{1,m} f(x, y) = \sum_{i=0}^m \varphi_i(y) s_{m,i}(x),$

$$A_{2,n} f(x, y) = \sum_{j=0}^n \psi_j(x) s_{n,j}(y),$$

а невідомі функції $\varphi_i(y)$, $\psi_j(x)$ знаходяться МНК з умов

$$\int_0^1 \sum_{p=0}^1 \left(\frac{\partial^p}{\partial x^p} [f(x, y) - A_{1,m} f(x, y)] \right)^2 dx \rightarrow \min_{\varphi_i(y)},$$

$$\int_0^1 \sum_{q=0}^1 \left(\frac{\partial^q}{\partial y^q} [f(x, y) - A_{2,n} f(x, y)] \right)^2 dy \rightarrow \min_{\psi_j(x)}.$$

2. Похибка наближення функції двох змінних, отримана за допомогою похибок наближення операторами, що діють на одну змінну

В роботі [3] доведені теореми щодо вигляду операторів наближення, запропонованих у [2].

Нехай $h_{1,k}(x)$, $h_{2,l}(y)$ – задана система лінійно незалежних функцій; функції $\varphi_k(y)$, $\psi_l(x)$ і сталі $C_{k,l}$ вважаються невідомими. Доведено, що запропонований у [6-8] оператор наближення функції $f(x, y)$

$$Z(x, y) = \sum_{k=0}^N \varphi_k(y) h_{1,k}(x) + \sum_{l=0}^N \psi_l(x) h_{2,l}(y) - \sum_{k=0}^N \sum_{l=0}^N C_{k,l} h_{1,k}(x) h_{2,l}(y),$$

у якому невідомі $\varphi_k(y)$, $\psi_l(x)$ і $C_{k,l}$ знаходяться з умови (1), де замість $L_{m,n}(x, y, C)$ стоїть $Z(x, y)$, має вигляд

$$Z^*(x, y) = h_1(x) B_1^{-1} F_1^T(y) + F_2(x) B_2^{-1} h_2^T(y) - h_1(x) B_1^{-1} F B_2^{-1} h_2^T,$$

може бути також представлений так:

$$Z^*(x, y) = A_1 f(x, y) + A_2 f(x, y) - A_1 A_2 f(x, y), \quad (2)$$

де оператори A_1 , A_2 визначаються МНК з умов

$$\int_0^1 [f(x, y) - C_1(x) h_2^T(y)]^2 dy \rightarrow \min_{C_1},$$

$$\int_0^1 [f(x, y) - h_1(x) C_2^T(y)]^2 dx \rightarrow \min_{C_2},$$

$$\int_0^1 \int_0^1 [f(x, y) - h_1(x) C h_2^T(y)]^2 dx dy \rightarrow \min_C$$

формулами:

$$A_1 f(x, y) = h_1(x) \varphi^T(y) = h_1(x) B_1^{-1} F_1^T(y),$$

$$A_2 f(x, y) = \psi(x) h_2^T(y) = F_2(x) B_2^{-1} h_2^T(y),$$

$$A_1 A_2 f(x, y) = h_1(x) B_1^{-1} F B_2^{-1} h_2^T(y),$$

де $F = \int_0^1 \int_0^1 h_1^T(x) f(x, y) h_2(y) dx dy$;

$$F_1^T(y) = \int_0^1 h_1^T(x) f(x, y) dx,$$

$$F_2(x) = \int_0^1 f(x, y) h_2(y) dy, \quad B_1 = \int_0^1 h_1^T(x) h_1(x) dx,$$

$$B_2 = \int_0^1 h_2^T(y) h_2(y) dy.$$

Також доведено, що оператор

$$Z_0(x, y; C) = Z_0(x, y) = \sum_{k=0}^N \sum_{l=0}^N C_{k,l} h_{1,k}(x) h_{2,l}(y) = h_1(x) C h_2^T(y),$$

у якому матриця C знаходиться МНК, можна представити так

$$Z_0(x, y) = Z_0 f(x, y) = A_1 A_2 f(x, y). \quad (3)$$

Зображення операторів наближення (2) та (3) у вигляді операторів, що діють на одну змінну (x або y), дозволяють довести теореми про вигляд похибки наближення.

Теорема 2 [8]. Похибка наближення функції $f(x, y)$ за допомогою $Z_0 f(x, y)$ має вигляд:

$$Rf(x, y) = f(x, y) - Z_0 f(x, y) = (R_1 + R_2 - R_1 R_2) f(x, y), \quad (4)$$

$$\text{де } R_1 f(x, y) = f(x, y) - A_1 f(x, y),$$

$$R_2 f(x, y) = f(x, y) - A_2 f(x, y).$$

Теорема 3 [8]. Похибка наближення функції $f(x, y)$ за допомогою $Zf(x, y)$, де $Zf(x, y) = Z^*(x, y)$ визначається формулою (2), має вигляд:

$$Rf(x, y) = f(x, y) - Zf(x, y) = R_1 R_2 f(x, y). \quad (5)$$

Наслідок 1. Припустимо, що

$$\max_{0 \leq y \leq 1} |f(x, y) - A_1 f(x, y)| = O(\varepsilon), \quad 0 \leq x \leq 1,$$

$$\max_{0 \leq x \leq 1} |f(x, y) - A_2 f(x, y)| = O(\varepsilon), \quad 0 \leq y \leq 1.$$

Тоді для залишку $f(x, y) - Zf(x, y)$ справедлива оцінка $|f(x, y) - Zf(x, y)| = O(\varepsilon^2) \forall (x, y) \in [0, 1]^2$.

Відповідну оцінку похибки для випадку, коли функція задається своїми слідами на системі взаємно перпендикулярних прямих з похибками, можна отримати за допомогою такої теореми.

Теорема 4. У випадку, якщо відомі сліди функції $f(x_p, y)$ та $f(x, y_q)$ ($p, q = \overline{0, M}$) і наближуємо $f(x, y)$ з умови

$$J(C) = \sum_{p=0}^M \int_0^1 \left[f(x_p, y) - \sum_{k=0}^N \sum_{l=0}^N C_{k,l} h_{1,k}(x_p) h_{2,l}(y) \right]^2 dy + \sum_{q=0}^M \int_0^1 \left[f(x, y_q) - \sum_{k=0}^N \sum_{l=0}^N C_{k,l} h_{1,k}(x) h_{2,l}(y_q) \right]^2 dx \rightarrow \min_C, \quad (6)$$

де функції $h_{1,k}(x)$ та $h_{2,l}(y)$ – відомі лінійно-незалежні функції, $C_{k,l}$, $k, l = \overline{0, N}$ – невідомі сталі, то для матриці C виконується співвідношення:

$$C = A^{-1} F, \quad (7)$$

де $A = [a_{i,j}]_{i,j=0}^M$,

$$a_{(N+1)\mu+v, (N+1)k+l} = \tilde{A}_{k,l,\mu,v} = B_{2,l,v} b_{1,k,\mu} + B_{1,k,\mu} b_{2,l,v},$$

$$F_{(N+1)\mu+v} = \tilde{F}_{\mu,v} = F_{1,\mu,v} + F_{2,\mu,v},$$

$$B_{1,k,\mu} = \int_0^1 h_{1,k}(x) h_{1,\mu}(x) dx, \quad B_{2,l,v} = \int_0^1 h_{2,l}(y) h_{2,v}(y) dy,$$

$$b_{1,k,\mu} = \sum_{p=0}^M h_{1,k}(x_p) h_{1,\mu}(x_p), \quad b_{2,l,v} = \sum_{q=0}^M h_{2,l}(y_q) h_{2,v}(y_q),$$

$$F_{1,\mu,v} = \sum_{p=0}^M \int_0^1 f(x_p, y) h_{2,v}(y) dy h_{1,\mu}(x_p),$$

$$F_{2,\mu,v} = \sum_{q=0}^M \int_0^1 f(x, y_q) h_{1,\mu}(x) dx h_{2,v}(y_q).$$

Доведення. З умови (6) отримуємо рівняння для знаходження матриці C :

$$\frac{\partial J}{\partial C_{\mu,v}} = 0 \Rightarrow$$

$$\begin{aligned} & \sum_{p=0}^M \int_0^1 \left[f(x_p, y) - \sum_{k=0}^N \sum_{l=0}^N C_{k,l} h_{1,k}(x_p) h_{2,l}(y) \right] \times \\ & \quad \times h_{1,\mu}(x_p) h_{2,\nu}(y) dy + \\ & + \sum_{q=0}^M \int_0^1 \left[f(x, y_q) - \sum_{k=0}^N \sum_{l=0}^N C_{k,l} h_{1,k}(x) h_{2,l}(y_q) \right] \times \\ & \quad \times h_{1,\mu}(x) h_{2,\nu}(y_q) dx = 0, \\ & \quad \mu, \nu = \overline{0, N}. \end{aligned}$$

Перепишемо отримані рівності у вигляді:

$$\begin{aligned} & \sum_{k=0}^N \sum_{l=0}^N C_{k,l} \left[\int_0^1 h_{2,l}(y) h_{2,\nu}(y) dy \sum_{p=0}^M h_{1,k}(x_p) h_{1,\mu}(x_p) \right] + \\ & + \sum_{k=0}^N \sum_{l=0}^N C_{k,l} \left[\int_0^1 h_{1,k}(x) h_{1,\mu}(x) dx \sum_{q=0}^M h_{2,l}(y_q) h_{2,\nu}(y_q) \right] = \\ & = \sum_{p=0}^M \int_0^1 f(x_p, y) h_{2,\nu}(y) dy h_{1,\mu}(x_p) + \\ & + \sum_{q=0}^M \int_0^1 f(x, y_q) h_{1,\mu}(x) dx h_{2,\nu}(y_q), \quad \mu, \nu = \overline{0, N} \end{aligned}$$

або

$$\begin{aligned} & \sum_{k=0}^N \sum_{l=0}^N C_{k,l} (B_{2,l,\nu} b_{1,k,\mu} + B_{1,k,\mu} b_{2,l,\nu}) = F_{1,\mu,\nu} + F_{2,\mu,\nu}, \\ & C_{k,l} (A_{1,k,l,\mu,\nu} + A_{l,k,\mu,\nu}) = F_{1,\mu,\nu} + F_{2,\mu,\nu}, \\ & C_{k,l} A_{k,l,\mu,\nu} = F_{\mu,\nu}. \end{aligned}$$

Перейдемо від чотиривимірному масиву до двовимірною: $t = (N+1)k + l$ та $s = (N+1)\mu + \nu$.

$$\text{Тоді отримуємо } \sum_{t=0}^{(N+1)^2-1} A_{s,t} C_t = F_s, \quad s = \overline{0, (N+1)^2-1},$$

звідки $C = A^{-1} F$.

Теорема 4 доведена.

Для перевірки твердження теореми 4 розглянуто приклад.

Приклад. Нехай $f(x, y) = \ln[(x+a)^2 + (y+b)^2]$, $a = b = 1$. Якщо наближувати функцію $f(x, y)$, задану за допомогою слідів на системі взаємно перпендикулярних прямих, то з умови (6) отримуємо похибку наближення $\varepsilon = O(10^{-4})$.

3. Похибка наближення функції двох змінних, якщо вона задана наближено своїми слідами

Дослідимо рівень загальної похибки для випадку, коли функція задається своїми слідами на системі взаємноперпендикулярних прямих з деякими похибками. У такому випадку необхідно враховувати як власне похибку методу наближення функції, так і похибку даних.

Теорема 5. Якщо $f(x, y) \in C^{2,2}[0,1]^2$ та сліди $f(x_k, y)$, $f(x, y_l)$ задані функціями $\varphi_k(y)$, $\psi_l(x)$ з похибками, значення функції $f(x_k, y_l)$ також задано числами $\tilde{f}_{k,l}$ з похибками, тобто

$$\|f(x_k, y) - \varphi_k(y)\|_{C[0,1]} \leq \eta, \quad \|f(x, y_l) - \psi_l(x)\|_{C[0,1]} \leq \eta,$$

$$\max_{k,l} |f(x_k, y_l) - \tilde{f}_{k,l}| \leq \eta,$$

то загальна похибка наближення функції $f(x, y)$ оператором сплайн-інтерлінації

$$\begin{aligned} \tilde{O}f(x, y) &= \sum_{k=0}^N \varphi_k(y) h_{1,k}(x) + \sum_{l=0}^N \psi_l(x) h_{2,l}(y) - \\ & - \sum_{k=0}^N \sum_{l=0}^N \tilde{f}_{k,l} h_{1,k}(x) h_{2,l}(y) \end{aligned}$$

з використанням допоміжних функцій у вигляді сплайнів першого степеня визначаються таким чином:

$$\|f(x, y) - \tilde{O}f(x, y)\|_{C[0,1]^2} \leq O(\varepsilon^2) + O(\eta),$$

$$\varepsilon = O\left(\frac{1}{N^2}\right). \quad (8)$$

Доведення. Для оцінки норми

$$\|f(x, y) - \tilde{O}f(x, y)\|_{C[0,1]^2}$$

запишемо наступну низку співвідношень:

$$\begin{aligned} & \|f(x, y) - \tilde{O}f(x, y)\|_{C[0,1]^2} = \\ & = \|f(x, y) - Of(x, y) + Of(x, y) - \tilde{O}f(x, y)\|_{C[0,1]^2} \leq \\ & \leq \|f(x, y) - Of(x, y)\|_{C[0,1]} + \|Of(x, y) - \tilde{O}f(x, y)\|_{C[0,1]}, \end{aligned} \quad (9)$$

де

$$\begin{aligned} Of(x, y) &= \sum_{k=0}^N f(x_k, y) h_{1,k}(x) + \sum_{l=0}^N f(x, y_l) h_{2,l}(y) - \\ & - \sum_{k=0}^N \sum_{l=0}^N f(x_k, y_l) h_{1,k}(x) h_{2,l}(y). \end{aligned}$$

З урахуванням тверджень, наведених у [6-7], запишемо:

$$\|f(x, y) - Of(x, y)\|_{C[0,1]} = O(\varepsilon^2) = O\left(\frac{1}{N^4}\right),$$

$$N \rightarrow \infty.$$

Для оцінки другого доданку у правій частині (9) запишемо різницю під знаком норми так:

$$\begin{aligned} & \|Of(x, y) - \tilde{O}f(x, y)\|_{C[0,1]} = \\ & = \left\| \sum_{k=0}^N (f(x_k, y) - \varphi_k(y)) h_{1,k}(x) + \right. \\ & \quad \left. + \sum_{l=0}^N (f(x, y_l) - \psi_l(x)) h_{2,l}(y) - \right. \\ & \quad \left. - \sum_{k=0}^N \sum_{l=0}^N (f(x_k, y_l) - \tilde{f}_{k,l}) h_{1,k}(x) h_{2,l}(y) \right\|_{C[0,1]}. \end{aligned}$$

Звідси отримуємо

$$\begin{aligned} & \|Of(x, y) - \tilde{O}f(x, y)\|_{C[0,1]} = \\ & = \max_{0 \leq x, y \leq 1} \left\| \sum_{k=0}^N (f(x_k, y) - \varphi_k(y)) h_{1,k}(x) + \right. \\ & \quad \left. + \sum_{l=0}^N (f(x, y_l) - \psi_l(x)) h_{2,l}(y) - \right. \\ & \quad \left. - \sum_{k=0}^N \sum_{l=0}^N (f(x_k, y_l) - \tilde{f}_{k,l}) h_{1,k}(x) h_{2,l}(y) \right\|_{C[0,1]} \leq \\ & \leq \max_{k,l} \left\{ \|f(x_k, y) - \varphi_k(y)\|_{C[0,1]}; \right. \\ & \left. \|f(x, y_l) - \psi_l(x)\|_{C[0,1]}; \|f(x_k, y_l) - \tilde{f}_{k,l}\| \right\} = O(\eta). \end{aligned}$$

Теорема 5 доведена.

Зауваження. Таким чином, якщо $O(\eta) > O(\varepsilon^2)$, то внесок у загальне значення похибки від похибок задання слідів $f(x_k, y)$, $f(x, y_l)$ та значень $f(x_k, y_l)$ буде більшим, ніж похибка інтерлінації, тобто загальна похибка для $f(x, y) \in C^{2,2}[0,1]^2$ складе

$$R = \|f(x, y) - \tilde{O}f(x, y)\|_{C[0,1]^2} = O\left(\frac{1}{N^4}\right) + O(\eta),$$

$$N \rightarrow \infty, \eta \rightarrow 0.$$

Висновки

Випадок наближення функцій двох і більше змінних, заданих слідами на системі взаємно перпендикулярних прямих, має важливе практичне значення, оскільки сліди функції двох і більше змінних на системі заданих перетинних ліній або поверхонь все частіше знаходять застосування в різноманітних задачах науки і техніки (комп'ютерній томографії, цифровій обробці багатовимірних сигналів, конструюванні просторових математичних моделей розподілу корисних копалин тощо).

Сьогодні залишається проблема нестачі та неточності вхідних даних, що є особливо гострою для сфери економіки, освіти і науки та ін., та гальмує використання математичних методів в їх дослідженні. В таких випадках виникає необхідність дослідження похибок наближення функції та вхідних даних.

Отримані авторами результати використано при дослідженні сфери освітніх послуг, де дані найчастіше і задаються з похибками, а саме для дослідження рівня попиту на освітні послуги в залежності від рейтингу вищого навчального закладу та ціни освітньої послуги.

Список літератури: 1. *Ciesielsky Z.* Properties of the orthonormal Franklin systems [Текст] / Z. Ciesielsky // *Studia mathematica*, 1963. – 23, № 2. – С. 141-157. 2. *Завьялов, Ю.С.* Методы сплайн-функций [Текст] / Ю.С. Завьялов, Б.И. Квасов, В.Л. Мирошниченко. - М.: Наука, 1980. – 350 с. 3. *Каюмов, А.Р.* О задаче наилучшего среднеквадратического приближения L-сплайнами 2-го порядка [Текст] / А.Р. Каюмов. – Труды междунар. научн. конф. «Теория приближения», Екатеринбург, 2000. 4. *Корнейчук, Н.П.* Сплайны в теории приближения [Текст] / Н.П. Корнейчук. – М.: Наука, 1984. – 350 с. 5. *Стечкин, С.Б.* Сплайны в вычислительной математике [Текст] / С.Б. Стечкин, Ю.Н. Субботин. – М.: Наука, 1976. – 248 с. 6. *Литвин, О.М.* Інтерлінація функцій та деякі її застосування [Текст] / О.М. Литвин. – Х.: Основа, 2002. – 544 с. 7. *Литвин, О.М.* Методи обчислень. Додаткові розділи: навч. посіб. [Текст] / О.М. Литвин. – К.: Наукова думка, 2005. – 344 с. 8. *Литвин, О.Н.* Приближение функции $f(x, y)$ суммами вида $\varphi_0(x)\psi_0(x) + \dots + \varphi_N(x)\psi_N(x)$ [Текст] / О.Н. Литвин, Е.В. Ярмош // *Компьютерная математика*. – Киев, 2011. – Вып. 2. – С. 151-159.

Надійшла до редколегії 16.01.2012

УДК 519.6

Об ошибке аппроксимации функции двух переменных билинейным сплайнами МНК в интегральной форме / О.Н. Литвин, Е.В. Ярмош // *Бионика интеллекта: научн.-техн. журнал*. – 2012. – № 1 (78). – С. 33-36.

В статье рассмотрена погрешность приближения функций двух переменных, заданных следами на системе взаимноперпендикулярных прямых, с использованием вспомогательных функций в виде билинейных сплайнов методом наименьших квадратов в интегральной форме, приближение которых получено с помощью одномерных операторов. Сформулированы и доказаны теоремы о виде оператора и погрешность приближения функции для случая, когда следы функции на системе взаимноперпендикулярных прямых задаются с ошибками. Рассмотрен пример.

Библиогр.: 8 назв.

UDK 519.6

The error of the approximation of two variables functions by means bilinear splines with help least squares method in integral form / O.N. Lytvyn, O.V. Iarmosh // *Bionics of Intelligense: Sci. Mag.* – 2012. – № 1 (78). – P. 33-36.

The paper considers the approximation of two variables functions defined on a system of signs mutually perpendicular lines, using the auxiliary functions in the form of bilinear splines with help least squares method in integral form which is obtained by one-dimensional operators. Formulated and proved the theorem about the form of the operator and the approximation error function for the case where the traces of functions on a system of mutually perpendicular lines are set with error. Example are considered.

Ref.: 8 items.

УДК 519.6



О.М. Литвин, Ю.І. Першина

Українська інженерно-педагогічна академія, Харків, Україна
yulia_pershina@mail.ru

ВІДНОВЛЕННЯ ОБ'ЄКТІВ, ЩО ОПИСУЮТЬСЯ РОЗРИВНИМИ ФУНКЦІЯМИ, З ВИКОРИСТАННЯМ КРИВОЛІНІЙНИХ ТРАПЕЦІЙ

Представляється метод побудови розривних інтерполяційних та апроксимаційних сплайнів для наближення розривних функцій, область визначення яких розбивається на криволінійні трапеції. Причому побудовані розривні конструкції включають в себе, як частинний випадок, класичні неперервні сплайни. Запропоновані методи наближення можна буде використати для математичного моделювання розривних процесів в медичних, геологічних, космічних та інших дослідженнях.

РОЗРИВНА ФУНКЦІЯ, КРИВОЛІНІЙНА ТРАПЕЦІЯ, РОЗРИВНА ІНТЕРПОЛЯЦІЯ, РОЗРИВНА АПРОКСИМАЦІЯ

Вступ

Задача наближення неперервних функцій неперервними сплайнами від однієї та декількох змінних з достатньою повнотою описана в багатьох роботах вітчизняних та зарубіжних дослідників (див. наприклад, [1]). На практиці використання кусково-аналітичних наближень, заданих різними формулами (поліномами відповідного степеня) в точках кожного елемента розбиття області наближення, приводить інколи до знаходження великої кількості невідомих параметрів. Це привело до появи неконформних елементів у методі скінченних елементів [2]. Аналогічна задача досліджувалась у працях Попова Б.А. [3] та інших авторів, де розглядалися наближення неперервних та неперервно-диференціальних функцій за допомогою розривних сплайнів в чебишовській нормі (рівномірне наближення). У роботі [4] була розглянута апроксимація розривних розв'язків (функцій однієї змінної) диференціальних рівнянь за допомогою розривного методу Гальоркіна. А в роботі [5] розглядається розривний метод Гальоркіна для еліптичної крайової задачі з використанням двовимірних неузгоджених сіток. Цей метод дозволяє враховувати неконформність елементів. Причому метод забезпечує неперервність розв'язку, хоча від базисних функцій узгодженості не вимагає.

Таким чином, у вказаних роботах досліджувалося наближення неперервних функцій за допомогою неперервних та розривних сплайнів або розривних функцій за допомогою неперервних. Але загальної теорії таких наближень не існує. В даній роботі ми пропонуємо таку загальну теорію побудови розривних сплайнів, множина яких як частинний випадок, включає множину неперервних сплайнів, що можуть мати розриви першого роду у заданих точках або на заданій множині ліній – границь елементів.

Задачі наближення розривних функцій виникають частіше, ніж задачі наближення неперервних функцій. Наприклад, в методах комп'ютерної томографії на даний час недостатньо вивчене

питання про використання інформації про внутрішню структуру тіла людини (різні органи мають свою форму та щільність тканин).

Тобто актуальною є розробка та дослідження теорії наближення розривних функцій за допомогою розривних функцій.

В роботі [6] був запропонований метод наближення розривних функцій двох змінних розривними інтерполяційними білінійними сплайнами, а в роботі [7] – інтерлінаційними розривними сплайнами на ректангульованій області визначення. Були також побудовані розривні інтерлінаційні сплайни для наближення функцій двох змінних, область визначення яких розбивається на прямокутні трикутники [8].

У даній роботі вперше будуються та досліджуються інтерполяційні та апроксимаційні розривні сплайни для наближення розривних функцій з областю визначення, що розбивається на криволінійні трапеції (прямокутники з однією криволінійною стороною).

Постановка задачі

Нехай задана розривна функція двох змінних $f(x, y)$ в області $D = [0, 1]^2$. Будемо вважати, що область D розбита на криволінійні трапеції. Ці елементи не вкладаються один в один, і їх сторони не перетинаються. Функція $f(x, y)$ має розриви першого роду на границях між цими елементами. Метою роботи є побудова та дослідження операторів розривної кусково-поліноміальної інтерполяції та апроксимації, які в кожному елементі розбиття є операторами поліноміальної інтерполяції або апроксимації функції $f(x, y)$.

1. Побудова розривного інтерполяційного сплайна

Якщо (x_i, y_j) , $i = \overline{1, n}$, $j = \overline{1, m}$ – вузол, в якому знаходиться прямиий кут прямокутника, то може зустрітися вісім типів трапецій (рис. 1)

$$TP_{ij}^{(1)} = \{x_i < x < x_{i+1}, y_j < y < g_{j+1}^{(1)}(x)\};$$

$$TP_{ij}^{(2)} = \{x_{i-1} < x < x_i, y_j < y < g_{j+1}^{(2)}(x)\};$$

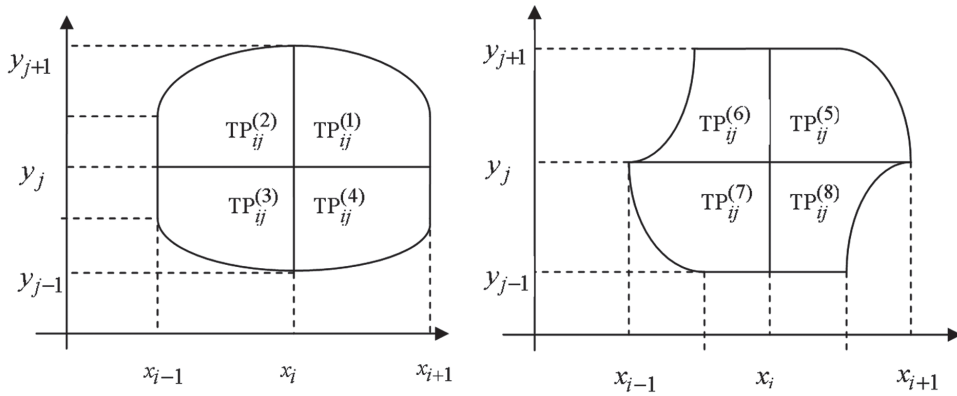


Рис. 1. Зображення можливих трапецевидних елементів з прямим кутом у вузлі (x_i, y_j)

$$\begin{aligned} TP_{ij}^{(3)} &= \{x_{i-1} < x < x_i, g_{j-1}^{(3)}(x) < y < y_j\}; \\ TP_{ij}^{(4)} &= \{x_i < x < x_{i+1}, g_{j-1}^{(4)}(x) < y < y_j\}; \\ TP_{ij}^{(5)} &= \{x_i < x < q_{i+1}^{(5)}(y), y_j < y < y_{j+1}\}; \\ TP_{ij}^{(6)} &= \{q_{i-1}^{(6)}(y) < x < x_i, y_j < y < y_{j+1}\}; \\ TP_{ij}^{(7)} &= \{q_{i-1}^{(7)}(y) < x < x_i, y_{j-1} < y < y_j\}; \\ TP_{ij}^{(8)} &= \{x_i < x < q_{i+1}^{(8)}(x), y_{j-1} < y < y_j\}, \end{aligned}$$

де функції

$$\begin{aligned} g_u^{(v)}(x), q_s^{(v)}(y), \quad u = \{j+1, j-1\}, \\ s = \{i+1, i-1\}, \quad v = \{1, 2, 3, 4\} \end{aligned}$$

є квадратичними функціями, тобто задаються виразами $g(x) = ax^2 + bx + c$ або $q(y) = ay^2 + by + c$

Вважаємо, що на кожній із сторін заданих трапецій функція $f(x, y)$ може мати (а може і не мати) розриви першого роду, причому у вузлах заданої сітки функція набуває таких значень:

$$\begin{aligned} C_1^{(1)} &= f(x_i + 0, y_j + 0), \\ C_2^{(1)} &= f(x_{i+1} - 0, y_j + 0), \\ C_3^{(1)} &= f(x_i + 0, g_{j+1}^{(1)}(x_i) - 0), \\ C_4^{(1)} &= f(x_{i+1} - 0, g_{j+1}^{(1)}(x_{i+1}) - 0), \\ C_1^{(2)} &= f(x_i - 0, y_j + 0), \\ C_2^{(2)} &= f(x_{i-1} + 0, y_j + 0), \\ C_3^{(2)} &= f(x_i - 0, g_{j+1}^{(2)}(x_i) - 0), \\ C_4^{(2)} &= f(x_{i-1} + 0, g_{j+1}^{(2)}(x_{i-1}) - 0), \\ C_1^{(3)} &= f(x_i - 0, y_j - 0), \\ C_2^{(3)} &= f(x_{i-1} + 0, y_j - 0), \\ C_3^{(3)} &= f(x_i - 0, g_{j-1}^{(3)}(x_i) + 0), \\ C_4^{(3)} &= f(x_{i+1} - 0, g_{j-1}^{(3)}(x_{i+1} - 0)), \\ C_1^{(4)} &= f(x_i + 0, y_j - 0), \\ C_2^{(4)} &= f(x_{i+1} - 0, y_j - 0), \\ C_3^{(4)} &= f(x_i + 0, g_{j-1}^{(4)}(x_i) + 0), \\ C_4^{(4)} &= f(x_{i+1} - 0, g_{j-1}^{(4)}(x_{i+1}) + 0), \end{aligned}$$

$$\begin{aligned} C_1^{(5)} &= f(x_i + 0, y_j + 0), \\ C_2^{(5)} &= f(q_{i+1}^{(5)}(y_j) - 0, y_j + 0), \\ C_3^{(5)} &= f(x_i + 0, y_{j+1} - 0), \\ C_4^{(5)} &= f(q_{i+1}^{(5)}(y_{i+1}) - 0, y_{j+1} - 0), \\ C_1^{(6)} &= f(x_i - 0, y_j + 0), \\ C_2^{(6)} &= f(q_{i-1}^{(6)}(y_j) + 0, y_j + 0), \\ C_3^{(6)} &= f(q_{i-1}^{(6)}(y_{j+1}) + 0, y_{j+1} - 0), \\ C_4^{(6)} &= f(x_i - 0, y_{j+1} - 0), \\ C_1^{(7)} &= f(x_i - 0, y_j - 0), \\ C_2^{(7)} &= f(q_{i-1}^{(7)}(y_j) + 0, y_j - 0), \\ C_3^{(7)} &= f(q_{i-1}^{(7)}(y_{j+1}) + 0, y_{j+1} - 0), \\ C_4^{(7)} &= f(x_i - 0, y_{j+1} - 0), \\ C_1^{(8)} &= f(x_i + 0, y_j - 0), \\ C_2^{(8)} &= f(q_{i+1}^{(8)}(y_j) - 0, y_j - 0), \\ C_3^{(8)} &= f(x_i + 0, y_{j-1} + 0), \\ C_4^{(8)} &= f(q_{i+1}^{(8)}(y_{j-1}) - 0, y_{j-1} + 0). \end{aligned}$$

Визначення. Будемо називати розривним інтерполяційним поліноміальним сплайном в області $TP_{ij}^{(k)} \subset D, k = \overline{1, 8}$ наступну функцію

$$\begin{aligned} S(x, y) = s_{ij}^{(k)}(x, y) = C_1^{(k)} \frac{\omega 4_{ij}^{(k)}(x, y) \omega 3_{ij}^{(k)}(x, y)}{\omega 4_{ij}^{(k)}(A_1^{(k)}) \omega 3_{ij}^{(k)}(A_1^{(k)})} + \\ + C_2^{(1)} \frac{\omega 2_{ij}^{(k)}(x, y) \omega 3_{ij}^{(k)}(x, y)}{\omega 2_{ij}^{(k)}(A_2^{(k)}) \omega 3_{ij}^{(k)}(A_2^{(k)})} + \\ + C_3^{(1)} \frac{\omega 1_{ij}^{(k)}(x, y) \omega 4_{ij}^{(k)}(x, y)}{\omega 1_{ij}^{(k)}(A_3^{(k)}) \omega 4_{ij}^{(k)}(A_3^{(k)})} + \\ + C_4^{(1)} \frac{\omega 1_{ij}^{(k)}(x, y) \omega 2_{ij}^{(k)}(x, y)}{\omega 1_{ij}^{(k)}(A_4^{(k)}) \omega 2_{ij}^{(k)}(A_4^{(k)})}, \end{aligned} \quad (1)$$

де

$$\begin{aligned} \omega 1_{ij}^{(k)}(x, y) &= y - y_j, \\ \omega 2_{ij}^{(k)}(x, y) &= x - x_i, \end{aligned}$$

$$\omega 3_{ij}^{(k)}(x, y) = \begin{cases} y - g^{(k)}(x), & k = \overline{1, 4} \\ y - y_{j+1}, & k = \overline{5, 8} \end{cases},$$

$$\omega 4_{ij}^{(k)}(x, y) = \begin{cases} x - x_{i+1}, & k = \overline{1, 4} \\ x - q^{(k)}(y), & k = \overline{5, 8} \end{cases},$$

$$A_1^{(k)} = (x_i, y_j),$$

$$A_2^{(k)} = \begin{cases} (x_{i+1} - 0, y_j + 0), & k = 1 \\ (x_{i-1} - 0, y_j + 0), & k = 2 \\ (x_{i-1} - 0, y_j - 0), & k = 3 \\ (x_{i+1} - 0, y_j - 0), & k = 4 \\ (q_{i+1}^{(5)}(y_j) - 0, y_j + 0), & k = 5, \\ (q_{i-1}^{(6)}(y_j) + 0, y_j + 0), & k = 6 \\ (q_{i-1}^{(7)}(y_j) - 0, y_j - 0), & k = 7 \\ (q_{i+1}^{(8)}(y_j) - 0, y_j - 0), & k = 8 \end{cases}$$

$$A_3^{(k)} = \begin{cases} (x_i + 0, g_{j+1}^{(1)}(x_i) - 0), & k = 1 \\ (x_i - 0, g_{j+1}^{(2)}(x_i) - 0), & k = 2 \\ (x_i - 0, g_{j-1}^{(3)}(x_i) + 0), & k = 3 \\ (x_i + 0, g_{j-1}^{(4)}(x_i) + 0), & k = 4 \\ (x_i + 0, y_{j+1} - 0), & k = 5 \\ (x_i - 0, y_{j+1} - 0), & k = 6 \\ (x_i - 0, y_{j-1} + 0), & k = 7 \\ (x_i + 0, y_{j-1} + 0), & k = 8 \end{cases}$$

$$A_4^{(k)} = \begin{cases} (x_{i+1} - 0, g_{j+1}^{(1)}(x_{i+1}) - 0), & k = 1 \\ (x_{i-1} + 0, g_{j+1}^{(2)}(x_{i-1}) - 0), & k = 2 \\ (x_{i-1} + 0, g_{j-1}^{(3)}(x_{i-1}) + 0), & k = 3 \\ (x_{i+1} - 0, g_{j-1}^{(4)}(x_{i+1}) + 0), & k = 4 \\ (q_{i+1}^{(5)}(y_{j+1}) - 0, y_{j+1} - 0), & k = 5 \\ (q_{i-1}^{(6)}(y_{j+1}) + 0, y_{j+1} - 0), & k = 6 \\ (q_{i-1}^{(7)}(y_{j-1}) + 0, y_{j-1} + 0), & k = 7 \\ (q_{i+1}^{(8)}(y_{j-1}) - 0, y_{j-1} + 0), & k = 8 \end{cases}$$

Теорема 1. Функція $S(x, y) = s_{ij}^{(k)}(x, y)$, $(x, y) \in \text{TP}_{ij}^{(k)} \subset D$, $k = \overline{1, 8}$ задовольняє інтерполяційним властивостям.

Доведення проводиться безпосередньою підстановкою відповідних значень аргументів у визначений розривний сплайн (1).

Теорема 2. Якщо $f(x, y)$ має розриви першого роду у деяких точках (x_i, y_j) та $f(x, y) \in C^{(r, r)}(\text{TP}_{ij}^{(1)})$, $i = \overline{1, m}$, $j = \overline{1, n}$, $r = 1, 2$, то залишок наближення функції $f(x, y)$ сплайном вигляду (1) на кожній трапеції буде мати вигляд

$$RS(x, y) = R_1 R_2 f(x, y) + \frac{x - x_{i+1}}{x_i - x_{i+1}} R_1 f(x_i, y) +$$

$$+ \frac{x - x_i}{x_{i+1} - x_i} R_1 f(x_{i+1}, y) + \frac{y - g_{j+1}^{(1)}(x)}{y_j - g_{j+1}^{(1)}(x)} R_2 f(x, y_j) +$$

$$+ \frac{y - y_j}{g_{j+1}^{(1)}(x) - y_j} R_2 f(x, g_{j+1}^{(1)}(x)),$$

де

$$R_1 f(x, y) = \int_{y_j}^{g_{j+1}^{(1)}(x)} f^{(0, r)}(x, \eta) G1(x, y, \eta) d\eta, \quad x \in [x_i, x_{i+1}],$$

$$R_2 f(x, y) = \int_{x_i}^{x_{i+1}} f^{(r, 0)}(\xi, y) G2(x, \xi) d\xi, \quad y \in [y_j, g_{j+1}^{(1)}(x)],$$

$$G1(x, y, \eta) = \begin{cases} \frac{y - g_{j+1}^{(1)}(x)}{y_j - g_{j+1}^{(1)}(x)} \frac{(y_j - \eta)^{r-1}}{(r-1)!}, & y_j \leq \xi \leq y \leq g_{j+1}^{(1)}(x), \\ \frac{y - y_j}{g_{j+1}^{(1)}(x) - y_j} \frac{(g_{j+1}^{(1)}(x) - \eta)^{r-1}}{(r-1)!}, & y_j \leq y \leq \xi \leq g_{j+1}^{(1)}(x) \end{cases}$$

$$G2(x, \xi) = \begin{cases} \frac{x - x_{i+1}}{x_i - x_{i+1}} \frac{(x_i - \xi)^{r-1}}{(r-1)!}, & x_i \leq \xi \leq x \leq x_{i+1} \\ \frac{x - x_i}{x_{i+1} - x_i} \frac{(x_{i+1} - \xi)^{r-1}}{(r-1)!}, & x_i \leq x \leq \xi \leq x_{i+1} \end{cases}$$

Доведення. Запишемо оператор інтерлінації на лініях $x = x_i$, $x = x_{i+1}$ (див. [9]):

$$S_1 f(x, y) = f(x_i, y) \frac{x - x_{i+1}}{x_i - x_{i+1}} + f(x_{i+1}, y) \frac{x - x_i}{x_{i+1} - x_i}$$

та на лініях $y = y_j$, $y = g_{j+1}^{(1)}(x)$:

$$S_2 f(x, y) = f(x, y_j) \frac{y - g_{j+1}^{(1)}(x)}{y_j - g_{j+1}^{(1)}(x)} + f(x, g_{j+1}^{(1)}(x)) \frac{y - y_j}{g_{j+1}^{(1)}(x) - y_j}.$$

Замінімо в операторах $S_1 f(x, y)$ та $S_2 f(x, y)$ сліди $f(x_i, y)$, $f(x_{i+1}, y)$, $f(x, y_j)$, $f(x, g_{j+1}^{(1)}(x))$ операторами інтерполяції

$$\tilde{S}_1 f(x, y) = \left(f(x_i, y_j) \frac{y - g_{j+1}^{(1)}(x_i)}{y_j - g_{j+1}^{(1)}(x_i)} + \right.$$

$$\left. + f(x_i, g_{j+1}^{(1)}(x_i)) \frac{y - y_j}{g_{j+1}^{(1)}(x_i) - y_j} \right) \frac{x - x_{i+1}}{x_i - x_{i+1}} +$$

$$\left(f(x_{i+1}, y_j) \frac{y - g_{j+1}^{(1)}(x_{i+1})}{y_j - g_{j+1}^{(1)}(x_{i+1})} + \right.$$

$$\left. + f(x_{i+1}, g_{j+1}^{(1)}(x_{i+1})) \frac{y - y_j}{g_{j+1}^{(1)}(x_{i+1}) - y_j} \right) \frac{x - x_i}{x_{i+1} - x_i};$$

$$\tilde{S}_2 f(x, y) = \left(f(x_i, y_j) \frac{x - x_{i+1}}{x_i - x_{i+1}} + \right.$$

$$\left. + f(x_{i+1}, y_j) \frac{x - x_i}{x_{i+1} - x_i} \right) \frac{y - g_{j+1}^{(1)}(x)}{y_j - g_{j+1}^{(1)}(x)} +$$

$$+ \left(f(x_i, g_{j+1}^{(1)}(x_i)) \frac{x - x_{i+1}}{x_i - x_{i+1}} + f(x_{i+1}, g_{j+1}^{(1)}(x_{i+1})) \frac{x - x_i}{x_{i+1} - x_i} \right) \frac{y - y_j}{g_{j+1}^{(1)}(x) - y_j}.$$

Легко побачити, що $S(x, y) = S_2 S_1 f(x, y) = (\tilde{S}_1 + \tilde{S}_2 - S_1 S_2) f(x, y)$, причому перестановність операторів відсутня, тобто $S_1 S_2 \neq S_2 S_1$.

Тепер розглянемо похибку оператора інтерполяції $S(x, y)$:

$$\begin{aligned} RS(x, y) &= (I - S)f(x, y) = (I - \tilde{S}_1 - \tilde{S}_2 + S_1 S_2)f(x, y) = \\ &= (I - \tilde{S}_1 - \tilde{S}_2 + S_1 S_2)f(x, y) + \\ &+ (S_1 + S_2 - S_1 S_2)f(x, y) - (S_1 + S_2 - S_1 S_2)f(x, y) = \\ &= (I - S_1 - S_2 + S_1 S_2)f(x, y) + (S_1 - \tilde{S}_1)f(x, y) + \\ &+ (S_2 - \tilde{S}_2)f(x, y). \end{aligned}$$

Доданок $(I - S_1 - S_2 + S_1 S_2)f(x, y)$ є залишком наближення функції $f(x, y)$ оператором інтерлінації. Згідно з теоремою 3.2.1 роботи [9] залишок наближення формулами інтерлінації виражається як операторний добуток залишків наближення функції $f(x, y)$ операторами $S_1 f(x, y)$ та $S_2 f(x, y)$, тобто

$$(I - S_1 - S_2 + S_1 S_2)f(x, y) = (f(x, y) - S_1 f(x, y))(f(x, y) - S_2 f(x, y)) = R_1 f(x, y) R_2 f(x, y),$$

де

$$R_1 f(x, y) = \int_{y_j}^{g_{j+1}^{(1)}(x)} f^{(0,r)}(x, \eta) G1(x, y, \eta) d\eta, \quad x \in [x_i, x_{i+1}],$$

$$R_2 f(x, y) = \int_{x_i}^{x_{i+1}} f^{(r,0)}(\xi, y) G2(x, \xi) d\xi, \quad y \in [y_j, g_{j+1}^{(1)}(x)],$$

$$G1(x, y, \eta) = \begin{cases} \frac{y - g_{j+1}^{(1)}(x) (y_j - \eta)^{r-1}}{y_j - g_{j+1}^{(1)}(x) (r-1)!}, & y_j \leq \xi \leq y \leq g_{j+1}^{(1)}(x), \\ -\frac{y - y_j (g_{j+1}^{(1)}(x) - \eta)^{r-1}}{g_{j+1}^{(1)}(x) - y_j (r-1)!}, & y_j \leq y \leq \xi \leq g_{j+1}^{(1)}(x) \end{cases}$$

$$G2(x, \xi) = \begin{cases} \frac{x - x_{i+1} (x_i - \xi)^{r-1}}{x_i - x_{i+1} (r-1)!}, & x_i \leq \xi \leq x \leq x_{i+1} \\ -\frac{x - x_i (x_{i+1} - \xi)^{r-1}}{x_{i+1} - x_i (r-1)!}, & x_i \leq x \leq \xi \leq x_{i+1} \end{cases}$$

$$\begin{aligned} (S_1 - \tilde{S}_1)f(x, y) &= \frac{x - x_{i+1}}{x_i - x_{i+1}} \left(f(x_i, y) - f(x_i, y_j) \frac{y - g_{j+1}^{(1)}(x_i)}{y_j - g_{j+1}^{(1)}(x_i)} + \right. \\ &\left. + f(x_i, g_{j+1}^{(1)}(x_i)) \frac{y - y_j}{g_{j+1}^{(1)}(x_i) - y_j} \right) + \end{aligned}$$

$$\begin{aligned} &+ \frac{x - x_i}{x_{i+1} - x_i} \left(f(x_{i+1}, y) - f(x_{i+1}, y_j) \frac{y - g_{j+1}^{(1)}(x_{i+1})}{y_j - g_{j+1}^{(1)}(x_{i+1})} + \right. \\ &\left. + f(x_{i+1}, g_{j+1}^{(1)}(x_{i+1})) \frac{y - y_j}{g_{j+1}^{(1)}(x_{i+1}) - y_j} \right) = \\ &= \frac{x - x_{i+1}}{x_i - x_{i+1}} R_1 f(x_i, y) + \frac{x - x_i}{x_{i+1} - x_i} R_1 f(x_{i+1}, y); \\ (S_2 - \tilde{S}_2)f(x, y) &= \frac{y - g_{j+1}^{(1)}(x)}{y_j - g_{j+1}^{(1)}(x)} \left(f(x, y_j) - \right. \\ &\left. - f(x_i, y_j) \frac{x - x_{i+1}}{x_i - x_{i+1}} + f(x_{i+1}, y_j) \frac{x - x_i}{x_{i+1} - x_i} \right) + \\ &+ \frac{y - y_j}{g_{j+1}^{(1)}(x) - y_j} \left(f(x, g_{j+1}^{(1)}(x)) - f(x_i, g_{j+1}^{(1)}(x_{i+1})) \times \right. \\ &\left. \times \frac{x - x_{i+1}}{x_i - x_{i+1}} + f(x_{i+1}, g_{j+1}^{(1)}(x_{i+1})) \frac{x - x_i}{x_{i+1} - x_i} \right) = \\ &= \frac{y - g_{j+1}^{(1)}(x)}{y_j - g_{j+1}^{(1)}(x)} R_2 f(x, y_j) + \frac{y - y_j}{g_{j+1}^{(1)}(x) - y_j} R_2 f(x, g_{j+1}^{(1)}(x)). \end{aligned}$$

Теорема 2 доведена.

Теорема 3. Оцінка похибки наближення функції $f(x, y)$ побудованим розривним інтерполяційним сплайном $S(x, y) = S_{ij}(x, y)$ на кожній трапеції має вигляд

$$|f(x, y) - S(x, y)| \leq Q,$$

$$Q = \|f^{(2,2)}(x, y)\|_{L_\infty[x_i, x_{i+1}] \times [y_j, g_{j+1}(x)]} \times \frac{(\Delta_x)^2}{64} \times \max\{(\Delta 1_y)^2, (\Delta 2_y)^2\} +$$

$$+ \max\left\{ \|f^{(0,2)}(x_i, y)\|_{L_\infty[y_j, g_{j+1}(x)]} \times \frac{(\Delta 1_y)^2}{8}, \right.$$

$$\left. \|f^{(0,2)}(x_{i+1}, y)\|_{L_\infty[y_j, g_{j+1}(x)]} \times \frac{(\Delta 2_y)^2}{8} \right\} +$$

$$+ \max\left\{ \|f^{(2,0)}(x, y_j)\|_{L_\infty[x_i, x_{i+1}]} \times \frac{(\Delta_x)^2}{8}, \right.$$

$$\left. \|f^{(2,0)}(x, g_{j+1}(x))\|_{L_\infty[x_i, x_{i+1}]} \times \frac{(\Delta_x)^2}{8} \right\},$$

$$\Delta_x = x_{i+1} - x_i, \quad \Delta 1_y = g_{j+1}(x_i) - y_j,$$

$$\Delta 2_y = g_{j+1}(x_{i+1}) - y_j.$$

Доведення.

$$|f(x, y) - S(x, y)| =$$

$$= \left| R_1 R_2 f(x, y) + \frac{x - x_{i+1}}{x_i - x_{i+1}} R_1 f(x_i, y) + \right.$$

$$\left. + \frac{x - x_i}{x_{i+1} - x_i} R_1 f(x_{i+1}, y) + \frac{y - g_{j+1}(x)}{y_j - g_{j+1}(x)} R_2 f(x, y_j) + \right.$$

$$\begin{aligned}
 & + \left| \frac{y - y_j}{g_{j+1}(x) - y_j} R_2 f(x, g_{j+1}(x)) \right| \leq |R_1 R_2 f(x, y)| + \\
 & + \left| \frac{x - x_{i+1}}{x_i - x_{i+1}} R_1 f(x_i, y) + \frac{x - x_i}{x_{i+1} - x_i} R_1 f(x_{i+1}, y) \right| + \\
 & + \left| \frac{y - g_{j+1}(x)}{y_j - g_{j+1}(x)} R_2 f(x, y_j) + \frac{y - y_j}{g_{j+1}(x) - y_j} R_2 f(x, g_{j+1}(x)) \right|.
 \end{aligned}$$

Користуючись оцінками з роботи [5], отримаємо

$$1) \quad |R_1 R_2 f(x, y)| \leq \|f^{(2,2)}(x, y)\|_{L_\infty[x_i, x_{i+1}] \times [y_j, g_{j+1}(x)]} \times \frac{(x_{i+1} - x_i)^2 (y_j - g_{j+1}(x))^2}{8 \cdot 8};$$

$$2) \quad \left| \frac{x - x_{i+1}}{x_i - x_{i+1}} R_1 f(x_i, y) + \frac{x - x_i}{x_{i+1} - x_i} R_1 f(x_{i+1}, y) \right| \leq \max \left\{ \frac{x - x_{i+1}}{x_i - x_{i+1}} \|f^{(0,2)}(x_i, y)\|_{L_\infty[y_j, g_{j+1}(x)]} \times \frac{(\Delta 1_y)^2}{8}, \frac{x - x_i}{x_{i+1} - x_i} \|f^{(0,2)}(x_{i+1}, y)\|_{L_\infty[y_j, g_{j+1}(x)]} \times \frac{(\Delta 2_y)^2}{8} \right\};$$

$$\left| \frac{y - g_{j+1}(x)}{y_j - g_{j+1}(x)} R_2 f(x, y_j) + \frac{y - y_j}{g_{j+1}(x) - y_j} R_2 f(x, g_{j+1}(x)) \right| \leq \max \left\{ \frac{y - g_{j+1}(x)}{y_j - g_{j+1}(x)} \|f^{(2,0)}(x, y_j)\|_{L_\infty[x_i, x_{i+1}]} \times \frac{(\Delta x)^2}{8}, \frac{y - y_j}{g_{j+1}(x) - y_j} \|f^{(2,0)}(x, g_{j+1}(x))\|_{L_\infty[x_i, x_{i+1}]} \times \frac{(\Delta x)^2}{8} \right\}.$$

Теорема 3 доведена.

2. Побудова розривного апроксимаційного сплайна

Визначення. Будемо називати розривним апроксимаційним поліноміальним сплайном в області $TR_{ij} \subset D$ функцію (1), в якій коефіцієнти $C_{ij}^k, i = \overline{1, n}, j = \overline{1, m}, k = \overline{1, 4}$ сплайна $S(x, y)$ знаходяться методом найменших квадратів з умови

$$\iint_{TR_{ij}} (f(x, y) - S(x, y))^2 dx dy \rightarrow \min_C. \quad (2)$$

Теорема 4. Для оператора наближення розривної функції $f(x, y) \in C^{(2,2)}(TR_{ij})$ розривним апроксимаційним сплайном $S(x, y)$ вигляду (1), побудованого за допомогою методу найменших квадратів, на кожному елементі розбиття $TR_{ij}, i = \overline{1, n}, j = \overline{1, m}$, справедлива наступна оцінка:

$$\|Sp(x, y)\|_\infty \leq \max\{|f(x_i, y_j)|, |f(x_{i+1}, y_j)|, |f(x_i, g_{j+1}(x_i))|, |f(x_{i+1}, g_{j+1}(x_{i+1}))|\} + Q,$$

де Q визначається в теоремі 3.

Доведення. Проведемо доведення на прикладі трапеції $TR_{ij}^{(1)}$. Тоді, позначаючи $g(x) = g_{j+1}^{(1)}(x)$, формула (1) перетвориться у наступний вираз

$$\begin{aligned}
 S(x, y) = & C_1^{(1)} \frac{x - x_{i+1}}{x_i - x_{i+1}} \frac{y - g(x)}{y_j - g(x)} + \\
 & + C_2^{(1)} \frac{x - x_i}{x_{i+1} - x_i} \frac{y - g(x)}{y_j - g(x)} + \\
 & + C_3^{(1)} \frac{y - y_j}{g(x_i) - y_j} \frac{x - x_{i+1}}{x_i - x_{i+1}} + \\
 & + C_4^{(1)} \frac{y - y_j}{g(x_{i+1}) - y_j} \frac{x - x_i}{x_{i+1} - x_i}.
 \end{aligned}$$

Розв'яжемо мінімізаційну задачу:

$$\begin{aligned}
 P_{ij}(C) = & \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} \left(C_1^{(1)} \frac{x - x_{i+1}}{x_i - x_{i+1}} \frac{y - g(x)}{y_j - g(x)} + \right. \\
 & + C_2^{(1)} \frac{x - x_i}{x_{i+1} - x_i} \frac{y - g(x)}{y_j - g(x)} + \\
 & + C_3^{(1)} \frac{y - y_j}{g(x_i) - y_j} \frac{x - x_{i+1}}{x_i - x_{i+1}} + \\
 & \left. + C_4^{(1)} \frac{y - y_j}{g(x_{i+1}) - y_j} \frac{x - x_i}{x_{i+1} - x_i} - f(x, y) \right)^2 dx dy \rightarrow \min_C.
 \end{aligned}$$

Вираз у дужках позначимо через $L(x, y, C)$.

Випишемо систему лінійних алгебраїчних рівнянь

$$\frac{\partial P_{ij}(C)}{\partial C_1^{(1)}} = 0, \frac{\partial P_{ij}(C)}{\partial C_2^{(1)}} = 0, \frac{\partial P_{ij}(C)}{\partial C_3^{(1)}} = 0, \frac{\partial P_{ij}(C)}{\partial C_4^{(1)}} = 0$$

відносно невідомих $C_k^{(1)}, k = 1, 2, 3, 4$:

$$\begin{cases} \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} (S(x, y) - f(x, y)) \frac{x - x_{i+1}}{x_i - x_{i+1}} \frac{y - g(x)}{y_j - g(x)} dy dx = 0 \\ \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} (S(x, y) - f(x, y)) \frac{x - x_i}{x_{i+1} - x_i} \frac{y - g(x)}{y_j - g(x)} dy dx = 0 \\ \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} (S(x, y) - f(x, y)) \frac{y - y_j}{g(x_i) - y_j} \frac{x - x_{i+1}}{x_i - x_{i+1}} dy dx = 0 \\ \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} (S(x, y) - f(x, y)) \frac{y - y_j}{g(x_{i+1}) - y_j} \frac{x - x_i}{x_{i+1} - x_i} dy dx = 0 \end{cases} \quad (3)$$

У системі зробимо заміну

$$\begin{aligned}
 C_1^{(1)} & = f(x_i + 0, y_j + 0) + \varepsilon_{i,j}, \\
 C_2^{(1)} & = f(x_{i+1} - 0, y_j + 0) + \varepsilon_{i+1,j}, \\
 C_3^{(1)} & = f(x_i + 0, g(x_i) - 0) + \varepsilon_{i,j+1}, \\
 C_4^{(1)} & = f(x_{i+1} - 0, g(x_{i+1}) - 0) + \varepsilon_{i+1,j+1}
 \end{aligned}$$

і замінимо $f(x, y)$ інтерполяційним сплайном, побудованим на трапецевидному елементі $TR_{ij}^{(1)}$ із залишковим членом $R(x, y)$, який був виведений

у роботі [9]. В результаті отримаємо наступні вирази для інтегральних членів отриманої системи, приймаючи до уваги, що $g(x) = ax^2 + bx + c$:

$$\begin{aligned} & \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} \left(\frac{x - x_{i+1}}{x_i - x_{i+1}} \right)^2 \left(\frac{y - g(x)}{y_j - g(x)} \right)^2 dx dy = \Delta_x \times \\ & \times \frac{(12ax_i^2 + 6ax_i x_{i+1} + 15bx_i + 2ax_{i+1}^2 + 5bx_{i+1} - 20y_j + 20c)}{180}; \\ & \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} \frac{(x - x_i)(x - x_{i+1})}{(x_{i+1} - x_i)^2} \left(\frac{y - g(x)}{y_j - g(x)} \right)^2 dx dy = \Delta_x \times \\ & \times \frac{(3ax_i^2 + 4ax_i x_{i+1} + 5bx_i + 3ax_{i+1}^2 + 5bx_{i+1} - 10y_j + 10c)}{180}; \\ & \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} \left(\frac{x - x_{i+1}}{x_i - x_{i+1}} \right)^2 \frac{(y - y_j)(y - g(x))}{(g(x) - y_j)^2} dx dy = -\Delta_x \times \\ & \times \frac{(12ax_i^2 + 6ax_i x_{i+1} + 15bx_i + 2ax_{i+1}^2 + 5bx_{i+1} - 20y_j + 20c)}{360}; \\ & \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} \frac{(x - x_i)(x - x_{i+1})}{(x_{i+1} - x_i)^2} \frac{(y - y_j)(y - g(x))}{(g(x) - y_j)^2} dy dx = \Delta_x \times \\ & \times \frac{(3ax_i^2 + 4ax_i x_{i+1} + 5bx_i + 3ax_{i+1}^2 + 5bx_{i+1} - 10y_j + 10c)}{360}; \\ & \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} \left(\frac{x - x_i}{x_{i+1} - x_i} \right)^2 \left(\frac{y - g(x)}{y_j - g(x)} \right)^2 dx dy = \Delta_x \times \\ & \times \frac{(2ax_i^2 + 6ax_i x_{i+1} + 5bx_i + 12ax_{i+1}^2 + 15bx_{i+1} - 20y_j + 20c)}{180}; \\ & \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} \left(\frac{x - x_i}{x_{i+1} - x_i} \right)^2 \frac{(y - y_j)(y - g(x))}{(g(x) - y_j)^2} dx dy = \Delta_x \times \\ & \times \frac{(2ax_i^2 + 6ax_i x_{i+1} + 5bx_i + 12ax_{i+1}^2 + 15bx_{i+1} - 20y_j + 20c)}{360}; \\ & \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} \left(\frac{x - x_{i+1}}{x_i - x_{i+1}} \right)^2 \left(\frac{y - y_j}{g(x) - y_j} \right)^2 dx dy = \Delta_x \times \\ & \times \frac{(12ax_i^2 + 6ax_i x_{i+1} + 15bx_i + 2ax_{i+1}^2 + 5bx_{i+1} - 20y_j + 20c)}{180}; \\ & \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} \frac{(x - x_i)(x - x_{i+1})}{(x_{i+1} - x_i)^2} \left(\frac{y - y_j}{g(x) - y_j} \right)^2 dx dy = -\Delta_x \times \\ & \times \frac{(3ax_i^2 + 4ax_i x_{i+1} + 5bx_i + 3ax_{i+1}^2 + 5bx_{i+1} - 10y_j + 10c)}{180}; \\ & \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} \left(\frac{x - x_i}{x_{i+1} - x_i} \right)^2 \left(\frac{y - y_j}{g(x) - y_j} \right)^2 dx dy = \Delta_x \times \\ & \times \frac{(2ax_i^2 + 6ax_i x_{i+1} + 5bx_i + 12ax_{i+1}^2 + 15bx_{i+1} - 20y_j + 20c)}{180}. \end{aligned}$$

Для аналізу доданків, до складу яких входить залишковий член $R(x, y)$, скористаємося формулою з роботи [9]:

$$|f(x, y) - S(x, y)| \leq Q,$$

де Q визначається в теоремі 3. Тоді отримаємо:

$$\begin{aligned} & \left| \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} R(x, y) \frac{x - x_{i+1}}{x_i - x_{i+1}} \frac{y - g(x)}{y_j - g(x)} dy dx \right| \leq Q \cdot \Delta_x \times \\ & \times \frac{(3ax_i^2 + 2ax_i x_{i+1} + 4bx_i + ax_{i+1}^2 + 2bx_{i+1} - 6y_j + 6c)}{24}; \\ & \left| \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} R(x, y) \frac{x - x_i}{x_{i+1} - x_i} \frac{y - g(x)}{y_j - g(x)} dy dy \right| \leq Q \cdot \Delta_x \times \\ & \times \frac{(ax_i^2 + 2ax_i x_{i+1} + 2bx_i + 3ax_{i+1}^2 + 4bx_{i+1} - 6y_j + 6c)}{24}; \\ & \left| \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} R(x, y) \frac{x - x_{i+1}}{x_i - x_{i+1}} \frac{y - y_j}{g(x) - y_j} dy dx \right| \leq Q \cdot \Delta_x \times \\ & \times \frac{(3ax_i^2 + 2ax_i x_{i+1} + 4bx_i + ax_{i+1}^2 + 2bx_{i+1} - 6y_j + 6c)}{24}; \\ & \left| \int_{x_i}^{x_{i+1}} \int_{y_j}^{g(x)} R(x, y) \frac{x - x_i}{x_{i+1} - x_i} \frac{y - y_j}{g(x) - y_j} dy dx \right| \leq Q \cdot \Delta_x \times \\ & \times \frac{(ax_i^2 + 2ax_i x_{i+1} + 2bx_i + 3ax_{i+1}^2 + 4bx_{i+1} - 6y_j + 6c)}{24}. \end{aligned}$$

Використовуючи позначення

$$\|\varepsilon\| = \max\{\varepsilon_{i,j}, \varepsilon_{i+1,j}, \varepsilon_{i,j+1} + \varepsilon_{i+1,j+1}\}$$

та спростивши отримані вирази, рівняння системи (3) будуть мати однаковий вигляд

$$\begin{aligned} & \frac{(3ax_i^2 + 2ax_i x_{i+1} + 4bx_i + ax_{i+1}^2 + 2bx_{i+1} - 6y_j + 6c)}{24} \|\varepsilon\| \leq \\ & \leq Q \cdot \frac{(3ax_i^2 + 2ax_i x_{i+1} + 4bx_i + ax_{i+1}^2 + 2bx_{i+1} - 6y_j + 6c)}{24}. \end{aligned}$$

Тобто $\|\varepsilon\| \leq Q$.

Теорема доведена.

Наслідок. Якщо наближувана функція $f(x, y)$ є кусково-лінійною або кусково-сталою функцією в кожному трапецевидному елементі розбиття з точками розриву (x_i, y_j) , $i = \overline{1, m}$, $j = \overline{1, n}$ у випадку наближення її кусково-лінійним сплайном $S(x, y)$, визначеним формулами (1) з невідомими $C_m^{(k)}$, $m = \overline{1, 4}$, $k = \overline{1, 8}$, що знаходяться з умови (2), отримаємо точно наближувану функцію, тобто $S(x, y) = f(x, y)$, де $f(x, y) = A(const)$ або $f(x, y) = A_0 + A_1x + A_2y + A_3xy$.

Зауваження. Якщо $C_1^{(1)} = C_1^{(2)} = C_1^{(3)} = C_1^{(4)} = S(x_i, y_j)$, $i = \overline{1, m}$, $j = \overline{1, n}$ або $C_1^{(5)} = C_1^{(6)} = C_1^{(7)} = C_1^{(8)} = S(x_i, y_j)$, то побудований розривний

апроксимаційний сплайн вигляду (1) є неперервним лінійним апроксимаційним сплайном.

Приклад. Нехай функція задана на одиничному квадраті $[0,1] \times [0,1]$ (рис. 2)

$$f(x,y) = \begin{cases} x+y, & \text{если} \\ & 0,5 < x < 1; 0,5 < y < (x-1)^2 + 0,7; \\ 1,5 - 4x^2 - y^2, & \text{если} \\ & 0 < x < 0,5; 0,5 < y < -(x-0,5)^2 + 0,95; \\ 0,5, & \text{если} \\ & 0 < x < 0,5; (x-0,5)^2 + 0,05 < y < 0,5; \\ 1-x+y^2, & \text{если} \\ & 0,5 < x < 1; -(x-1)^2 + 0,3 < y < 0,5. \end{cases}$$

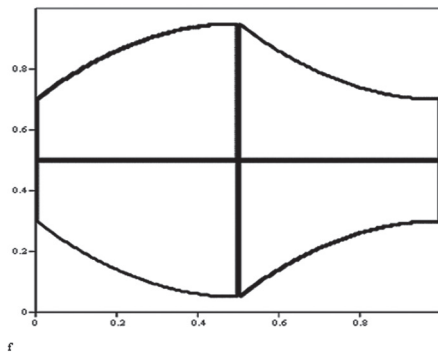
Тобто на лініях фігури, зображеної на рис. 2а), функція $f(x,y)$ має розриви першого роду. Нехай задані лінії:

$$x_1 = 0; x_2 = 0,5; x_3 = 1,$$

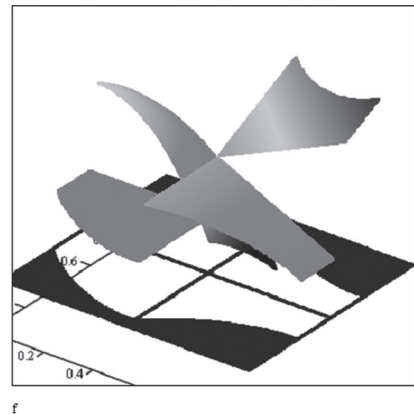
$$y_1 = 0; y_2^1 = (x-0,5)^2 + 0,05; y_2^2 = -(x-1)^2 + 0,3,$$

$$y_3 = 0,5; y_3^1 = (x-1)^2 + 0,7; y_3^2 = -(x-0,5)^2 + 0,95.$$

Вони розбивають область визначення функції $f(x,y)$ на вісім трапецевидних елементів з однією криволінійною стороною у кожному елементі.



а



б

Рис. 2. Графічне зображення: а – області визначення функції $f(x,y)$; б – функції $f(x,y)$

Спочатку побудуємо розривний інтерполяційний сплайн вигляду (1), його графік наведений на рис.3а). Визначимо максимальне відхилення наближуваної функції $f(x,y)$ від побудованого сплайну $S(x,y)$:

$$\max|f(x,y) - S(x,y)| = 0,3.$$

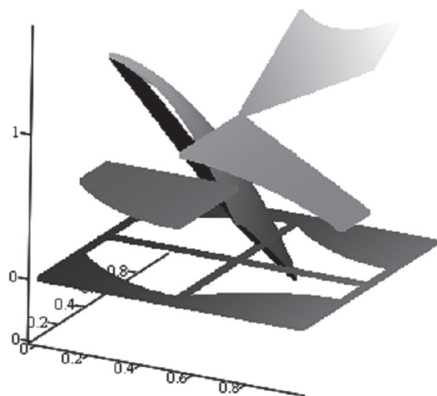
Тепер побудуємо розривний апроксимаційний сплайн за формулами (1), коефіцієнти якого знаходяться з умови (2). Графічне зображення цього сплайну наведено на рис. 3б). Визначимо максимальне відхилення наближуваної функції $f(x,y)$ від побудованого сплайну $S(x,y)$:

$$\max|f(x,y) - S(x,y)| \approx 0,08.$$

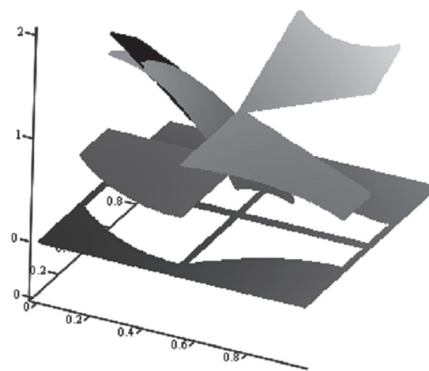
Як бачимо, побудований розривний апроксимаційний сплайн наближує розривну функцію краще, ніж інтерполяційний. Побудовані розривні сплайни точно наближують ту частину функції, де вона є постійною або лінійною, що і підтверджує викладену вище теорію.

Висновки

В роботі пропонується метод побудови розривного інтерполяційного та апроксимаційного лінійного сплайнів для наближення функції з розривами



а



б

Рис. 3. Графічний вигляд розривного: а – інтерполяційного, б – апроксимаційного сплайнів (чорний колір) та заданої функції (сірий колір)

першого роду, область визначення яких розбита на криволінійні трапеції. Причому побудовані розривні сплайни включають в себе, як частинний випадок, класичні неперервні сплайни першого степеня на заданій сітці вузлів.

У подальшому авторами планується розробити методи наближення розривних функцій розривними сплайнами, коли розриви наближуваної функції не падають з вузлами наближувачого сплайну. А також планується застосувати розроблену теорію наближення розривних функцій розривними сплайнами до розв'язання двовимірної задачі комп'ютерної томографії.

Список літератури: 1. Корнейчук, Н.П. Сплайны в теории приближения [Текст] / Н.П. Корнейчук. – М: Наука, 1984. – 352 с. 2. Сьярле, Ф. Метод конечных элементов для эллиптических задач [Текст] / Ф.Сьярле. – Изд-во “Мир”, Москва, 1980. – 512 с. 3. Попов, Б. А. Равномерное приближение сплайнами [Текст] / Б.А. Попов. – К.: Наук. думка, 1989. – 272 с. 4. Петровская, Н.Б. Аппроксимация разрывных решений для одного класса схем высокого порядка [Текст] / Н.Б. Петровская // Математическое моделирование. – Москва. – 2005. – Т. 17, №1. – С. 79–92. 5. Arnold D.N. Unified analysis of discontinuous Galerkin methods for elliptic problems [Text] / Arnold D.N. // SIAM Journal on Numerical Analysis. – 2002. – Vol. 39, № 5. – P. 1749-1779. 6. Литвин, О.М. Побудова кусково-білінійних сплайнів для наближення функцій з розривами першого роду у вузлах ректангуляції двовимірної області [Текст] / О.М. Литвин, Ю.І. Першина // Таврічний вісник інформатики та математики. – Симферополь. – 2011. – №1. – С. 63–72. 7. Литвин, О.Н. Приближение разрывной функции двух переменных с помощью разрывных сплайнов двух переменных (прямоугольные элементы) [Текст] / О.Н. Литвин, Ю.И. Першина // Компьютерная математика. – Киев. – 2011. – №1. – С. 96–105. 8. Литвин, О.М. Приближение разрывных функций двух переменных с разрывами первого рода на линиях триангуляции

двумерной области [Текст] / О.М. Литвин, Ю.І. Першина // Управляющие системы и машины. – Киев, – 2011, № 5. – С. 34–47. 9. Литвин, О.М. Інтерлінація функцій та деякі її застосування [Текст] / О.М. Литвин. – Х: Основа, 2002. – 544 с.

Надійшла до редколегії 27.01.2012

УДК 519.6

Восстановление объектов, которые описываются разрывными функциями, с использованием криволинейных трапеций / О.Н. Литвин, Ю.И. Першина // Бионика интеллекта: науч.-техн. журнал. – 2012. – № 1 (78). – С. 37-44.

Построены разрывные интерполяционные и аппроксимационные сплайны для приближения разрывных функций, область определения которых разбивается на криволинейные трапеции. Сформулированы и доказаны теоремы об общем виде погрешности приближения интерполяционным сплайном и об оценке погрешности приближения. Также в работе представлена оценка построенного аппроксимационного разрывного сплайна. И показано, что построенные разрывные конструкции включают в себя, как частный случай, классические неперервные сплайны.

Ил. 3. Библиогр.: 9 назв.

UDK 519.6

Restoration of objects which are described by discontinuous functions with use of curvilinear trapezes / O.N. Lytvyn, Y.I. Pershina // Bionics of Intelligense: Sci. Mag. – 2012. – № 1 (78). – P. 37-44.

Are constructed discontinuous interpolational and approximal splines for approach of the discontinuous functions which range of definition breaks into curvilinear trapezes. Theorems of a general view of an error approach by ineprolational spline and about an estimation of an error of approach are formulated and proved As in work the estimation constructed approximal discontinuous spline is presented. Also it is shown, that the constructed discontinuous designs include classical continuous splines as a special case.

Fig. 3. Ref.: 9 items.



О.Г. Руденко, М.С. Сныткин

Харьковский национальный университет радиоэлектроники,
г. Харьков, Украина

СЖАТИЕ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ FUZZY ART

Рассматривается JPEG-подобный алгоритм сжатия видеоинформации на основе нейронной сети Fuzzy ART, осуществляющей операцию векторного квантования. Результаты моделирования предлагаемого алгоритма в среде Matlab свидетельствуют о возможности применения данного алгоритма для сжатия изображения и целесообразности его использования при сжатии изображений, имеющих повторяющиеся участки.

АЛГОРИТМ СЖАТИЯ, ВЕКТОРНОЕ КВАНТОВАНИЕ, ДИСКРЕТНО-КОСИНУСНОЕ ПРЕОБРАЗОВАНИЕ, НЕЙРОННАЯ СЕТЬ FUZZY ART

Введение

Увеличение объемов зрительной информации, хранимой, передаваемой и обрабатываемой в цифровом виде, обуславливает потребность в ее компактном представлении. В связи с этим возникает необходимость постоянного совершенствования алгоритмов и методов обработки информации.

Одним из перспективных представляется развитие подхода, в основе которого лежат искусственные нейронные сети (ИНС). При этом ИНС могут использоваться как при сжатии без потерь (например, в статистических методах кодирования для оценки вероятностей появления символов [1]), так и при реализации сжатия с потерями. В последнем случае целесообразно применение ИНС, осуществляющих векторное квантование.

В данной статье рассматривается применение ИНС на основе теории адаптивного резонанса – Fuzzy ART для развития JPEG-подобных алгоритмов сжатия видеоинформации.

1. Описание схемы сжатия

Общая схема сжатия видеоинформации состоит из таких основных шагов: дискретно-косинусное преобразование, векторное квантование, дифференциальное кодирование, энтропийное кодирование.

Дискретно-косинусное преобразование (ДКП) по сути представляет собой разложения участков изображения по амплитудам некоторых частот. Для сокращения объема хранимой информации используется тот факт, что в матрице частот многие коэффициенты либо близки, либо равны нулю [2].

Векторное квантование, или квантование коэффициентов частотной матрицы, представляет собой операцию, которая определяет степень сжатия и потери информации, увеличивает количество одинаковых и равных нулю элементов. При этом из пространства выбирается ограниченное число векторов (кодовых слов) для наиболее точного отображения распределения исходных векторов, полученных из представленного в виде блоков

изображения. Таким образом, большее количество кодовых слов будет расположено в той области пространства, где находится больше точек в исходном распределении (изображении) и наоборот. Затем каждый вектор исходного изображения заменяется ближайшим кодовым словом. На последнем этапе индекс кодового слова, содержащий меньшее количество бит, чем количество бит в исходном блоке, передается вместо самого слова.

Дифференциальное кодирование. Если предположить, что большинство частей изображения имеют плавные переходы, то дифференциальное кодирование, примененное после векторного квантования, приведет к «уменьшению» кодов в целом.

Энтропийное кодирование – кодирование словами (кодами) переменной длины, при которой длина кода символа имеет обратную зависимость от вероятности появления символа в передаваемом сообщении [3].

Кодирование длин серий является очень простой формой сжатия данных, в которой серии данных, т. е. последовательности, в которых одно и то же значение данных встречается во многих последовательных элементах данных, сохраняются в виде одиночного значения данных и их количества вместо оригинальной серии. Это сжатие часто используется для данных, содержащих много таких серий (например, простые графические изображения, такие как иконки и графические рисунки).

2. Сеть Fuzzy ART

Сеть Fuzzy ART является расширением сети ART1 путем применения теории нечетких множеств, что позволяет новой сети работать как с бинарными, так и с аналоговыми входными образами [4]. Для Fuzzy ART основные фазы классификации следующие.

Предварительная обработка. Все величины входного образа должны быть в интервале $[0,1]$

$$i_k \in [0,1] \quad \forall k. \quad (1)$$

Распознавание. Восходящая сетевая активность, ведущая к предварительному выбору прототипа,

определяется с использованием нечеткой конъюнкции (\wedge), по формулам:

$$\begin{aligned} x \wedge y &= \min\{x, y\} \\ X \wedge Y &= (x_1 \wedge y_1, \dots, x_m \wedge y_m), \end{aligned} \quad (2)$$

где Y – нечеткое подмножество X , если $X \wedge Y = Y$. Размер вектора ($|X|$) определяется его нормой L1, т. е. суммой его элементов.

Активность t_j , каждого нейрона можно рассматривать как степень принадлежности прототипа W_j нечеткому подмножеству входного образа I

$$t_j = \frac{|I \wedge W_j|}{\alpha + |W_j|}, \quad (3)$$

где $\alpha = const$ – величина, играющая регуляризующую роль, т.е. предотвращающая возникновение переполнения при операции деления при $|W_j| \rightarrow 0$.

Сравнение. Сходство между входом I и победившим прототипом W_j определяется степенью принадлежности образа I нечеткому подмножеству W_j . Резонанс и адаптация происходят, если

$$\rho \leq \frac{|I \wedge W_j|}{|I|}. \quad (4)$$

Адаптация. Адаптация победившего прототипа W_j происходит путем изменения его компонентов по отношению к вектору $I \wedge W_j$:

$$W_j^{(new)} = \eta \cdot (I \wedge W_j^{(old)}) + (1 - \eta) \cdot W_j^{(old)}, \quad (5)$$

где $\eta \in [0, 1]$ – показатель обучения, определяющий скорость сходимости прототипов к общему минимуму значений элементов всех входных образов, принадлежащих одному классу.

Сеть ART может работать в режиме классификации, если для предварительно обученной сети установить $\eta = 0$, что предотвратит модификацию прототипов новыми входными образами. Начальная инициализация прототипов выполняется постоянной величиной

$$w_{ij} \geq 1 \quad \forall i. \quad (6)$$

Таким образом обеспечивается поиск сначала среди фиксированных прототипов, а затем - среди остальных. Часто используемый метод ускорения обучения в сетях ART это установка коэффициента обучения $\eta = 1$, когда прежде неиспользованный прототип адаптируется к текущему входному вектору. Входной вектор I становится первым прототипом в новом классе, если другие ранее сформированные прототипы не подходят. Однако уже сформированные прототипы должны адаптироваться более медленно ($\eta < 1$), чтобы предотвратить их искажение зашумленными входными образами.

Дополнительное кодирование. В сети Fuzzy ART существует проблема кластерного распространения, состоящая в том, что поскольку векторные

элементы прототипов после адаптации только уменьшаются, сеть стремится создавать больше прототипов, которые соответствуют входным образам с большими значениями входных величин, тогда как прототипы с малыми значениями могут никогда не быть доступны. Это устраняется путем нормализации, например, путем нормализации входных образов.

Обычно используется модифицированный вариант нормализации, называемый дополнительным кодированием, который преобразовывает все входные образы к одинаковой длине вектора. При этом оригинальный вектор $A = (a_1, \dots, a_k)$ кодируется во входной образ $I = (i_1, \dots, i_m)$ с добавлением своих дополнительных элементов к оригинальному вектору. Это удваивает длину всех входных образов и прототипов

$$I = (A, A^C) = (a_1, \dots, a_k, 1 - a_1, \dots, 1 - a_k) \quad a_i \in [0, 1] \quad \forall i.$$

Норма L1 векторов, закодированных этим методом и имеющих одинаковую длину, является величиной постоянной, независимой от величин элементов

$$\begin{aligned} |I| &= \sum_{i=1}^{2k} i_i = \sum_{i=1}^k a_i + \sum_{i=1}^k (1 - a_i) = \\ &= \sum_{i=1}^k a_i + k - \sum_{i=1}^k a_i = k = m / 2. \end{aligned}$$

Использование дополнительного кодирования упрощает выражение (4):

$$\rho \leq \frac{|I \wedge W_j|}{k}.$$

Начальная инициализация прототипов происходит в соответствии с (6). При работе в режиме быстрого обучения ($\eta \rightarrow 1$) прототип W_j в Fuzzy ART является общим MIN-вектором для всех входных образов I_p , $p = (1, \dots, l)$, входящих в один и тот же класс j , и может быть представлен в таком виде:

$$\begin{aligned} W_j &= (I_1, I_2, \dots, I_l) = \\ &= (\min\{i_{11}, \dots, i_{l1}\}, \dots, \min\{i_{1m}, \dots, i_{lm}\}). \end{aligned}$$

Использование дополнительного кодирования входных образов $I_p = (A_p, A_p^C)$ приводит к прототипам, представляющим общий MIN- и MAX- вектор всех некодированных образов A_p :

$$\begin{aligned} W_j &= (\min\{A_1, \dots, A_l\}, \min\{A_1^C, \dots, A_l^C\}) = \\ &= (\min\{A_1, \dots, A_l\}, \max\{A_1, \dots, A_l\}^C). \end{aligned}$$

При малых значениях η сетевые прототипы сходятся более медленно к этим MIN и MAX границам.

3. Алгоритм сжатия

Алгоритм сжатия включает следующие шаги:

1. Условное разбиение исходного изображения на квадратные блоки размером $n \times n$ пикселей, например, 4×4 или 8×8 .

2. Применение к каждому блоку дискретно-косинусного преобразования.

3. Представление каждого блока в виде вектора в 16- или 64-мерном пространстве соответственно.

4. Низкочастотная фильтрация (исключение из векторов коэффициентов, соответствующих высокочастотным составляющим).

5. Обучение нейронной сети ART.

6. Считывание индексов нейронов, соответствующих каждому входному вектору.

7. Формирование таблицы соответствия между индексом и усредненным вектором (кластерным центром).

8. Сжатие последовательности индексов с использованием кодирования длин серий и алгоритма Хаффмана.

Как следует из описания алгоритма, в нем, в отличие от JPEG, применена сеть ART (шаги 5-7), результаты работы которой используются для векторного квантования.

Для декодирования используются соответствующие операции, но в обратном порядке.

4. Моделирование

При моделировании в среде Matlab процесса сжатия для оценки качества сжатия использовались показатели PSNR (peak signal-to-noise ratio) – отношение максимально возможного уровня сигнала к уровню искажающего его шума (поскольку большинство сигналов имеют очень широкий динамический диапазон, PSNR обычно представляют в логарифмическом масштабе), и MSE (mean squared error) – среднеквадратичная ошибка.

Для двух монохромных изображений I и K размерностью $m \times n$ (где одно из изображений является зашумленным представлением второго) MSE вычисляется следующим образом:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2.$$

Для цветных изображений с тремя RGB компонентами MSE определяется как сумма всех квадратичных разностей, деленная на размер изображения и на 3.

Показатель PSNR определяется так:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right),$$

где MAX_I представляет собой максимальное значение пикселя в изображении. В случае, если пиксели представлены 8-ми битовыми значениями, $MAX_I = 255$. В общем случае при использовании для представления B бит максимально возможное значение для MAX_I равно $2^B - 1$.

Обычно для алгоритмов сжатия PSNR находится в диапазоне 30-40 dB.

Для моделирования процесса сжатия были использованы блоки размером 8×8 точек;

квантование осуществлялось при помощи сети Fuzzy ART. Для иллюстрации полученных результатов используется общепринятое изображение «Lena». В табл. 1 представлены результаты моделирования предложенного алгоритма, а также, для сравнения результатов сжатия с другими алгоритмами, представлены данные для алгоритмов JPEG и ART2 взятые из [5].

На рис. 1 и 2 приведены обобщенные графики зависимости PSNR и MSE от степени сжатия (CR) для всех алгоритмов.

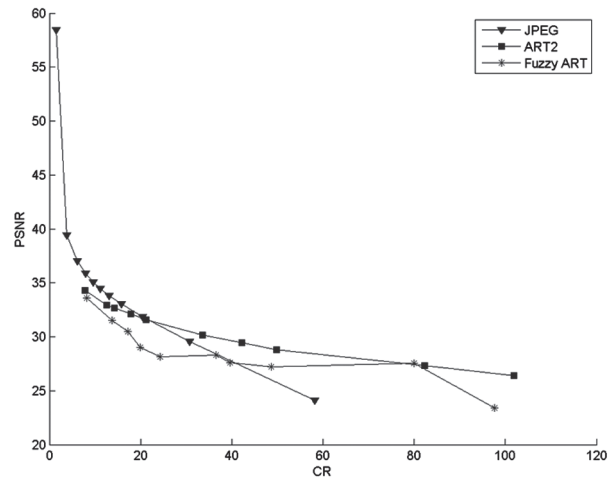


Рис. 1. Зависимости MSE от степени сжатия

Таблица 1

	Q/ρ	CR	PSNR	MSE
JPEG	0	58,215	24,110	252,389
	10	30,692	29,566	71,851
	20	20,403	31,848	42,490
	30	15,750	33,042	32,274
	40	13,078	33,814	27,023
	50	11,178	34,444	23,371
	60	9,611	35,059	20,284
	70	7,863	35,889	16,755
	80	6,073	37,046	12,837
	100	1,481	58,440	0,093
ART2	0,950	101,893	26,395	149,127
	0,960	82,319	27,356	119,519
	0,970	49,762	28,821	85,298
	0,975	42,235	29,467	73,513
	0,980	33,552	30,184	62,331
	0,984	21,209	31,590	45,090
	0,986	17,818	32,113	39,971
	0,988	14,296	32,691	34,990
	0,989	12,602	32,931	33,110
	0,990	7,853	34,330	23,994
Fuzzy ART	0,970	97,633	23,396	297,49
	0,980	79,983	27,525	114,97
	0,982	48,626	27,211	123,58
	0,984	39,585	27,613	112,66
	0,986	36,596	28,296	96,263
	0,988	24,274	28,166	99,186
	0,990	19,964	29,021	81,473
	0,992	17,191	30,475	58,29
0,994	13,678	31,528	45,739	
0,996	8,1339	33,596	28,409	

На рис. 3 показан пример сжатия изображения «Lena» с помощью сети Fuzzy ART при задании параметра сходства $\rho = 0,986$ и показателя обучения $\eta = 0,25$. Степень сжатия в этом случае равна 36,6, а PSNR – 28,3 dB.

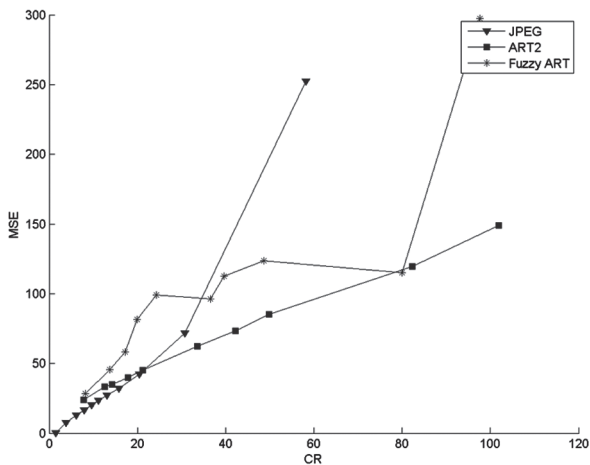


Рис. 2. Зависимости PSNR от степени сжатия



Рис. 3. Пример сжатия изображения

Выводы

В данной статье представлен новый алгоритм сжатия изображений на основе нейросетевого подхода. Основной его отличительной особенностью является то, что на этапе векторного квантования для создания соответствия между входным пространством и выходным пространством, состоящим из элементов кодовой книги - кодовых слов, или нейронов, была применена ИНС Fuzzy ART. Результаты моделирования данного алгоритма свидетельствуют о возможности применения данного вида ИНС для сжатия изображений.

Рассмотренный алгоритм сжатия более эффективен, чем алгоритм, основанный на использовании карт Кохонена и описанный в [6], и показал аналогичные результаты с алгоритмами, описанными в [5]. Дальнейшее направление исследований видится в применении различных модификаций

алгоритмов работы сети Fuzzy ART, таких как стратегия голосования, алгоритм упорядочивания, модифицированный алгоритм обучения [7-9] для дополнительного увеличения степени сжатия изображений.

Список литературы: 1. Schmidhuber J. Sequential Neural text Compression / J. Schmidhuber, S. Heil // IEEE Trans. on Neural Networks. – 1996. – V. 7. – N.1 – P.142-146. 2. Ahmed N. Discrete cosine transform / N. Ahmed, T. Natarajan and K. R. Rao // IEEE Trans. Comput. – 1974. – C-23 – P. 90-93. 3. Lei Sh.-M. An entropy coding system for digital HDTV applications / Sh.-M. Lei, M.-T. Sun // IEEE Trans. Circuits Syst. Video Technol. – 1991. – 1 – P. 147-155. 4. Carpenter G. A. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system / G. A. Carpenter, S. Grossberg, D. B. Rosen // Neural Networks. – 1991. – Vol. 4. – P. 759-771. 5. Руденко, О. Г. Сжатие изображений на основе нейронной сети ART [Текст] / О. Г. Руденко, М. С. Сныткін // Кибернетика и системный анализ. – 2008. – №6. – С.10-16. 6. Amerijckx Ch. Image Compression by Self-Organized Kohonen Map / Ch. Amerijckx, M. Verleysen, P. Thissen, J.-D. Legat // IEEE Trans. on Neural Net. – 1998. – 9 – P. 503-507. 7. Dagher I. An ordering algorithm for pattern presentation in Fuzzy ARTMAP that tends to improve generalization performance / I. Dagher, M. Georgiopoulos, G. L. Heileman, G. Bebis // IEEE Transactions on Neural Networks. – 1999. – Vol. 10, №4. – P. 768-778. 8. Meneganti M. Fuzzy neural networks for classification and detection of anomalies / M. Meneganti, F. S. Saviello, R. Tagliaferri // IEEE Transactions on Neural Networks. – 1998. – Vol. 9, №5. P. 848-861. 9. Rizzi A. A recursive algorithm for fuzzy min-max networks / A. Rizzi, M. Pannela, F. M. Fratalle Mascioli, G. Martinelli // Proceedings of International Joint Conference on Neural Networks (IJCNN'2000). – 2000. – Vol. 6. – P. 541-546.

Поступила в редколлегию 23.11.2011

УДК 519.71

Стиснення зображень за допомогою нейронної мережі Fuzzy ART / О. Г. Руденко, М. С. Сныткін // Біоніка інтелекту: наук.-техн. журнал. – 2012. – № 1 (78). – С. 45-48.

В статті розглядається алгоритм стиснення зображень на основі нейронної мережі Fuzzy ART, яка виконує операцію векторного квантування. Наведено опис алгоритму роботи і навчання нейронної мережі та алгоритму стиснення зображень. Також наводяться результати моделювання, які свідчать про ефективність застосування запропонованого алгоритму для стиснення зображень.

Табл. 1. Іл. 3. Бібліогр.: 9 найм.

UDC 519.71

Image compression by neural network Fuzzy ART / O. Rudenko, M. Snytkin // Bionics of Intelligense: Sci. Mag. – 2012. – № 1 (78). – P. 45-48.

In this paper an algorithm of image compression is considered. The algorithm is based on Fuzzy ART neural network. The network is used for vector quantization. The network's structure, its training algorithm and algorithm of image compression are described. The simulation results that show efficiency of the proposed algorithm of image compression are presented.

Tab. 1. Fig. 3. Ref.: 9 items.

УДК 519.854.2



Ф.Г. Ващук¹, О.О. Мельник²

¹ ЗакДУ, м. Ужгород, Україна, ZakDU@ukrport.net;

² ЗакДУ, м. Ужгород, Україна, me3000@ukr.net

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ АЛГОРИТМУ СКЛАДАННЯ РОЗКЛАДІВ СУМАРНОГО ВИПЕРЕДЖЕННЯ І ЗАПІЗНЕННЯ ІЗ НАЛАГОДЖЕННЯМИ, ЩО ЗАЛЕЖАТЬ ВІД ПОСЛІДОВНОСТІ

Розглядається задача складання розкладів за критерієм мінімізації сумарного випередження і запізнення відносно директивних строків при виконанні незалежних завдань одним приладом при наявності налагоджень (МВЗН). Проведені експериментальні дослідження запропонованого евристичного алгоритму пошуку локального оптимального розв'язку задачі показали, що він генерує за прийнятний час розв'язки, достатньо близькі до оптимальних, та може ефективно застосовуватись для розв'язання задач великої розмірності.

СКЛАДАННЯ РОЗКЛАДІВ, КРИТЕРІЙ МІНІМІЗАЦІЇ, НАЛАГОДЖЕННЯ, ФАКТОР ЗАПІЗНЕННЯ

Вступ

У [1, 2] описано новий підхід до розв'язання задачі за критерієм мінімізації сумарного випередження і запізнення (МВЗ). У статті [3] розширюється цей підхід для більш загальної та більш практичної задачі, що виникає особливо у виробничих задачах складання розкладів – задачі із часами налагодження, – та розробляється ефективний алгоритм A її розв'язання. Стаття присвячена дослідженню ефективності алгоритму A . Оцінювалась ефективність алгоритму у порівнянні з розробленим алгоритмом гілок і границь, вплив розмірності задачі на час її розв'язання, вплив діапазону директивних строків R і фактору запізнення T .

1. Постановка задачі

Множина з n незалежних завдань $J = \{j_1, j_2, \dots, j_n\}$ повинна бути призначена на виконання без переривань на одному приладі, який може працювати не більш, ніж з одним завданням одночасно. Прилад і завдання передбачаються безупинно доступними з моменту часу нуль, а простої приладу не допускаються. Завдання j , де $j = 1, 2, \dots, n$, вимагає часу виконання p_j і в ідеалі повинно бути закінчене у свій директивний строк d_j . Для окремих завдань задано час налагодження s_j , це означає, що в розкладі, у якому завдання виконується відразу після завдання i , повинен бути час налагодження s_j одиниць часу між моментом завершення завдання i , позначеним через C_i , і часом початку завдання j , що дорівнює $C_j - p_j$. Протягом цього періоду налагодження ніяке інше завдання не може бути виконано приладом. Часи налагодження є залежними від послідовності, тому що вони залежать як від i , так і від j . Для будь-якого заданого розкладу випередження та запізнення завдання j можуть бути визначені виразами (2) і (3). Мета полягає у тому, щоб знайти розклад, який мінімізує сумарне випередження і запізнення всіх завдань:

$$\sum_{j=1}^n (E_j + T_j), \quad (1)$$

де випередження і запізнення визначаються відповідно як:

$$E_j = \max(0, d_j - C_j) = (d_j - C_j)^+, \quad (2)$$

$$T_j = \max(0, C_j - d_j) = (C_j - d_j)^+. \quad (3)$$

2. Огляд існуючих результатів

Сформульована задача (позначимо її МВЗН) узагальнює задачу МВЗ [1, 2] і відноситься до класу NP -повних.

Складанню розкладів відносно директивних строків приділяється значна увага в літературі. Одна з причин цього явища – зростаючий тиск конкуренції на міжнародних ринках: фірми повинні запропонувати велику розмаїтість різних та індивідуальних виробів, у той час як клієнти очікують, що замовлені товари будуть поставлені вчасно. Принцип виробництва «точно в строк» установлює, що необхідна кількість товарів повинна бути вироблена або поставлена точно в заданий час. Виконання роботи з випередженням приводить до витрат на складування, у той час як запізнення робіт – до штрафів і, в остаточному підсумку, втраті доброзичливості клієнтів і репутації фірми.

У [4] розглядається актуальність задачі та приводиться огляд літератури зі складання розкладів «точно в строк» і розкладів з налагодженнями. Представлено алгоритм гілок і границь та евристична процедура одержання локального оптимуму. В [5] описується процедура змішаного цілочисельного програмування, яка дозволяє розв'язувати задачі невеликої розмірності.

На жаль, відомо замало публікацій про розв'язання цієї задачі, і точні методи на основі гілок і границь дозволяють розв'язувати задачі лише невеликої розмірності. Тому використовуються алгоритми локального пошуку як практичний

підхід для розв'язання задач комбінаторної оптимізації. Починаючи із припустимого розв'язку, ці алгоритми ітеративно пробують поліпшити поточний розв'язок, вишукуючи кращий розв'язок в околиці поточного розв'язку, поки не знайдений локальний оптимум. Ефективність цих алгоритмів якісно залежить від визначення околиці: при більших околицях якість розв'язку краще, але час обчислень більше. Тому на практиці більші околиці не корисні, якщо тільки вони не можуть ефективно досліджуватися.

3. Алгоритм розв'язання задачі

Евристичний алгоритм A розв'язання задачі МВЗН розроблено в [3].

Алгоритм складається з двох етапів.

На першому етапі налагодження не враховується. У блоці 1 розв'язується задача мінімізації сумарного запізнення при виконанні незалежних завдань одним приладом МСЗ [2]. Алгоритм побудовано на перестановках, він полягає в оптимальному використанні завдань, що запізнюються, та резервів завдань, що не запізнюються. Таким чином, реалізується зменшення сумарного запізнення за рахунок зменшення сумарного випередження. У блоці 2 здійснюється зменшення значення сумарного випередження і запізнення за допомогою послідовного збільшення моментів початку виконання завдань.

Позначимо $r_{\min} = \min\{r_j\}$, N_r – число завдань з резервами ($r_j > 0$); N_3 – число завдань, що запізнюються (до їх числа включаються завдання з нульовим резервом).

Твердження [6] (використання резервів завдань, що випереджають). Якщо у послідовності σ виконується $N_r > N_3$, то при збільшенні початку виконання завдань на величину, рівну r_{\min} , значення функціонала випередження/запізнення зменшується на величину $(N_r - N_3)r_{\min}$.

Розглядається послідовність, що отримана в результаті розв'язання задачі МСЗ. Якщо в отриманій послідовності резерви відсутні, то ця послідовність оптимальна за критерієм МВЗ. На кожній ітерації за умови $N_r \geq N_3$ збільшуються моменти початку виконання завдань у поточній послідовності на r_{\min} . Такі процедури виконуються, поки не буде виконано умову $N_r < N_3$. Отриману послідовність позначаємо σ^R .

На другому етапі здійснюється оптимізація послідовності σ^R з урахуванням налагоджень приладу, що залежать від послідовності, за допомогою процедури локального пошуку ефективного розв'язання поставленої задачі. Аналізується послідовність. Для тих завдань, для яких задані налагодження приладу, включаємо ці налагодження у тривалості виконання завдань. Отриману послідовність позначаємо σ^{R1} .

Використовуються такі типи перестановок: API (суміжна попарна перестановка), NAPI (несуміжна попарна перестановка), EBSR (витяг і повторна вставка зі зрушенням назад) та EFSR (витяг і повторна вставка зі зрушенням уперед).

Нижче приводяться результати досліджень ефективності алгоритму A .

4. Обчислювальні результати

Алгоритм A був закодований мовою C# у середовищі розробки Visual Studio 2010 під бібліотеку Microsoft .NET 4.0. Випробування проводилися на персональному комп'ютері із процесором Pentium CORE 2 Duo 2.0 ГГц із оперативною пам'яттю 2 Гбайта під управлінням ОС Microsoft Windows Vista. Досліджувалися задачі розмірності до 250 завдань.

Для визначення ефективності алгоритму були проведені дослідження залежності часу розв'язання задачі від розмірності та від складності задачі, а також дослідження проценту відхилення розв'язку від оптимального.

Схема генерації даних, запропонована Фішером [7], використовувалася для тестування алгоритму на різних типах прикладів, тип задачі визначається комбінацією фактора запізнення T і діапазону директивних строків R . Для кожної задачі спочатку генеруються тривалості виконання і часи налагоджень з рівномірного розподілу із заданими границями. Потім обчислюються директивні строки з розподілу, рівномірного на

$$\left[p^*(1-T-R/2), p^*(1-T+R/2) \right],$$

де p^* – сума всіх тривалостей. Значення T і R вибираються з множин $\{0,2; 0,4; 0,6; 0,8\}$ і $\{0,2; 0,4; 0,6; 0,8; 1,0\}$, відповідно, даючи по 20 задач кожного типу.

5. Дослідження впливу значень факторів R і T

Для визначення впливу діапазону директивних строків R і фактора запізнення T була промодельована робота алгоритму на задачах з параметрами R і T у діапазоні від 0 до 1 з кроком 0,2. Для кожної пари значень критеріїв і її розмірності було розв'язано не менш, ніж 20 задач. Це дозволило визначити такі параметри i , при яких алгоритм буде розв'язувати задачі відносно швидко, і які значення параметрів є найбільш складними для досліджуваного алгоритму.

Щоб продемонструвати загальну картину на невеликих задачах, у табл. 1–4 представлено середньоарифметичний час розв'язання задач різної розмірності. З ростом розмірності характеристики будуть такими ж. Тривалості виконання обрані в межах [50, 100]. Кількість завдань, що потребують налагодження, обрана на межі 50%.

Таблиця 1

Середній час розв'язання задач для $n = 10$ (мс)

$T \backslash R$	0	0,2	0,4	0,6	0,8	1
0	0,1	0,1	0,1	0,1	0,1	0,1
0,2	0,2	0,6	0,3	0,1	0,1	0,2
0,4	0,2	22,1	10,9	1,8	0,4	0,4
0,6	0,2	26,3	44,2	47,0	63,3	76,9
0,8	0,2	2,1	2,0	5,1	23,3	68,2
1	0,2	0,4	0,5	0,6	1,1	3,8

Таблиця 2

Середній час розв'язання задач для $n = 15$ (мс)

$T \backslash R$	0	0,2	0,4	0,6	0,8	1
0	0,2	0,2	0,2	0,2	0,2	0,2
0,2	0,3	0,9	0,4	0,2	0,2	0,3
0,4	0,3	33,2	16,4	2,8	0,6	0,5
0,6	0,3	39,4	66,5	70,8	95,5	116,2
0,8	0,3	3,2	3,0	7,6	35,0	102,5
1	0,3	0,6	0,8	0,9	1,7	5,8

Таблиця 3

Середній час розв'язання задач для $n = 20$ (мс)

$T \backslash R$	0	0,2	0,4	0,6	0,8	1
0	0,2	0,2	0,2	0,3	0,3	0,3
0,2	0,4	1,2	0,6	0,3	0,3	0,4
0,4	0,4	44,2	21,9	3,7	0,9	0,7
0,6	0,4	52,5	88,8	94,8	127,7	155,6
0,8	0,4	4,2	4,1	10,2	46,6	136,9
1	0,4	0,8	1,1	1,2	2,3	7,8

Таблиця 4

Середній час розв'язання задач для $n = 25$ (мс)

$T \backslash R$	0	0,2	0,4	0,6	0,8	1
0	0,3	0,3	0,3	0,4	0,4	0,4
0,2	0,5	1,4	0,7	0,4	0,4	0,4
0,4	0,5	55,3	27,4	4,6	1,1	0,9
0,6	0,5	65,7	111,2	118,8	160,1	195,3
0,8	0,5	5,3	5,1	12,7	58,3	171,4
1	0,5	1,0	1,4	1,4	2,8	9,8

З результатів добре видно, що множина розв'язків задач із різними значеннями критеріїв i має точку максимуму зі значеннями $R=1,0$; $T=0,6$. Також можна виділити області з локальним максимумом у точках зі значеннями параметрів $R=0,2$; $T=0,6$ і $R=0,8$; $T=0,6$. Це саме ті області, у яких розв'язання задач є самим важким для алгоритмів. Зі збільшенням розмірності задач різниця у швидкодії між задачами з параметрами, віднесеними до області максимуму, і всіма іншими задачами зростає. Це пояснюється тим, що у більшості областей алгоритм працює поліноміальний час, у той час як у вищезгаданих максимумах він зростає експоненційно, відповідно різниця збільшується за експоненційним законом.

Утворюємо на основі таблиць 1–4 три комбінації параметрів R і T , що відображають для кожної розмірності найбільш складні випадки задачі – комбінація I: $R=1,0$; $T=0,6$; комбінація II: $R=0,8$; $T=0,6$; комбінація III: $R=0,2$; $T=0,6$.

6. Результати порівняння із методом гілок і границь

Для того щоб показати, які результати дає алгоритм на задачах різної розмірності для комбінацій I, II, III параметрів T і R , зведемо результати з таблиць 1–4 в табл. 5. У колонці «Опт» показано час оптимального розв'язання задачі, отриманого алгоритмом гілок і границь.

Таблиця 5

Середній час розв'язання t_{cp} у порівнянні з точним алгоритмом (с)

Параметри задачі		t_{cp}		Параметри задачі		t_{cp}	
n	Комбінація параметрів	Опт	A	n	Комбінація параметрів	Опт	A
10	I	3,2	0,077	20	I	40,5	0,156
10	II	3,3	0,063	20	II	38,2	0,128
10	III	3,1	0,026	20	III	31,0	0,053
15	I	15,9	0,116	25	I	115,9	0,195
15	II	16,2	0,095	25	II	119,1	0,160
15	III	14,8	0,039	25	III	77,2	0,066

Результати з табл. 5 показують, що алгоритм A був дуже ефективним. У середньому на розв'язання треба було менш, ніж 0,2 секунди для будь-якого набору з 20 задач. Середня кількість часу, необхідного на оптимальне розв'язання алгоритмом гілок і границь, становила не більше 2 хвилин. Однак, час, необхідний для оптимального розв'язання, швидко збільшувався при зростанні розмірності. Це вказує на те, що використовувати алгоритм гілок і границь для розв'язання задач більших розмірностей непрактично.

7. Вплив розмірності задачі на її розв'язання

У табл. 6 показана залежність середнього часу розв'язання від розмірності для найскладнішої комбінації параметрів I ($R=1,0$; $T=0,6$).

Таблиця 6

Залежність часу розв'язання t_{cp} від розмірності при $R=1,0$; $T=0,6$ (с)

n	Кількість завдань, що потребують налагодження, %									
	10	20	30	40	50	60	70	80	90	100
50	0,37	0,37	0,38	0,38	0,38	0,39	0,39	0,39	0,41	0,41
75	1,19	1,21	1,22	1,23	1,24	1,26	1,28	1,29	1,31	1,32
100	2,10	2,12	2,13	2,16	2,18	2,19	2,22	2,24	2,26	2,29
125	2,70	2,73	2,74	2,75	2,77	2,80	2,81	2,83	2,85	2,88
150	3,11	3,13	3,17	3,20	3,24	3,27	3,32	3,35	3,40	3,45
175	3,78	3,83	3,89	3,94	4,01	4,08	4,15	4,23	4,30	4,38
200	3,92	4,05	4,19	4,34	4,49	4,65	4,83	5,00	5,20	5,38
225	5,15	5,25	5,38	5,51	5,65	5,79	5,94	6,10	6,26	6,44
250	5,52	5,70	5,89	6,09	6,31	6,52	6,76	7,01	7,26	7,53

8. Ефективність алгоритму

У табл. 7 наведено середнє відхилення значення функціонала від оптимального у відсотках. Для алгоритму A відсоток відхилення Δ обчислюється таким чином:

$$\Delta = (Z - Z_{opt}) \cdot 100 / Z_{opt},$$

де Z_{opt} – сумарне випередження/запізнення розв'язку, виконаного алгоритмом гілок і границь, а Z – сумарне випередження/запізнення розв'язку, виконаного евристичним алгоритмом.

Таблиця 7

Середнє відхилення значення функціонала від оптимального, %

n	Комбінація параметрів		
	I	II	III
10	0,0	0,0	0,0
15	4,1	5,0	2,5
20	8,9	7,4	12,3
25	21,3	20,4	18,1

Результати з табл. 7 показують, що алгоритм гілок і границь приводить до кращого значення функціонала, ніж алгоритм A . Фактично, для усіх задач з розмірністю 10 алгоритм A знайшов оптимальний розв'язок. Середній відсоток відхилення для алгоритму A становив біля 8,3% на всіх множинах задач і менше 9% для 8 з 12 множин задач.

Висновки

Ми представили результати досліджень ефективності алгоритму розв'язання задачі МВЗН із [3]. Дослідження показали, що алгоритм A дає розв'язки, близькі до оптимальних, та дозволяє за прийнятний час ефективно розв'язувати задачі великої розмірності (табл. 6 і 7).

Список літератури: 1. Павлов, О.А. Дослідження властивостей та розв'язання задачі «Мінімізація сумарного штрафу як за випередження, так і за запізнення відносно директивних строків при виконанні незалежних завдань одним приладом» [Текст] / О.А. Павлов, О.Б. Місюра, О.В. Мельников // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К. : «ВЕК+», 2008. – №48. – С. 3–6. 2. Згуровский, М.З. Принятие решений в сетевых системах с ограниченными ресурсами [Текст] / М.З. Згуровский, А.А. Павлов : Монография.

– К. : Наукова думка, 2010. – 573 с. 3. Ващук, Ф.Г. Складання розкладів сумарного випередження і запізнення із налагодженнями, що залежать від послідовності [Текст] / Ф.Г. Ващук, О.А. Павлов, О.Б. Місюра, О.О. Мельник // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К. : «ВЕК+», 2011. – №53. – С.192–194. 4. Sourd F. Earliness–tardiness scheduling with setup considerations / F. Sourd // Computers and Operations Research. – 2005. – Vol. 32. – P. 1849–1865. 5. Coleman B.J. A simple model for optimizing the single machine early/tardy problem with sequence dependent setups / B.J. Coleman // Production and Operations Management. – 1992. – Vol. 1. – P. 225–228. 6. Згуровский, М.З. ПДС-алгоритмы и труднорешаемые задачи комбинаторной оптимизации [Текст] / М.З. Згуровский, А.А. Павлов, Е.Б. Мисюра // Системні дослідження та інформаційні технології. – 2009. – №4. – С. 14–31. 7. Fisher M.L. A dual algorithm for the one-machine scheduling problem / Fisher M.L. // Math. Programming. – 1976. – №11. – P. 229–251.

Надійшла до редколегії 7.12.2011

УДК 519.854.2

Исследование эффективности алгоритма составления расписаний суммарного опережения и запаздывания с наладками, зависящими от последовательности / Ф.Г. Ващук, Е.А. Мельник // Бионика интеллекта: науч.-техн. журнал. – 2012. – № 1 (78). – С. 49–52.

В статье рассмотрены результаты исследования эффективности алгоритма поиска локального оптимального решения задачи составления расписаний по критерию минимизации суммарного опережения и запаздывания. Приведены результаты сравнения указанного алгоритма с методом ветвей и границ для задач большой размерности. Исследования показали, что рассмотренный эвристический алгоритм дает решения, близкие к оптимальным.

Табл. 7. Библиогр.: 7 назв.

UDK 519.854.2

Research of efficiency of algorithm machine scheduling of minimizing the total earliness and tardiness that depend on sequence / Vashchuk F.G., Melnyk O.O. // Bionics of Intelligense: Sci. Mag. – 2012. – № 1 (78). – P. 49–52.

The results of the research of efficiency of algorithm to search a local optimal solution of the machine scheduling of minimizing the total earliness and tardiness problem are under consideration in this article. The results of comparison of the proposed algorithm with the branch and bound algorithm for large dimension problems are given. The researches have shown that the proposed heuristic algorithm gives the solutions close to optimum.

Tab. 7. Ref.: 7 items.

УДК 681.518:004.93'1



А.С. Довбиш¹, А.М. Романюк², М.С. Руденко³

¹ Сумський державний університет, м. Суми, Україна, kras@id.sumdu.edu.ua

² Сумський державний університет, м. Суми, Україна, info@pathomorph.sumdu.edu.ua

³ Сумський державний університет, м. Суми, Україна, makrudenko@yandex.ru

ІДЕНТИФІКАЦІЯ КАДРІВ В ЗАДАЧАХ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ МЕДИЧНИХ І БІОЛОГІЧНИХ ОБ'ЄКТІВ

Розглянута задача розпізнавання у рамках інформаційно-екстремальної інтелектуальної технології аналізу і синтезу систем розпізнавання, що навчаються, морфологічних зображень тканин пацієнтів при діагностуванні онкологічних захворювань шляхом порівняння кадрів зображень з еталонами. Досліджено вплив системи контрольних допусків для ознак розпізнавання на функціональну ефективність навчання комп'ютеризованої системи діагностування.

ІДЕНТИФІКАЦІЯ КАДРІВ, ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ, ЕТАЛОННЕ ЗОБРАЖЕННЯ, ОПТИМІЗАЦІЯ, НАВЧАННЯ, ІНФОРМАЦІЙНИЙ КРИТЕРІЙ

Вступ

Розпізнавання морфологічних зображень тканин, отриманих за методом біопсії, при діагностуванні онкопатологій на практиці здійснюється за довільних початкових умов формування зображень, що обумовлює апріорну невизначеність. Тому основним шляхом підвищення функціональної ефективності комп'ютеризованої системи діагностування (КСД), яка виконує функції системи підтримки прийняття рішень (СППР) для лікаря-онколога, є надання їй властивості адаптивності на основі машинного навчання та розпізнавання образів [1, 2]. Більшість відомих методів розпізнавання носять модельний характер, оскільки вони не враховують перетин класів розпізнавання, що має місце в практичних задачах діагностування [3, 4]. У цих випадках виникає необхідність попередньої нормалізації образів з метою їх наближення до еталонних. Одним із перспективних напрямів аналізу і синтезу КСД, що функціонують за умов апріорної невизначеності, є використання ідей і методів інформаційно-екстремальної інтелектуальної технології (ІЕІ-технологія), яка ґрунтується на максимізації одержаної в процесі навчання системи розпізнавання кількості інформації шляхом введення додаткових інформаційних обмежень [5, 6]. У праці [7] запропоновано інформаційно-екстремальний алгоритм навчання КСД онкологічних захворювань молочної залози, але авторам не вдалося побудувати безпомилкові за навчальною матрицею вирішальні правила через суттєвий вплив фону зображень на достовірність розпізнавання. Це пояснюється тим, що на рецепторному полі зображень існують однакові для всіх захворювань фонові ділянки, які вносять додаткові завади в навчальну матрицю.

З метою підвищення функціональної ефективності КСД у статті задача розпізнавання морфологічних зображень тканин пацієнтів з онкологічними захворюваннями розглядається як

задача ідентифікації кадрів загального зображення, розв'язок якої дозволяє безпосередньо виявляти локалізації конгломератів онкологічних клітин.

1. Постановка задачі

Нехай задано алфавіт класів розпізнавання $\{X_m^o | m = 1, M\}$, елементи якого характеризують різні патології молочної залози. Існує розбиття рецепторного поля зображення на K кадрів певного розміру, і для кожного k -го кадру сформовано навчальну матрицю яскравості $\|y_{k,i}^{(j)} | i = \overline{1, N}; j = \overline{1, n}\|$, де N, n – кількість ознак розпізнавання і векторів-реалізацій образу відповідно. Для кожного класу X_m^o , які приймаються за еталони, сформовано навчальні матриці яскравості аналогічної розмірності.

Необхідно в процесі ідентифікації k -го кадру рецепторного поля, який характеризується класом X_k^o , побудувати вирішальні правила шляхом відновлення у радіальному базисі простору ознак контейнерів для кожної пари «кадр – еталон» і обчислити критерій схожості.

$$C_{k,m} = 1 - E_{k,m}^* \quad (1)$$

де $E_{k,m}^*$ – глобальний максимум нормованого інформаційного критерію функціональної ефективності (КФЕ) системи діагностування, що обчислюється в робочій допустимій області визначення його функції.

Необхідно за максимальним значенням критерію схожості (1) прийняти рішення про належність ідентифікованого кадру відповідному класу із заданого алфавіту $\{X_m^o\}$.

2. Математична модель

Вхідний математичний опис системи компараторного розпізнавання кадрів подамо у вигляді структури

$$\Delta = \langle G \times T \times \Omega \times Z \times K, Y^{[2]}, X^{[2]}; \Phi_1, \Phi_2 \rangle,$$

де G – множина вхідних сигналів (факторів), які впливають на КСД; T – множина моментів часу

зняття інформації; Ω – простір ознак розпізнавання; Z – простір можливих функціональних станів КСД; K – множина кадрів для ідентифікації; $Y^{[2]}$ – навчальна матриця для двох класів; $X^{[2]}$ – бінарна навчальна матриця; $\Phi_1 : G \times T \times \Omega \times Z \rightarrow Y^{[2]}$ – оператор формування вхідної навчальної матриці; $\Phi_2 : Y^{[2]} \rightarrow X^{[2]}$ – оператор формування бінарної навчальної матриці $X^{[2]}$.

Розглянемо категорійну модель розпізнавання образів у рамках ІЕІ-технології, показану на рис. 1. Оператор $\theta : X^{[2]} \rightarrow \mathfrak{R}^{[2]}$ будує розбиття простору ознак на два класи розпізнавання, яке є апріорно нечітким. Оператор класифікації ψ перевіряє основну статистичну гіпотезу про належність реалізацій $\{x_m^{(j)} \mid j = \overline{1, n}\}$ класу X_m^o . Оператор $\gamma : I^{[l]} \rightarrow \mathfrak{S}^{[q]}$ шляхом оцінки статистичних гіпотез формує множину точнісних характеристик, де l – кількість статистичних гіпотез; $q = l^2$ – кількість точнісних характеристик. Оператор ϕ формує терм-множину E , яка складається із значень інформаційного критерію функціональної ефективності (КФЕ). Контур оптимізації геометричних параметрів розбиття $\mathfrak{R}^{[2]}$ замикається оператором d , а контур оптимізації системи контрольних допусків (СКД), яка утворює множину D , замикається послідовно оператором $\delta_1 : E \rightarrow D$ і оператором $\delta_2 : D \rightarrow Y$, який змінює реалізацію образу в процесі навчання. Оператор $U : E \rightarrow G \times T \times \Omega \times Z \times K$ – регламентує процес навчання КСД.

Подання слабо формалізованого процесу навчання КСД у вигляді категорійної моделі суттєво спрощує розроблення структурної схеми алгоритму навчання.

3. Алгоритм ідентифікації кадрів

Згідно з постановкою задачі алгоритм ідентифікації кадрів полягає в обчисленні для кожного поточного кадру критерію схожості (1) з відповідними еталонами із алфавіту $\{X_m^o\}$. Обчислення максимального значення інформаційного КФЕ $E_{k,m}^*$ здійснюється в процесі інформаційно-екстремального навчання КСД за ітераційною двоциклічною процедурою

$$E_{k,m}^* = \max_{G_\delta} \max_{G_E \cap G_d} E_{k,m}, \quad (2)$$

де $E_{k,m}$ – значення КФЕ навчання системи розпізнавати реалізації поточного k -го кадру і m -го еталону, що обчислюється в робочій допустимій області визначення його функції; G_δ, G_E, G_d – області допустимих значень параметра δ поля контрольних допусків, значень КФЕ і радіусів гіперсферичних контейнерів класів компараторного розпізнавання, що відновлюються в радіальному базисі простору ознак Ω відповідно.

Таким чином зовнішній цикл оптимізує параметр δ поля контрольних допусків, за допомогою якого обчислюються нижні $A_{H,k}$ і верхні $A_{B,k}$ контрольні допуски

$$A_{H,k} = y_k - \delta; \quad A_{B,k} = y_k + \delta, \quad (3)$$

де y_k – еталонний (усереднений) вектор-реалізація навчальної матриці $Y^{[2]}$ для k -го кадру.

З метою підвищення оперативності алгоритму навчання розглянемо паралельну оптимізацію СКД на ознаки розпізнавання, за якою контрольні допуски на кожному кроці навчання змінюються за формулою (3) для всіх ознак одночасно.

Структурну схему алгоритму паралельної оптимізації СКД на ознаки розпізнавання показано на рис. 2. Вхідними даними є масив навчальних вибірок $Y[kl][realization][j]$, де kl – номер навчальної матриці класу розпізнавання, $realization$ – номер реалізації в навчальній матриці і j – номер ознаки розпізнавання в реалізації образу. Розглянемо коротко призначення функціональних блоків у структурній схемі алгоритму навчання КСД з паралельною оптимізацією СКД на ознаки розпізнавання:

Блок 3. Занулюється значення тимчасової змінної e_{optim_sred} , яка використовується для прямого пошуку глобального максимуму усередненого значення КФЕ ідентифікації кадрів.

Блок 4. Ініціалізується лічильник циклу параметра δ ($delta$) поля контрольних допусків.

Блок 5. Формуються на кожному кроці навчання за формулою (3) масиви поточних верхніх та нижніх контрольних допусків на ознаки розпізнавання.

Блок 6. Виконується базовий інформаційно-екстремальний алгоритм навчання [5], який реалізує внутрішній цикл процедури (2).

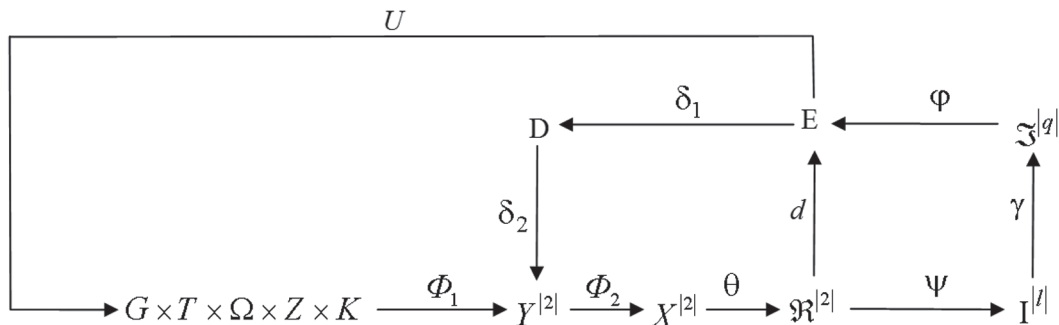


Рис. 1. Категорійна модель навчання системи

Функціями базового інформаційно-екстремального алгоритму навчання системи розпізнавання для ідентифікації кадрів є:

- обчислення на кожному кроці ідентифікації кадрів інформаційного КФЕ;
- пошук глобального максимуму КФЕ в робочій (допустимій) області визначення його функції.

Оскільки критерій $E_{k,m}^*$ є функціоналом від точнісних характеристик ідентифікації кадрів, то за робочу приймається область визначення його функції, в якій ймовірності правильних рішень перебільшують відповідні ймовірності помилкових рішень, що складають однакові групи подій.

Як КФЕ навчання при компаративному розпізнаванні можна використовувати будь-яку статистичну інформаційну характеристику, що є мірою різноманітності двох класів розпізнавання. Наприклад, при двох альтернативних рішеннях і рівноймовірних гіпотезах модифікація нормованого ентропійного критерію (за Шенноном) [6] при компаративному розпізнаванні буде мати вигляд:

$$E_{k,m} = 1 + \frac{1}{2} \left(\frac{\alpha_{k,m}(d_k)}{\alpha_{k,m}(d_k) + D_{2k,m}(d_k)} \log_2 \frac{\alpha_{k,m}(d_k)}{\alpha_{k,m}(d_k) + D_{2k,m}(d_k)} + \frac{D_{1k,m}(d_k)}{D_{1k,m}(d_k) + \beta_{k,m}(d_k)} \log_2 \frac{D_{1k,m}(d_k)}{D_{1k,m}(d_k) + \beta_{k,m}(d_k)} + \frac{\beta_{k,m}(d_k)}{D_{1k,m} + \beta_{k,m}(d_k)} \log_2 \frac{\beta_{k,m}(d_k)}{D_{1k,m} + \beta_{k,m}(d_k)} + \frac{D_{2k,m}(d_k)}{\alpha_{k,m}(d_k) + D_{2k,m}(d_k)} \log_2 \frac{D_{2k,m}(d_k)}{\alpha_{k,m}(d_k) + D_{2k,m}(d_k)} \right), (4)$$

де $\alpha_{k,m}(d_k)$, $\beta_{k,m}(d_k)$, $D_{1k,m}(d_k)$, $D_{2k,m}(d_k)$ – значення точнісних характеристик, отриманих в процесі ідентифікації k -го кадру з m -м еталоном: помилки першого і другого родів, перша і друга достовірності відповідно; d_k – радіус гіперсферичного класифікатора, що відновлюється на кожному кроці ідентифікації в радіальному просторі ознак розпізнавання.

Блоки 7, 8. В масиві E_{optim} виконується безпосередній пошук оптимального параметра поля контрольних допусків ($delta$) для ознак розпізнавання шляхом пошуку максимального усередненого значення КФЕ, отриманого в блоках 5,6, який знаходиться в робочій області для двох класів розпізнавання.

Після проведення навчання обчислюється критерій схожості (1) між кадром та еталоном зображення і приймається рішення про належність ідентифікованого кадру m -му еталону. Чим вищий критерій схожості, тим вище достовірність правильного діагностування захворювання, оскільки значення інформаційного КФЕ, який є мірою

різноманітності, стає досить малим, що свідчить про відсутність суттєвих відмінностей між кадром, що ідентифікується, і еталоном зображенням.

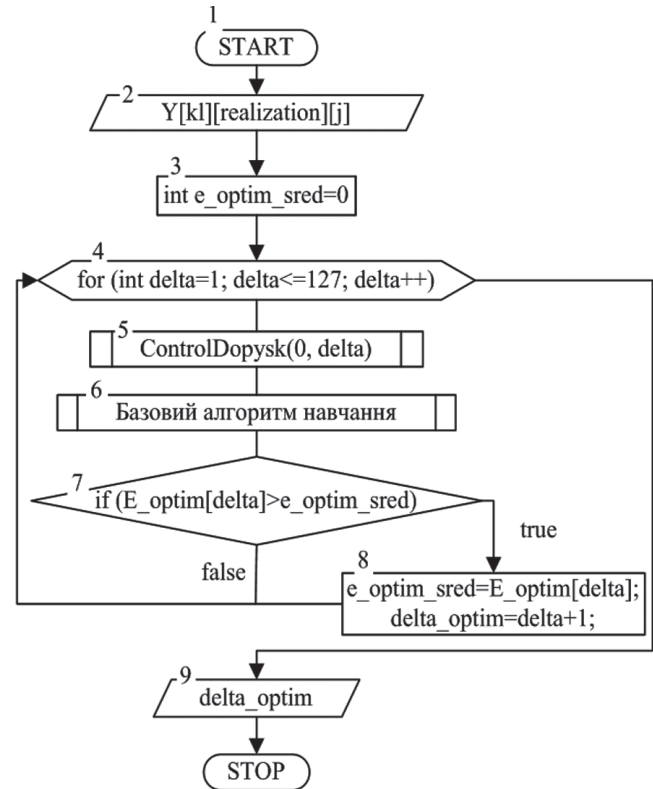


Рис. 2. Структурна схема алгоритму ідентифікації кадрів з оптимізацією СКД

4. Результати фізичного моделювання

Реалізацію вищенаведеного алгоритму ідентифікації кадрів розглянемо на прикладі розпізнавання морфологічних зображень тканин пацієнтів, одержаних за методом біопсії. Зображення формуються за допомогою вбудованого в мікроскоп цифрового фотоапарата, з'єданого з ЕОМ, і відображається на екрані монітора.

Як етальонні розглядалися зображення трьох онкологічних захворювань молочної залози (мастопатія, рак і фіброаденома) розміром 200×200 пікселів. При цьому загальне зображення тканини пацієнта розміром 1280×1024 , яка мала конгломерати онкологічних клітин, розбивалося на кадри, що ідентифікувалися, розмір яких дорівнював еталоном зображенням.

Ідентифікація кадрів здійснювалася за вищенаведеним алгоритмом, концепцію якого схематично показано на рис. 3. Тут прийнято такі позначення: М – мастопатія; Р – рак молочної залози; Ф – фіброаденома і К – кадр, що ідентифікується.

На рис. 4 наведено приклади трьох етальонних зображень онкологічних захворювань відібраних експертом.

Результати реалізації базового алгоритму ідентифікації 11-го кадру загального зображення показано на рис. 5, де наведено графіки залежності

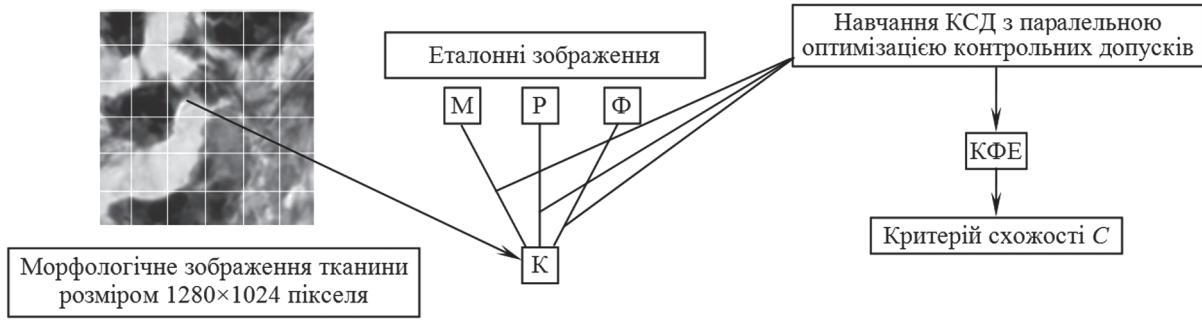
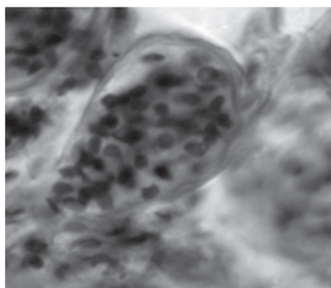
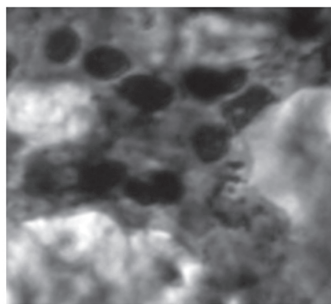


Рис. 3. Процес ідентифікації кадрів

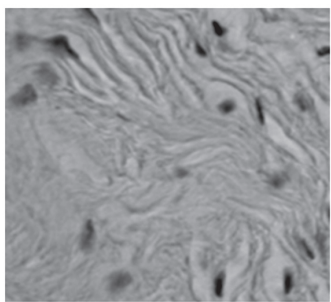
критеріїв оптимізації від радіусу d_{11} контейнера зображення кадру, який послідовно відновлюється в процесі ідентифікації з трьома еталонами (рис. 4).



а



б

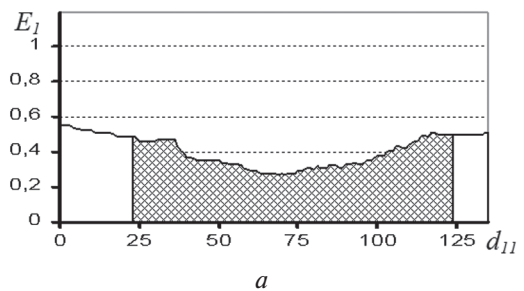


в

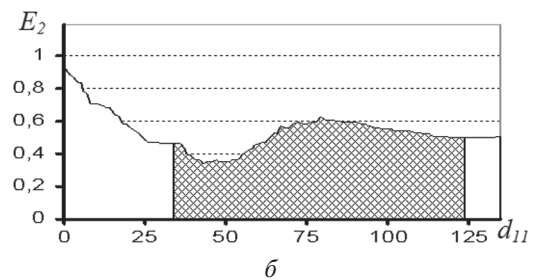
Рис. 4. Еталонні зображення трьох онкологічних захворювань розміром 200×200 11-го кадру (а – мастопатія, б – рак, в – фіброаденома)

На рис. 5 заштрихована ділянка визначає роботу (допустиму) область визначення КФЕ навчання системи ідентифікації кадрів, в якій перша та друга достовірності більше 0,5, тобто помилки першого і другого роду будуть менше відповідних достовірностей. Аналіз графіків показують, що КФЕ не досягає свого максимального граничного значення

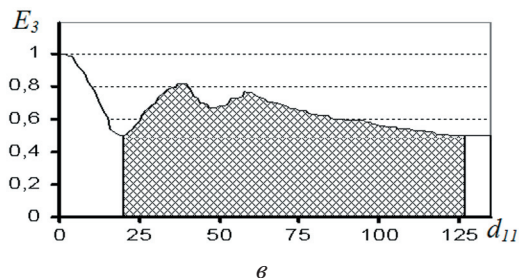
($E_{\text{гран}}^* = 1$) в процесі ідентифікації кадру з заданими еталонами. При цьому максимальне значення КФЕ для класу «мастопатія» дорівнює $E_1^* = 0,51$, а оптимальний радіус – $d_{11}^* = 119$ (тут і далі у кодових одиницях); для класу «рак» відповідно – $E_2^* = 0,62$ і $d_{11}^* = 80$; для класу «фіброаденома» – $E_3^* = 0,81$ і $d_{11}^* = 38$. При цьому критерії схожості відповідно дорівнюють $C_1 = 0,49$, $C_2 = 0,38$ і $C_3 = 0,19$. Тобто одержані результати не дозволяють з високою достовірністю визначити належність поточного кадру до одного із захворювань. Тому доцільним є проведення навчання КСД з паралельною оптимізацією СКД. На рис. 6 показано результати оптимізації СКД в процесі ідентифікації 11-го кадру загально-го зображення.



а



б



в

Рис. 5. Графіки залежності КФЕ від радіусів контейнера зображення кадру, що ідентифікується (а – мастопатія, б – рак, в – фіброаденома)

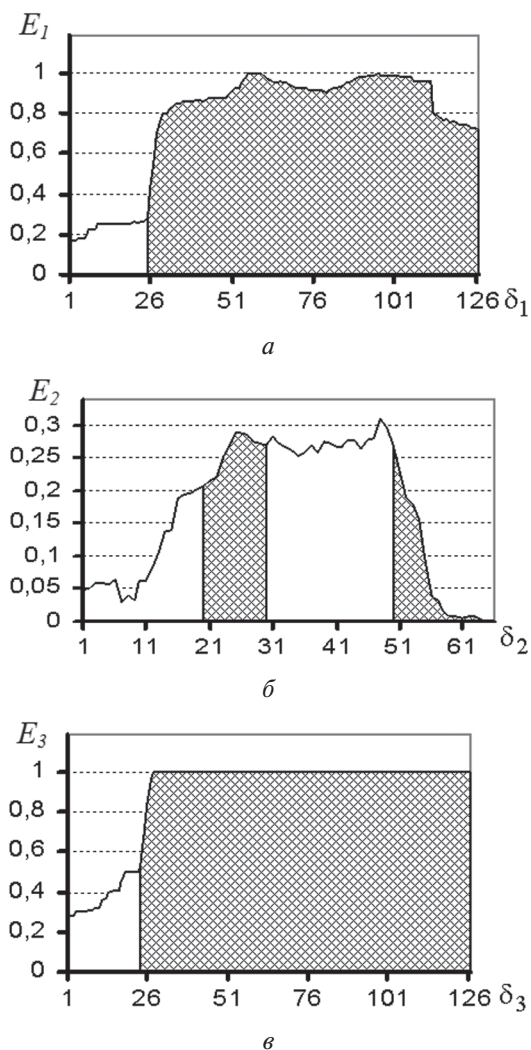


Рис. 6. Графіки залежності КФЕ від параметру поля контрольних допусків на ознаки розпізнавання (а – мастопатія, б – рак, в – фіброаденома)

Аналіз рис. 6 показує, що для мастопатії (рис. 6а) і для фіброаденоми (рис. 6в) значення КФЕ навчання КСД дорівнюють максимальному граничному ($E_1^* = E_3^* = 1$). Оскільки відповідні критерії схожості згідно з формулою (1) дорівнюють нулю, то можна зробити висновок, що гіпотези про належність кадру, що ідентифікується, до класів «мастопатія» та «фіброаденома» відкидаються. При цьому аналіз рис. 6б показує, що максимальне значення КФЕ дорівнює $E_2^* = 0,31$ при відповідному значенні критерію схожості $C_{11,2} = 0,79$. Таким чином, має місце висока достовірність ідентифікації кадру з класом «рак молочної залози».

В табл. 1 наведено результати ідентифікації всіх 30 кадрів загального зображення тканини пацієнта.

Аналіз табл. 1 показує, що після проведення процесу ідентифікації кадрів морфологічного зображення тканини пацієнта було виявлено 5 кадрів з критерієм схожості на рак більше 0,9 та 12 кадрів з критерієм схожості на рак більше 0,8. При цьому середнє значення критерію схожості на рак для всіх кадрів зображення дорівнює 0,75, що дозволяє

зробити з високою достовірністю діагностичний висновок про наявність раку молочної залози.

Таблиця 1

Результати процесу ідентифікації кадрів

№ кадру	Критерій схожості на мастопатію, C	Критерій схожості на рак, C	Критерій схожості на фіброаденому, C
1	0,36	0,75	0
2	0	0,78	0
3	0	0,82	0,12
4	0,1	0,69	0
5	0,28	0,71	0
6	0	0,89	0
7	0	0,85	0
8	0	0,83	0
9	0	0,92	0
10	0,08	0,63	0,37
11	0	0,95	0
12	0	0,83	0
13	0,39	0,49	0
14	0	0,91	0
15	0	0,32	0,15
16	0	0,85	0
17	0	0,88	0
18	0	0,73	0
19	0,23	0,5	0
20	0	0,63	0
21	0	0,89	0
22	0	0,86	0,34
23	0	0,93	0
24	0,4	0,69	0
25	0	0,25	0,06
26	0	0,89	0
27	0	0,84	0
28	0	0,91	0
29	0	0,81	0
30	0,02	0,54	0

Особливість системи компараторного розпізнавання, що здійснює ідентифікацію кадрів, полягає у відсутності етапу екзамену. Тому задачею навчання такої системи є не побудова вирішальних правил, як це має місце для традиційних задач розпізнавання образів, а обчислення в процесі ідентифікації кадрів максимального значення нормованого інформаційного критерію оптимізації параметрів навчання. З іншого боку, досягнення глобального максимуму критерію (4) в робочій області визначення його функції можливо за оптимальних в інформаційному розумінні параметрів навчання. Наприклад, аналіз рис. 5 показує, що оптимальними параметрами контрольних полів допусків на ознаки розпізнавання є для класу «мастопатія» $\delta_1^* = 56$ (тут і далі в градаціях яскравості пікселів рецепторного поля зображень), для класу «рак молочної залози» $\delta_2^* = 25$ і для класу «фіброаденома» $\delta_3^* = 29$.

Оптимальні параметри контейнерів, що відновлюються в процесі ідентифікації 11-го кадру, визначаються за показаними на рис. 7 графіками залежності КФЕ від радіуса d_{11} контейнерів зображення цього кадру в процесі паралельної оптимізації СКД.

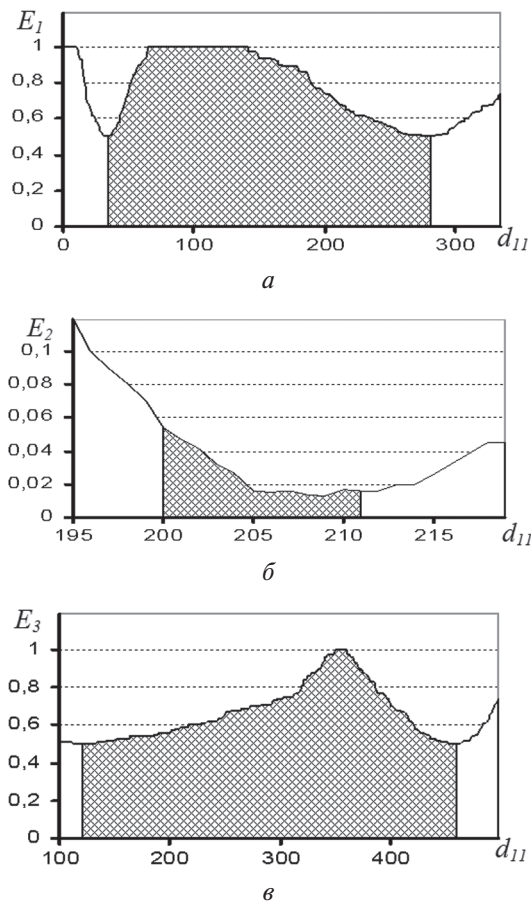


Рис. 7. Графіки залежності КФЕ від радіуса контейнерів для еталонних зображень (а – мастопатія, б – рак, в – фіброаденома)

Аналіз рис. 7б показує, що максимальне значення інформаційного КФЕ ідентифікації кадру щодо еталонного зображення дорівнює $E_2^* = 0,05$, що свідчить про погану роздільність поточного кадру з еталонним. При цьому критерій схожості (1) дорівнює $C = 0,95$. При порівнянні кадру, що ідентифікується, з кадрами-еталонами класів «мастопатія» і «фіброаденома» (рис. 7а і рис. 7в відповідно) КФЕ навпаки досягає максимального значення, що свідчить про відмінність поточного кадру від еталонних зображень цих класів. Таким чином, можна зробити остаточний висновок, що текстура даного кадру належить до раку молочної залози.

Висновки

1. Запропоновано новий інформаційно-екстремальний метод ідентифікації кадрів, в основу якого покладено принцип максимізації кількості інформації між поточним кадром, що ідентифікується, і кадром-еталоном класу відповідної онкопатології.
2. З метою підвищення достовірності ідентифікації при збільшенні потужності алфавіту класів

розпізнавання перспективним розвитком запропонованого методу є побудова ієрархічної структури інформаційно-екстремального алгоритму ідентифікації, що дозволить вирішити в певних межах проблему багатовимірності.

Список літератури: 1. Ивахненко, А.Г. О принципах построения обучающихся систем управления сложными процессами [Текст] / А.Г. Ивахненко. – М.: Наука, 1970. – 252 с. 2. Цыпкин, Я.З. Основы информационной теории идентификации [Текст] / Я.З. Цыпкин. – М.: Наука, 1984. – 320 с. 3. Васильев, В.И. Распознающие системы. Справочник. 2-е изд., перераб. и доп. [Текст] / В.И. Васильев. – Киев: Наукова думка, 1983. – 422 с. 4. Путятин, Е.П. Обработка изображений в робототехнике [Текст] / Е.П. Путятин, С.И. Аверин. – М: Машиностроение, 1990. – 320 с. 5. Краснополюсовский, А.С. Информационный синтез интеллектуальных систем керування: Підхід, що ґрунтується на методі функціонально-статистичних випробувань [Текст] / А.С. Краснополюсовский – Суми: Видавництво СумДУ, 2004. – 261 с. 6. Довбиш, А.С. Основы проектирования интеллектуальных систем: Навчальний посібник [Текст] / А.С. Довбиш. – Суми: Видавництво СумДУ, 2009. – 171 с. 7. Довбиш, А.С. Оптимізація параметрів навчання СППР для діагностування онкопатологій [Текст] / А.С. Довбиш., М.С. Руденко // Вісник СумДУ. Серія: Технічні науки. – 2010. – №2. – С. 48-55.

Надійшла до редколегії 29.12.2011

УДК 681.518:004.93'1

Идентификация кадров в задачах распознавания изображений медицинских и биологических объектов / А.С. Довбиш, А.М. Романюк, М.С. Руденко // Бионика интеллекта: науч.-техн. журнал. – 2012 – № 1 (78). – С. 53-58.

Предложен информационно-экстремальный метод компараторного распознавания, который заключается в идентификации кадров морфологических изображений тканей пациентов с целью диагностирования онкопатологий. В качестве критерия схожести кадра, который идентифицируется, рассматривается модифицированный энтропийный критерий Шеннона. Максимизация количества информации совершается путем оптимизации параметров функционирования в рамках информационно-экстремальной интеллектуальной технологии.

Табл. 1. Ил. 7. Библиогр.: 7 назв.

UDC 681.518:004.93'1

Identification frames in tasks of recognition image of oncological and medical objects / A.S. Dovbish, A.M. Romanyuk, M.S. Rudenko // Bionics of Intelligence: Sci. Mag. – 2012. – № 1 (78). – P. 53-58.

Proposed information-extremely method of comparative recognition, which consists of identification frames of the morphological image tissue of patients with the aim of diagnosing cancer pathology. As a criterion for the similarity of the frame, which is identified, considered the modified entropy Shannon's criterion. Maximizing the number of information is accomplished by optimizing the parameters of the operation within of the information-extremely intellectual technologies.

Tab. 1. Fig.: 7. Ref.: 7 items.

УДК 004.652.4, 004.652.6

Ю.А. Мальков¹, В.А. Радченко², С.С. Таянский³¹ ХНУРЭ, г. Харьков, Украина, malkov@smtp.ru² ХНУРЭ, г. Харьков, Украина, sayrad@gmail.com³ ХНУРЭ, г. Харьков, Украина, tanyansky_ss@yahoo.com

ОРГАНИЗАЦИЯ ХРАНЕНИЯ ДИЗЬЮНКТОВ ДЕДУКТИВНЫХ БАЗ ДАННЫХ НА ОСНОВЕ РЕЛЯЦИОННОГО ПОДХОДА

Рассматриваются основные синтаксические конструкции дедуктивных баз данных. На основе анализа структуры этих конструкций представлен метод их декомпозиции в реляционные отношения с целью хранения и эффективного использования.

БАЗЫ ДАННЫХ, ДЕДУКТИВНЫЕ БАЗЫ ДАННЫХ, DATALOG, ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ, ПРАВИЛА ВЫВОДА

Введение

Дедуктивные базы данных (ДБД) являются концептуальным расширением классического понятия базы данных. В общем случае ДБД является системой логического программирования, разработанной для оперирования большими объемами данных. Они расширяют понятия базы данных, используя выразительную мощь логического программирования, а их применение значительно упрощает разработку приложений, обеспечивая расширенные возможности обработки знаний на уровне базы данных. При таком подходе базы данных становятся не только источниками данных, но и источниками знаний. Таким образом, дедуктивные базы данных представляют собой сочетание обычной базы данных, содержащей факты, базы знаний, содержащей правила, описывающие свойства данных, и систему логического вывода, позволяющую продуцировать информацию на основе фактов и правил.

Зачастую при реализации СУДБД [1, 2] для хранения множества фактов (экстенциональной информации) используется реляционная СУБД. Причиной выбора реляционной системы является простое, интуитивно понятное представление данных в виде таблиц (в терминах модели — реляционных отношений), а также лежащего в основе математического аппарата для выполнения операций над отношениями — реляционной алгебры и реляционного исчисления.

Вопрос хранения правил, образующих интенционал дедуктивной базы данных, в литературе ранее рассмотрен не был. В связи с этим, целью данной работы является разработка логической структуры реляционной системы баз данных для хранения правил ДБД.

1. Компоненты дедуктивной базы данных

В ДБД все синтаксические конструкции записываются в виде выражений логического программирования, называемых дизьюнктами [3], а язык, на котором они определяются, называется Дейталог.

В общем виде дизьюнкт записывается в виде формулы:

$$\forall X_1, \dots, \forall X_n (A \vee \neg B_1 \vee \dots \vee \neg B_m), \quad (1)$$

где $m \geq 0$, A, B_1, \dots, B_m — предикатные символы, X_1, \dots, X_n — переменные, появляющиеся в предикатных символах A, B_1, \dots, B_m . Поскольку такая форма представления достаточно сложная для восприятия, дизьюнкты часто записывают в пролог нотации, упуская при этом переменные

$$A \leftarrow B_1, \dots, B_m. \quad (2)$$

Все переменные, входящие в дизьюнкт, связаны квантором всеобщности, а запятой между атомами A, B_1, \dots, B_m обозначен символ конъюнкции (\vee). Литерал A называют головой правила, литералы B_1, \dots, B_m образуют тело правила, а символом \leftarrow обозначена операция импликации. Приведенная запись интерпретируется следующим образом: «Если истинно B_1, \dots, B_m , то истинно A ».

Литералы, образующие дизьюнкты, состоят из n -арных предикатных символов и списка аргументов (t_1, \dots, t_n) , где t_i — терм, являющийся константой или переменной.

Существует три вида дизьюнктов: факты, правила вывода и ограничения целостности.

Дизьюнктивные выражения вида (3), не содержащие положительных литералов, называют фактами.

$$A \leftarrow \quad (3)$$

Знак импликации в таком случае опускают и записывают без него, например A .

Дизьюнктивные выражения вида (4), не содержащие положительных литералов, называют ограничениями целостности. Если при подстановке фактов дизьюнкт такого вида вернет истинное значение, целостность базы данных будет нарушена.

$$\leftarrow B_1, \dots, B_m. \quad (4)$$

Когда дизьюнкт содержит один положительный и один или несколько отрицательных литералов, его называют правилом вывода или дедуктивной

аксиомой вида (5). Правила вывода дают определение литерала в голове правила в терминах литералов, представленных в теле правила:

$$A \leftarrow B_1, \dots, B_m. \quad (5)$$

Дизъюнкты являются замкнутыми формулами, т.е. не содержат свободных переменных. Из этого следует, что имя переменной, появляющейся в дизъюнкте C , имеет смысл только в этом дизъюнкте и ее значение не распространяется на что-либо вне дизъюнкта.

Дедуктивную базу данных вида (6) в общем случае разделяют на EDB – экстенциональную базу данных (экстенционал), P – множество правил вывода и ограничений целостности или логическую программу:

$$M_D = \langle EDB, P \rangle, \quad (6)$$

Экстенциональная база данных представляет собой данные, которые в действительности хранятся в отношениях реляционной базы данных. В связи с этим EDB также будем называть экземпляром реляционной базы данных.

В дедуктивной системе каждый кортеж реляционного отношения r , имеющего схему $R(A_1, A_2, \dots, A_n)$, отображается в факт вида $p(a_1, a_2, \dots, a_n)$. Следовательно, если EDB представляет собой экстенционал, то все факты из EDB , имеющие один и тот же предикатный символ, хранятся в одном отношении.

2. Построение структуры для хранения правила вывода

Как было сказано выше, основным хранилищем фактов дедуктивной базы данных являются отношения реляционной базы данных. В связи с этим, определим структуру реляционных отношений для хранения и обработки дизъюнктов ДБД,

Из выражений (4, 5) следует, что каждое правило логической программы P , в общем случае, состоит из головы и тела, разделенных знаком импликации. Голова правила – это литерал определяющий имя продуцируемого объекта и список его аргументов. Каждый аргумент имеет четко определенную позицию. Тело правила вывода состоит из дизъюнкции некоторого числа предикатов. Каждый из них возвращает истинное значение, если его параметры соответствуют кортежу отношения, которое этот предикат представляет.

Поскольку дедуктивные системы являются системой интеграции логического программирования и баз данных, манипулирование данными в них осуществляется с помощью операций булевой алгебры.

Анализ структуры дизъюнктов позволяет построить схему базы данных для их хранения (рис. 1).

Отношение «Литералы» описывает литералы, входящие в состав правила и содержащие информацию об их именах, порядковых номерах в правилах и символах, стоящих после тела правила. Отношение «Аргументы» содержит данные об аргументах литералов: номере литерала в правиле, которому принадлежит аргумент, его позиции внутри литерала, а также его значение. Отношение «Логические операции» представляет собой справочник всех логических операций, определенных дедуктивной системой.

Рассмотрим пример. Пусть дано правило вывода вида:

$$p(A, B) : -r(A, X), d(X, B, M), M > 3. \quad (7)$$

Необходимо построить декомпозицию этого правила согласно схеме БД, представленной на рис. 1. Отметим, что обработка данных ненормализованной структуры БД (рис. 1) приводит к снижению эффективности формирования запроса, что в свою очередь снижает качество интеграции БД в целом.

3. Нормализация структуры для хранения правила вывода

Приведенная на рис. 1 схема БД не является окончательной. Для уменьшения избыточности хранимой информации необходимо провести нормализацию данной схемы, поскольку избыточность может привести к логически ошибочным результатам выборки либо изменения данных. Проведем нормализацию, проанализировав соответствующие функциональные зависимости (ФЗ).

Пусть R – некоторое отношение; $A, B \subseteq R$ – множества атрибутов, входящих в R . Определим условие существования функциональной зависимости между множествами A и B : если значения кортежей отношения R совпадают по атрибуту A , то они должны совпадать и по атрибуту B . Определим множество ФЗ, соответствующей рассматриваемой предметной области.

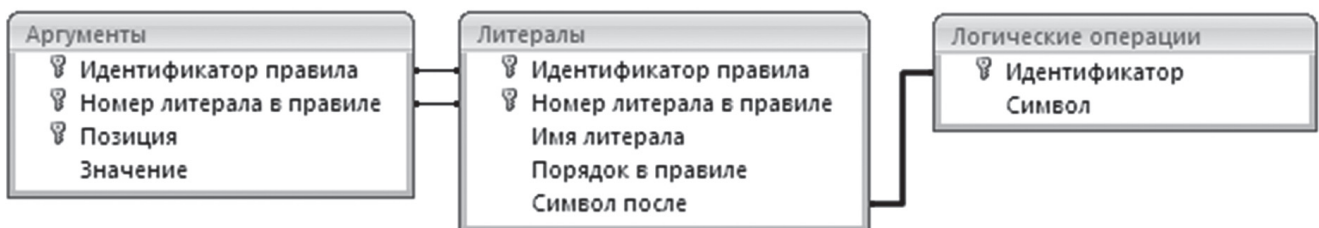


Рис. 1. Схема базы данных для хранения правил вывода

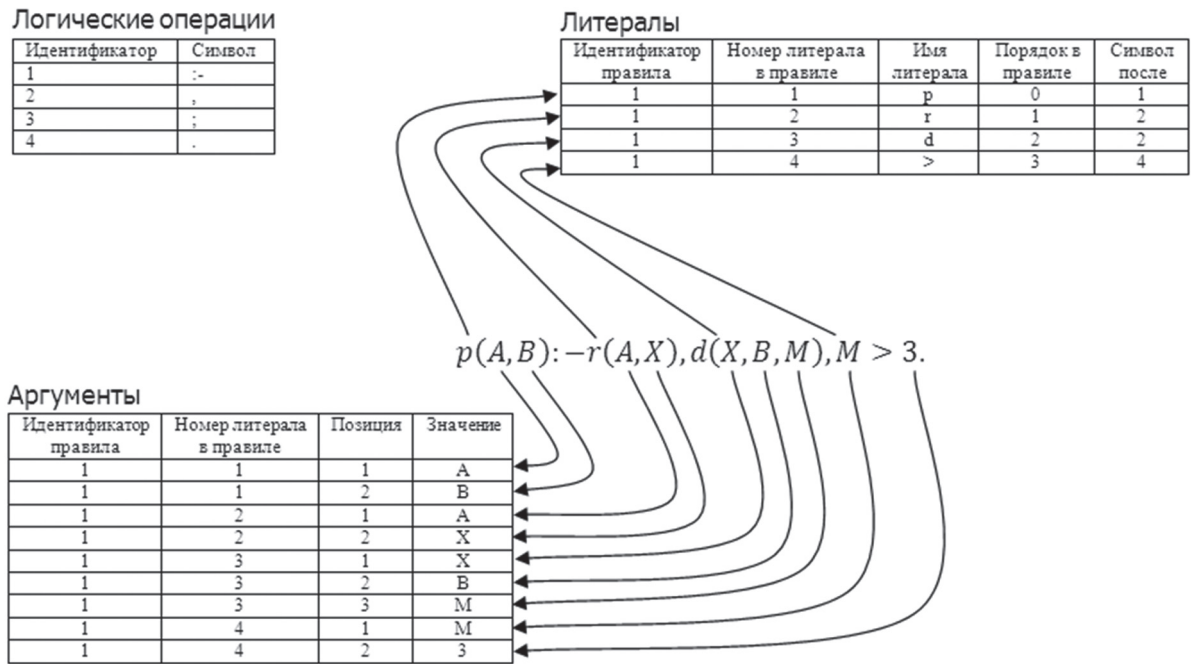


Рис. 2. Декомпозиция правила вывода

Отношение «Аргументы»:

F_1 : Ид. правила, Номер литерала, Позиция \rightarrow Значение;

F_2 : Номер литерала, Позиция \rightarrow Значение;

Отношение «Литералы»:

F_3 : Номер литерала \rightarrow Имя литерала, Символ после;

F_4 : Ид. Правила, Номер литерала \rightarrow Имя литерала, Порядок в правиле, Символ после;

Отношение «Логические операции»:

F_5 : Идентификатор \rightarrow Символ.

Основным требованием процедуры нормализации для перехода ко второй нормальной форме (2НФ) является отсутствие функционально неполной зависимости в заданном множестве F . В отношении «Аргументы» зависимость F_1 не является таковой, поскольку существует зависимость F_2 , детерминанта которой является подмножеством потенциального ключа отношения. Пусть K, L – детерминанты некоторых ФЗ, а M, N – множества зависимых атрибутов, тогда: $F' = K \rightarrow L, F'' = M \rightarrow N; M \subset K$ Для устранения функциональной неполноты необходимо

выделить F'' в отдельное отношение. Выделим F_2 в отношение «Аргументы». Атрибуты «Ид. правила», «Номер литерала» составляют отношение «Литералы в правиле». Аналогично для отношения «Литералы» F_3 необходимо оставить в отношении «Литералы», а оставшиеся атрибуты выделить в отдельное отношение «Правила».

Отношение «Литералы в правиле» может быть удалено из схемы, потому что его атрибуты дублируют аналогичные в отношении «Правила».

Полученная схема изображена на рис. 3.

Данная схема находится в третьей нормальной форме, поскольку не содержит транзитивных зависимостей (вида $F' = X \rightarrow Y, F'' = Y \rightarrow Z \Rightarrow X \rightarrow Z$).

Уменьшение избыточности хранимой информации для окончательной схемы БД заключается в точном отображении объектов предметной области в реляционную модель данных, то есть представление объекта «литерал» обрабатывается отдельно от конкретного идентификатора правила. Таким образом, один и тот же литерал может быть использован в нескольких правилах одновременно путем хранения в отношении «Правила»



Рис. 3. Конечная схема базы данных для хранения правил вывода

Логические операции

Идентификатор	Символ
1	:-
2	,
3	;
4	.

Правила

Идентификатор правила	Номер литерала	Порядок в правиле
1	1	0
1	2	1
1	3	2
1	4	3

Аргументы

Номер литерала в правиле	Позиция	Значение
1	1	A
1	2	B
2	1	A
2	2	X
3	1	X
3	2	B
3	3	M
4	1	M
4	2	3

Литералы

Номер литерала	Имя литерала	Символ после
1	r	1
2	r	2
3	d	2
4	>	4

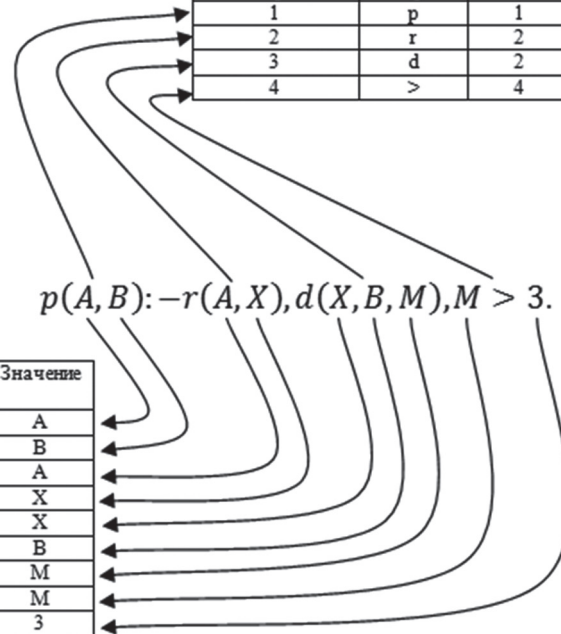


Рис. 4. Декомпозиция правила вывода для нормализованного набора отношений

соответствующего номера литерала. Аналогично из отношения «Аргументы» исключен атрибут «Идентификатор правила», вносящий дополнительную избыточность – аргумент имеет непосредственное отношение к литералу, но не к правилу.

Для иллюстрации приведенного алгоритма, рассмотрим декомпозицию правила (5) из предыдущего примера для нормализованного набора отношений (рис. 4).

Выводы

Предложенная логическая структура и технология декомпозиции правил вывода позволяют хранить экстенциональные и интенциональные компоненты дедуктивной базы данных в одном хранилище – реляционной базе данных. Это позволяет в значительной степени расширить возможность поддержки интенционала и увеличивает гибкость обработки экстенционала ДБД. Предложенная логическая структура для хранения правил разработана с соблюдением правил нормализации и является наиболее точным отображением рассматриваемой предметной области в реляционную модель данных с поддержкой целостности хранимой информации и устранением избыточности данных.

Список литературы: 1. Phipps G. Glue-Nail: A Deductive Database System [Текст] / G. Phipps, M. A. Derr, K. A. Ross // The VLDB Journal – The International Journal on Very Large Data Bases – Prototypes of deductive database systems. – 1991. – Т. 20, № 2. – С. 308-317. 2. Ramakrishnan R. The CORAL Deductive System [Текст]/ R. Ramakrishnan, D. Srivastava, S. Sudarshan, P. Seshadri // The VLDB Journal – The In-

ternational Journal on Very Large Data Bases - Prototypes of deductive database systems. – 1994. – Т. 3, №2. – С. 161-210. 3. Horn A. On Sentences Which are True of Direct Unions of Algebras [Текст]/ A.Horn // The Journal of Symbolic Logic. – Т. 16, № 1. – С. 14-21. 4. Дейт К. Дж. Введение в системы баз данных [Текст]: пер. с англ. - М.: Вильямс, 2006. – 1071 с. 5. Коннолли, Т. Базы данных: Проектирование, реализация, сопровождение. Теория и практика [Текст] / Т. Коннолли, К. Бегг, А. Страчан – М.: Издат. дом “Вильямс”, 2001. – 1120 с. 6. Чери, С. Логическое программирование и базы данных [Текст]: пер. с англ. / С. Чери, Г. Готлоб, Л. Танка. – М.: Мир, 1992. – 352 с.

Поступила в редколлегию 12.01.2012

УДК 004.652.4, 004.652.6

Організація збереження диз'юнктивних баз даних на основі реляційного підходу / Ю.А. Мальков, В.А. Радченко, С.С. Танянский // Біоніка інтелекту : наук.-техн. журнал. – 2012. – № 1 (78). – С. 59-62.

В статті розглянуто головні синтаксичні конструкції дедуктивних баз даних. На основі аналізу структури цих конструкцій пропонується метод їх декомпозиції в реляційні таблиці з метою збереження та ефективного використання.

Л. 4. Бібліогр.: 6 найм.

UDK 004.652.4, 004.652.6

Relational approach to store disjuncts of deductive databases / Y.A. Malkov, V.A. Radchenko, S.S. Tanyansky // Bionics of Intelligense: Sci. Mag. – 2012. – № 1 (78). – P. 59-62.

The article was focused on main syntactic structures of deductive databases. The analysis of disjunct's structure allows make it decomposition to relational schema for it saving.

Fig. 4. Ref.: 6 items.

УДК: 004.032.26:519.174.2



А.В. Шкловец, Н.Г. Аксак
Харьковский национальный университет радиоэлектроники, г. Харьков, Украина

ОПРЕДЕЛЕНИЕ РАССТОЯНИЙ МЕЖДУ ТОЧКАМИ НА КУСОЧНО-ГЛАДКИХ КАРТАХ КОХОНЕНА

В работе предложен метод для вычисления расстояний между данными, визуализированными с помощью кусочно-гладких карт Кохонена. Использование полученных соотношений для определения расстояний позволяет определять расстояния между точками многомерных данных с большей точностью и уменьшить ошибку визуализации структуры данных.

ТРИАНГУЛЯЦИЯ ДЕЛОНЕ, КАРТЫ КОХОНЕНА, ПАРАМЕТРИЧЕСКАЯ СПЛАЙН ПОВЕРХНОСТЬ, ОСТОВНОЙ ЛЕС

Введение

Как правило, данные, полученные в результате измерений какой-либо системы или процесса, описываются большим числом параметров. Для выбора метода анализа этих данных путем наглядного представления структуры данных или представления результатов анализа необходимо решить задачу визуализации многомерных данных. В [1] предложен метод визуализации многомерных данных с помощью карт Кохонена. Этот метод обладает рядом недостатков, устранение многих из которых предложено в работах [2–6]. В [7–9] предложены методы построения кусочно-гладких карт Кохонена путем аппроксимации кусочно-линейных карт Кохонена параметрическими сплайнами для решения проблемы отображения некоторых данных в одну точку или на грань карты, в результате чего возникали ошибки визуализации многомерных данных. Построение диаграмм Хиптона [2] на основе кусочно-гладких карт Кохонена требует введения новой метрики, учитывающей искажение карты, что позволяет точнее отображать структуру данных и находить расстояние между точками данных с меньшими ошибками, а также повышает точность визуализации данных.

1. Постановка задачи

Пусть для визуализации n -мерных данных $X = \{X^1, \dots, X^N\}$, $X^c \in R^n$, $c = \overline{1, N}$ имеются кусочно-линейные карты Кохонена, состоящие из l ($l \ll N$) выходных нейронов с матрицами

$$\rho_{\text{РК}}(\hat{X}^i, \hat{X}^j) = \begin{cases} ([\hat{X}^i] + 1 - \hat{X}^i) \rho(W^{[\hat{X}^i]}, W^{[\hat{X}^i] + 1}) + \sum_{k=[\hat{X}^i] + 1}^{[\hat{X}^j] - 1} \rho(W^k, W^{k+1}) + (\hat{X}^j - [\hat{X}^j]) \rho(W^{[\hat{X}^j]}, W^{[\hat{X}^j] + 1}), & \text{при } [\hat{X}^i] \neq [\hat{X}^j], \\ (\hat{X}^j - \hat{X}^i) \rho(W^{[\hat{X}^i]}, W^{[\hat{X}^i] + 1}), & \text{при } [\hat{X}^i] = [\hat{X}^j], \end{cases} \quad (1)$$

где $\rho(\bullet, \bullet)$ – евклидово расстояние в n -мерном пространстве, $[\bullet]$ – целая часть числа.

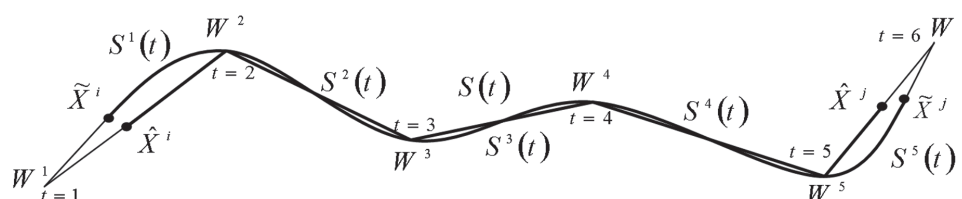


Рис. 1. Кратчайшая линия и Риманово расстояние на линейной карте

весовых коэффициентов W^i , $i = \overline{1, l}$, которые были аппроксимированы параметрическим сплайном $S(t)$ в одномерном случае, и параметрической сплайн поверхностью $S(t_1, t_2)$ в двумерном случае, где t, t_1, t_2 – параметры сплайна. В результате отображения многомерных данных X на кусочно-линейную карту Кохонена и на кусочно-гладкую карту Кохонена были построены множества \hat{X} и \tilde{X} , являющиеся визуализацией множества X . Множества \hat{X} и \tilde{X} принадлежат m -мерному пространству действительных чисел Θ^m ($m = \overline{1, 2}$).

Для увеличения точности отображения структуры данных на карте требуется определить расстояния между точками на кусочно-гладкой и кусочно-линейной картах Кохонена с учетом их искажений в n -мерном пространстве.

2. Одномерная карта Кохонена

Кратчайшее расстояние $\rho_{\text{РК}}(\hat{X}^i, \hat{X}^j)$ (1) между двумя точками \hat{X}^i и \hat{X}^j ($\hat{X}^i \leq \hat{X}^j$) на карте для дискретного и непрерывного отображений определяется как расстояние между этими точками по поверхности карты (рис. 1).

Для одномерной кусочно-гладкой карты Кохонена, представленной в виде параметрического сплайна $S(t)$, Риманово расстояние $\rho_{\text{РС}}(\tilde{X}^i, \tilde{X}^j)$ между двумя точками \tilde{X}^i и \tilde{X}^j ($\tilde{X}^i < \tilde{X}^j$) на сплайне определяется как длина сплайна между этими точками по формуле (2):

$$\rho_{\mathfrak{R}S}(\tilde{X}^i, \tilde{X}^j) = \begin{cases} \int_{\tilde{X}^i}^{[\tilde{X}^i]+1} \sqrt{\sum_{p=1}^n \left(\frac{dS_p^{[\tilde{X}^i]}(t)}{dt} \right)^2} dt + \sum_{k=[\tilde{X}^i]+1}^{[\tilde{X}^j]-1} \int_k^{k+1} \sqrt{\sum_{p=1}^n \left(\frac{dS_p^k(t)}{dt} \right)^2} dt + \int_{[\tilde{X}^j]}^{\tilde{X}^j} \sqrt{\sum_{p=1}^n \left(\frac{dS_p^{[\tilde{X}^j]}(t)}{dt} \right)^2} dt, & \text{при } [\tilde{X}^i] \neq [\tilde{X}^j], \\ \int_{\tilde{X}^i}^{\tilde{X}^j} \sqrt{\sum_{p=1}^n \left(\frac{dS_p^{[\tilde{X}^i]}(t)}{dt} \right)^2} dt, & \text{при } [\tilde{X}^i] = [\tilde{X}^j], \end{cases} \quad (2)$$

Для кусочно-линейной карты и параме трического сплайна $S(t)$ и пар точек \hat{X}^i, \hat{X}^j и \tilde{X}^i, \tilde{X}^j , приведенных на рис. 1, формулы (1) и (2) представятся в виде:

$$\rho_{\mathfrak{R}}(\hat{X}^i, \hat{X}^j) = \left(2 - \hat{X}^i\right) \rho(W^1, W^2) + \sum_{k=2}^5 \rho(W^k, W^{k+1}) + \left(\hat{X}^j - 5\right) \rho(W^5, W^6), \quad (3)$$

$$\rho_{\mathfrak{R}S}(\tilde{X}^i, \tilde{X}^j) = \int_{\tilde{X}^i}^2 \sqrt{\sum_{p=1}^n \left(\frac{dS_p^2(t)}{dt} \right)^2} dt + \sum_{k=2}^5 \int_k^{k+1} \sqrt{\sum_{p=1}^n \left(\frac{dS_p^k(t)}{dt} \right)^2} dt + \int_5^{\tilde{X}^j} \sqrt{\sum_{p=1}^n \left(\frac{dS_p^5(t)}{dt} \right)^2} dt, \quad (4)$$

3. Двумерная карта Кохонена

Рассмотрим кусочно-плоскую карту Кохонена

$$Z(t_1, t_2) = \bigcup_{k=1}^M Z^k(t_1, t_2), \quad (5)$$

где

$$Z^k(t_1, t_2) = W^{T_1^k} + t_1 \left(W^{T_2^k} - W^{T_1^k} \right) + t_2 \left(W^{T_3^k} - W^{T_1^k} \right) \quad (6)$$

на триангуляции $T = \{T^1, \dots, T^M\}$, в которой треугольник $T^k = \{T_1^k, T_2^k, T_3^k\}$ состоит из номеров нейронов $T_q^k \in \{1, \dots, l\}$, $q = 1, 3$, $k = 1, M$, M – количество треугольников.

Для нахождения расстояния между точками \hat{X}^i и \hat{X}^j с учетом растяжения карты необходимо найти кратчайшую ломаную на кусочной-плоской

поверхности $Z(t_1, t_2)$, соединяющую заданные точки. Расстояние между точками определяется как длина построенной ломаной. Решение рассматриваемой вариационной задачи очень трудоемко [11].

На рис. 2 представлено плоское изображение карты. Проведем прямую $\tilde{L}(\tau) = (t_1(\tau), t_2(\tau))$ между точками \hat{X}^i и \hat{X}^j в пространстве R^2 , где $\tilde{L}(0) = \hat{X}^i$ и $\tilde{L}(1) = \hat{X}^j$.

Пусть прямая $\tilde{L}(\tau)$ проходит через b треугольников $\{T^{k_1}, T^{k_2}, \dots, T^{k_b}\} \subset T$. Тогда существует $(b-1)$ точка τ^r , $r = 1, b-1$ пересечения прямой $\tilde{L}(\tau)$ с ребрами треугольников T^{k_r} и $T^{k_{r+1}}$. Построим ломаную $L(\tau)$, приближенную к кратчайшей между точками \hat{X}^i и \hat{X}^j так, чтобы её концами были точки \hat{X}^i и \hat{X}^j , и она терпела излом в точках $Z^{k_r}(t_1(\tau^r), t_2(\tau^r))$. Тогда её длина определяется соотношением (7):

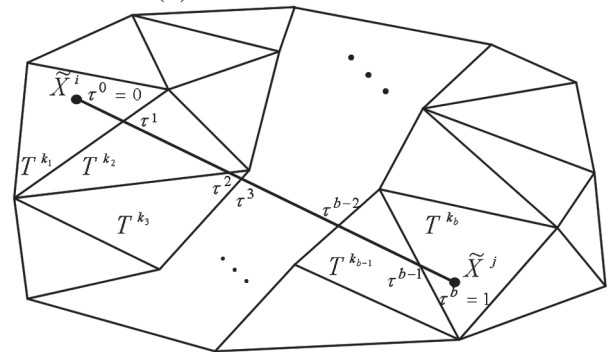


Рис 2. Ломаная, приближенная к кратчайшей для оценки расстояния между точками \hat{X}^i и \hat{X}^j

Формула (7) для вычисления расстояния не является метрикой, так как для неё не выполняется неравенство треугольника, однако она является хорошим приближением длины кратчайшей ломанной между точками \hat{X}^i и \hat{X}^j на поверхности карты $Z(t_1, t_2)$.

$$\rho_{\mathfrak{R}}(\hat{X}^i, \hat{X}^j) = \begin{cases} \rho\left(Z\left(\hat{X}^i\right), Z^{k_1}\left(t_1\left(\tau^1\right), t_2\left(\tau^1\right)\right)\right) + \sum_{r=2}^{b-1} \rho\left(Z^{k_{r-1}}\left(t_1\left(\tau^{r-1}\right), t_2\left(\tau^{r-1}\right)\right), Z^{k_r}\left(t_1\left(\tau^r\right), t_2\left(\tau^r\right)\right)\right) + \\ + \rho\left(Z^{b-1}\left(t_1\left(\tau^{b-1}\right), t_2\left(\tau^{b-1}\right)\right), Z\left(\hat{X}^j\right)\right) & \text{при } b > 0, \\ \rho\left(Z\left(\hat{X}^i\right), Z\left(\hat{X}^j\right)\right) & \text{при } b = 0. \end{cases} \quad (7)$$

Рассмотрим кусочно-гладкую карту Кохонена, представленную сплайн поверхностью $S(t_1, t_2)$. Для нахождения Риманова расстояния [12] между точками \tilde{X}^i и \tilde{X}^j с учетом растяжения сплайн поверхности необходимо построить геодезическую линию на $S(t_1, t_2)$, соединяющую точки \tilde{X}^i

и \tilde{X}^j . То есть, необходимо найти такую кривую $L(\tau) = (t_1(\tau), t_2(\tau)) \in S(t_1, t_2)$, $L(\tau^0) \in \tilde{X}^i$, $L(\tau^1) \in \tilde{X}^j$, которая является решением задачи (8)-(10)

$$\int_{\tau^0}^{\tau^1} U\left(\tau, t_1(\tau), t_2(\tau), \frac{dt_1}{d\tau}, \frac{dt_2}{d\tau}\right) d\tau \rightarrow \min, \quad (8)$$

где

$$U\left(\tau, t_1(\tau), t_2(\tau), \frac{dt_1}{d\tau}, \frac{dt_2}{d\tau}\right) = \sqrt{E(t_1(\tau), t_2(\tau)) \frac{d^2 t_1(\tau)}{d\tau^2} + 2F(t_1(\tau), t_2(\tau)) \frac{dt_1(\tau)}{d\tau} \frac{dt_2(\tau)}{d\tau} + G(t_1(\tau), t_2(\tau)) \frac{d^2 t_2(\tau)}{d\tau^2}}, \quad (9)$$

$$\begin{cases} E(t_1, t_2) = \bigcup_{k=1}^M E^k(t_1, t_2); \\ F(t_1, t_2) = \bigcup_{k=1}^M F^k(t_1, t_2); \\ G(t_1, t_2) = \bigcup_{k=1}^M G^k(t_1, t_2); \end{cases} \quad \begin{cases} E^k(t_1, t_2) = R_1^k(t_1, t_2) \cdot R_1^k(t_1, t_2); \\ F^k(t_1, t_2) = R_1^k(t_1, t_2) \cdot R_2^k(t_1, t_2); \\ G^k(t_1, t_2) = R_2^k(t_1, t_2) \cdot R_2^k(t_1, t_2); \end{cases} \quad \begin{cases} R_1^k(t_1, t_2) = \left(\frac{\partial S_1^k(t_1, t_2)}{\partial t_1}, \dots, \frac{\partial S_n^k(t_1, t_2)}{\partial t_1} \right); \\ R_2^k(t_1, t_2) = \left(\frac{\partial S_1^k(t_1, t_2)}{\partial t_2}, \dots, \frac{\partial S_n^k(t_1, t_2)}{\partial t_2} \right); \end{cases} \quad (10)$$

(••) - скалярное произведение векторов. Тогда расстояние между точками \tilde{X}^i и \tilde{X}^j определяется значением минимума целевой функции в задаче (8). Первая квадратичная форма (9) определяет метрический тензор на $S(t_1, t_2)$.

Решение вариационной задачи (8) эквивалентно решению системы (11)

$$\begin{cases} \frac{d}{d\tau} \left(\frac{\partial U}{\partial t_1'}(\tau) \right) - \frac{\partial U}{\partial t_1} = 0, \\ \frac{d}{d\tau} \left(\frac{\partial U}{\partial t_2'}(\tau) \right) - \frac{\partial U}{\partial t_2} = 0, \end{cases}$$

где $\begin{cases} \frac{\partial U}{\partial t_1'}(\tau) = \frac{\partial U(\tau, t_1(\tau), t_2(\tau), \xi_1, \xi_2)}{\partial \xi_1} \Big|_{\xi_1 = \frac{dt_1}{d\tau}, \xi_2 = \frac{dt_2}{d\tau}} \\ \frac{\partial U}{\partial t_2'}(\tau) = \frac{\partial U(\tau, t_1(\tau), t_2(\tau), \xi_1, \xi_2)}{\partial \xi_2} \Big|_{\xi_1 = \frac{dt_1}{d\tau}, \xi_2 = \frac{dt_2}{d\tau}} \end{cases} . \quad (11)$

Система дифференциальных уравнений второго порядка Эйлера-Лагранджа (11) аналитически не может быть решена. Проблема численного решения заключается в том, что коэффициенты системы – суть кусочно-гладкие функции, которые на каждой подобласти T^k описываются многочленом третьего порядка, и неизвестно, по каким областям T^k будет проходить геодезическая кривая.

Как и в случае кусочно-гладкой карты, построим прямую $\tilde{L}(\tau) = (t_1(\tau), t_2(\tau))$, проходящую через b подобластей $\{T^{k_1}, T^{k_2}, \dots, T^{k_b}\} \subset T$ и имеющую $(b-1)$ точку τ^r , $r = \overline{1, b-1}$ пересечения прямой $\tilde{L}(\tau)$ с границами подобластей T^{k_r} и $T^{k_{r+1}}$.

Построим кусочно-гладкую кривую $L(\tau) = \bigcup_{k=1}^M L^k(\tau)$ на поверхности $S(t_1, t_2)$ в простран-

стве R^n , приближенную к геодезической, соединяющей точки \tilde{X}^i и \tilde{X}^j так, чтобы её концами были точки \tilde{X}^i и \tilde{X}^j , и на каждой подобласти T^{k_r} она описывалась соотношением

$$L^r(\tau) = S^{k_r}(\tilde{L}(\tau)), \quad r = \overline{1, b}, \quad (12)$$

$$\tau \in \begin{cases} [0, \tau^1], & \text{при } r = 1 \\ [\tau^{r-1}, \tau^r], & \text{при } r = \overline{2, b-1} \\ [\tau^{b-1}, 1], & \text{при } r = b-1 \end{cases} \quad (13)$$

Тогда её длина определяется соотношением (14):

$$\rho_{\text{экс}}(\tilde{X}^i, \tilde{X}^j) = \begin{cases} \int_0^{\tau^1} \sqrt{\sum_{p=1}^n \left(\frac{dL_p^1(\tau)}{d\tau} \right)^2} d\tau + \\ + \sum_{k=2}^{b-1} \int_{\tau^{k-1}}^{\tau^k} \sqrt{\sum_{p=1}^n \left(\frac{dL_p^k(\tau)}{d\tau} \right)^2} d\tau + \\ + \int_{\tau^{b-1}}^1 \sqrt{\sum_{p=1}^n \left(\frac{dL_p^b(\tau)}{d\tau} \right)^2} d\tau, \quad \text{при } b > 0, \\ \int_0^1 \sqrt{\sum_{p=1}^n \left(\frac{dL_p^1(\tau)}{d\tau} \right)^2} d\tau, \quad \text{при } b = 0, \end{cases} \quad (14)$$

где $(L_1^k(\tau), \dots, L_n^k(\tau)) = L^k(\tau)$.

Формула (14) для вычисления расстояния не является метрикой, так как для неё не выполняется условие неравенства треугольника, однако она является хорошим приближением Римановой метрики.

4. Численные эксперименты

Для определения качества визуализации данных картами Кохонена предлагается использовать следующие критерии:

1. Критерий ошибки визуализации данных в евклидовой метрике (оценка изменения в структуре данных после отображения на карту)

$$K_v = \sqrt{\frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left(\frac{\rho(X^i, X^j)}{V} - \frac{\rho(\tilde{X}^i, \tilde{X}^j)}{\tilde{V}} \right)^2}, \quad (15)$$

где

$$V = \max_{\substack{i=1, N-1 \\ j=i+1, N}} \rho(X^i, X^j), \quad \tilde{V} = \max_{\substack{i=1, N-1 \\ j=i+1, N}} \rho(\tilde{X}^i, \tilde{X}^j) \quad (16)$$

2. Критерий ошибки визуализации данных в Римановой метрике

$$K_{v_{\text{РМ}}} = \sqrt{\frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left(\frac{\rho_{\text{РМ}}(X^i, X^j)}{V} - \frac{\rho_{\text{РМ}}(\tilde{X}^i, \tilde{X}^j)}{\tilde{V}} \right)^2}, \quad (17)$$

где

$$V = \max_{\substack{i=1, N-1 \\ j=i+1, N}} \rho_{\text{РМ}}(X^i, X^j), \quad \tilde{V} = \max_{\substack{i=1, N-1 \\ j=i+1, N}} \rho_{\text{РМ}}(\tilde{X}^i, \tilde{X}^j), \quad (18)$$

для кусочно-линейной или кусочно-плоской карт Кохонена и

$$K_{v_{\text{РМ}}} = \sqrt{\frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \left(\frac{\rho_{\text{РМ}}(X^i, X^j)}{V} - \frac{\rho_{\text{РМ}}(\tilde{X}^i, \tilde{X}^j)}{\tilde{V}} \right)^2}, \quad (19)$$

где

$$V = \max_{\substack{i=1, N-1 \\ j=i+1, N}} \rho_{\text{РМ}}(X^i, X^j), \quad \tilde{V} = \max_{\substack{i=1, N-1 \\ j=i+1, N}} \rho_{\text{РМ}}(\tilde{X}^i, \tilde{X}^j) \quad (20)$$

для кусочно-гладких карт Кохонена.

Численные эксперименты проводились для базы данных рынка полимеров компаний из стран СНГ. Визуализация компаний проводилась на основе прямоугольной карты Кохонена размером 9Ч9. Компании характеризовались следующими показателями: страна, вид полимера, прирост с начала года в тоннах и процентах, прирост за последний месяц в тоннах и процентах, доля в импорте с начала года и с начала месяца в процентах, импорт с начала года и за последний месяц в тоннах.

Результаты сравнительного анализа методов визуализации данных по предложенным критериям приведены в табл. 1.

Из табл. 1 видно, что использование предложенного метода вычисления расстояния с учетом искажения карты Кохонена значительно уменьшает

ошибку визуализации многомерных данных по сравнению с евклидовой метрикой. При использовании кусочно-гладких карт Кохонена ошибка несколько больше, чем при кусочно-линейных картах, что вызвано не всегда желательным изгибом сплайна.

Выводы

Для улучшения отображения структуры данных при их визуализации с помощью кусочно-плоских и кусочно-гладких карт Кохонена в работе впервые предложены соотношения для приближенного вычисления римановой метрики на карте данных. Проведенные эксперименты показали, что предложенные соотношения позволяют значительно уменьшить ошибку визуализации многомерных данных.

Список литературы: 1. Kohonen T. Self-Organizing Maps. Springer: Berlin-Heidelberg, 1997. 2. Зиновьев, А.Ю. Визуализация многомерных данных [Текст] / А. Ю. Зиновьев. – Красноярск: Изд-во КГТУ, 2000. – 168 с. 3. В. Back, К. Sere, Н. Vanharanta. Analyzing Financial Performance with Self-Organized Maps. // Proc. of International joint conference on Neural NetWorks. Washington, DC, 1998. 4. K.Kiviluoto, P. Bergius. two-level self-organizing-map's for analysis of financial statement. // Proc. Of international joint conference on Neural NetWorks. Washington, DC, 1998. 5. M. Chang, H. Yu, J. Heh. Evolutionary Self-Organizing map // proc. of international joint conference on Neural NetWorks. Washington, DC, 1998. 6. M.-C.Su, T.-K.Liu, H.-T. Chang. An efficient initialization scheme for the self-organizing feature map algorithm. // Proc. Of International Joint Conference on Neural NetWorks. Washington, DC, 1999. 7. Аксак, Н.Г. Метод аппроксимации карт Кохонена кубическим параметрическим сплайном [Текст] / Н. Г. Аксак, А. В. Шкловец // Системи управління навігації та зв'язку. – 2010. – 2(14). – с. 70-74. 8. Шкловец, А.В. Метод аппроксимации сплайнами минимальной длины линейных карт Кохонена для визуализации многомерных данных [Текст] / А. В. Шкловец, Н. Г. Аксак // XIII Всероссийская научно-техническая конференция «Нейроинформатика-2011»: Сб. науч. трудов. В 3 частях. Ч.1. М.: НИЯУ МИФИ, 2010. – 280 с. 9. Шкловец, А.В. Аппроксимация двумерных карт Кохонена кубическими сплайн поверхностями [Текст] / А.В. Шкловец, Н.Г. Аксак // сб. науч. тр. – Евпатория: МОИСУ, 2010. – 225 с. 10. Шкловец, А.В. Метод аппроксимации сплайн поверхностями кусочно-плоских карт Кохонена для визуализации многомерных данных [Текст] / А. В. Шкловец, Н. Г. Аксак // XIII всероссийская научно-техническая конференция «Нейроинформатика-2012»: Сб. науч.

Таблица 1

Сравнительный анализ методов визуализации многомерных данных

Критерий	Одномерная карта Кохонена		Двумерная карта Кохонена	
	Кусочно-линейная	Кусочно-гладкая	Кусочно-плоская	Кусочно-гладкая
K_v	0,2705	0,2324	0,2940	0,2580
$K_{v_{\text{РМ}}}$	0,1295	0,1302	0,0756	0,0779

трудов. В 3 частях. Ч.1. М.: НИЯУ МИФИ, 2011. – 280 с.
11. Люстерник, Л.А. Кратчайшие линии: Вариационные задачи [Текст] / Л. А. Люстерник. – М.: Гостехиздат, 1955. – 104 с. **12.** Эльсгольц, Л.Э. Дифференциальные уравнения и вариационное исчисление [Текст] / Л. Э. Эльсгольц. – М.: Наука, 1969. – 367 с.

Поступила в редколлегию 16.01.2012

УДК 004.032.26:519.174.2

Вимір відстаней між точками на кусочно-гладкій карті Кохонена / А.В. Шкловець, Н.Г. Аксак // Біоніка інтелекту: наук.-техн. журнал. – 2012. – № 1 (78). – С. 63-67.

В роботі запропоновано метод для обчислення відстаней між даними, що були візуалізовані за допомогою кусочно-гладких карт Кохонена. Використання отриманих співвідношень для виміру відстаней дозволяє

визначати відстані між точками багатовимірних даних з великою точністю та зменшити похибку візуалізації структури даних.

Табл 1. Іл. 2. Бібліогр.: 12. найм.

UDK 004.032.26:519.174.2

Distance determination between points on the piecewise-smooth Kohonen maps. / A.V. Shklovets, N.G. Axak // Bionics of Intelligense: Sci. Mag. – 2012. – № 1 (78). – P. 63-67.

A method to calculate the distances between the data visualized using a piecewise-smooth Kohonen maps was suggested. Using the obtained relations for the determination of distances allows determining the distance between the points of multidimensional data with greater accuracy and reducing error rendering data structures.

Tab. 1. Fig.2. Ref.: 12 Items

УДК 615.47:617-089

А.И. Бых¹, Т.В. Жемчужкина², Т.В. Носова³, В.А. Сухенький⁴¹ХНУРЭ, г. Харьков, Украина, bykh@kture.kharkov.ua;²ХНУРЭ, г. Харьков, Украина, zhemchuzhkina@rambler.ru;³ХНУРЭ, г. Харьков, Украина, E-mail: yasanosova@rambler.ru;⁴ХНУРЭ, г. Харьков, Украина, sva89kolomiets@gmail.com

ПОИСК ИНФОРМАТИВНЫХ КОЛИЧЕСТВЕННЫХ ПОКАЗАТЕЛЕЙ ЭЛЕКТРОМИОГРАФИЧЕСКОГО СИГНАЛА. СООБЩЕНИЕ 3

В результате проведения исследований методом дискриминантного анализа был произведен отбор информативных количественных показателей ЭМГ сигнала и найдены коэффициенты канонических дискриминантных функций. Проведена классификация 65-ти ЭМГ сигналов в норме и патологии, точность которой составила 95,4%.

ЭЛЕКТРОМИОГРАММА, ДИСКРИМИНАНТНЫЙ АНАЛИЗ, СТАТИСТИЧЕСКИЕ ХАРАКТЕРИСТИКИ, ТЕРРИТОРИАЛЬНАЯ КАРТА

Введение

На сегодняшний день исследование электрической активности мышц становится все более популярным среди врачей-диагностов. Электромиография (ЭМГ) является единственным объективным и информативным методом исследования функционального состояния периферической нервной системы, патология которой в структуре неврологических заболеваний занимает ведущее место [1]. Электромиографические исследования позволяют не только установить характер заболевания, проводить его топическую диагностику, но и объективно контролировать эффективность лечения, прогнозировать время и этапы восстановления пациента.

Автоматизированные системы измерения и обработки медико-биологической информации, использующие современные программные средства, существенно расширяют диагностические возможности современной медицины. Это касается и электромиографии – метода исследования нервно-мышечной системы посредством регистрации электрических потенциалов мышц.

Все патологические характеристики ЭМГ определяются врачом по внешним признакам, то есть носят субъективный характер. Таким образом, постановка диагноза зависит от квалификации, опыта, интуиции и физиологического и эмоционального состояния врача. Несмотря на широкое применение анализа ЭМГ в клинической медицине, остается актуальным получение информативных показателей ЭМГ–сигнала для улучшения диагностирования патологий нервно-мышечной системы. Более подробное изучение и анализ ЭМГ позволят обеспечить не только дифференцирование патологий, но и обнаруживать эти патологии на более ранних стадиях, а также сократить время диагностирования.

Актуальной задачей является определение количественных характеристик ЭМГ–сигнала при патологических состояниях нервно-мышечной

системы и поиска наиболее точных методов классификации и прогнозирования заболеваний.

1. Постановка задачи

Целями данной работы являются:

- 1) отбор информативных показателей ЭМГ–сигналов дискриминантным методом;
- 2) классификация пациентов по диагностическим группам;
- 3) изучение эффективности использования выбранного метода путем сравнения полученных результатов с диагнозами, поставленными врачами-экспертами.

В сообщениях 1, 2 [2, 3] нами были рассмотрены количественные показатели интерференционного ЭМГ–сигнала, полученные методами статистического исследования биомедицинских сигналов. Было установлено, что рассчитанные характеристики являются статистически независимыми и могут быть использованы для дифференциальной диагностики состояния нервно-мышечной ткани. Поэтому их мы и будем использовать в качестве исходных данных.

Для решения поставленной задачи был выбран метод дискриминантного анализа. Метод отвечает нашим требованиям, поскольку его использование позволяет:

1. Ответить на вопрос о том, возможно ли, используя данный набор параметров, отличить одну группу от другой; насколько хорошо эти параметры помогают провести дискриминацию; и какие из них наиболее информативны.
2. Получить несколько функций, обеспечивающих возможность отнесения данного объекта к одной из групп. Эти функции называются классифицирующими и зависят от значений параметров таким образом, что появляется возможность отнести каждый объект к одной из групп.

Таким образом, нам удастся получить принципиально новые результаты: как научные (уточнение информативных характеристик для проведения

дальнейших исследований), так и практические (возможность классификации ЭМГ-сигналов по группам нормы и разной степени патологии). Это, в свою очередь, позволит сократить время диагностирования и обеспечить не только дифференцирование патологий, но и обнаруживать эти патологии на более ранних стадиях.

2. Решение задачи

С помощью дискриминантного анализа на основании некоторых признаков (независимых переменных) индивидуум может быть причислен к одной из двух (или к одной из нескольких) заданных заранее групп. Ядром дискриминантного анализа является построение так называемой канонической дискриминантной функции

$$d_{km} = \beta_0 + \beta_1 x_{1km} + \dots + \beta_p x_{pkm},$$

где: d_{km} – значение дискриминантной функции для m -го объекта в группе k ; x_{pkm} – значение дискриминантной переменной x_1 для m -го объекта в группе k , соответствующих рассматриваемым случаям; $\beta_0 \dots \beta_1$ – неизвестные коэффициенты.

Целью анализа является определение таких коэффициентов, с помощью которых дискриминантные функции дали бы возможность с максимальной точностью провести разделение по группам [4].

Наиболее общим принципом применения дискриминантного анализа является включение в исследование по возможности большего числа переменных с целью определения тех из них, которые наилучшим образом разделяют выборки между собой. Пошаговая процедура дискриминантного анализа для отбора переменных основывается на F -критериях однофакторного дисперсионного анализа: “ F -включения” и “ F -исключения”. Значение F -статистики для переменной указывает на ее статистическую значимость при дискриминации между совокупностями и является мерой вклада признака в предсказание членства в группах. Тогда в качестве результата успешного анализа можно сохранить только наиболее информативные переменные модели, то есть те переменные, чей вклад в дискриминацию больше остальных [5].

В исследовании мы использовали количественные статистические характеристики 65-ти ЭМГ-сигналов (снятых с нескольких мышц пациентов в норме и патологии), полученные нами ранее [2, 3], а именно: минимум и максимум выборки амплитуд (MIN , MAX), вариационный размах ($Delta$), математическое ожидание (M), среднее квадратическое отклонение (SKO), дисперсия (D), коэффициент вариации (V), асимметрия (As), эксцесс (Ex), мода (Mo), амплитуда моды (AMo), амплитуда моды в процентах (AMo_perc), треугольный индекс (TI), индекс наклона гистограммы (N), индекс

напряженности (IN), значение периодограммы в нулевой частоте (PO), средняя спектральная мощность (SP), значение спектра в нулевой частоте (SO), эффективная ширина спектра (wS), средняя мощность периодограммы (PP), эффективная ширина периодограммы (wP), погрешность аппроксимации амплитуд волн ЭМГ нормальным распределением ($deltaNorm$), погрешность аппроксимации длительностей волн ЭМГ экспоненциальным распределением ($deltaExp$).

Очевидно, что классификация переменных будет осуществляться тем лучше, чем меньше рассеяние точек относительно центроида внутри группы и чем больше расстояние между центроидами групп. Один из методов поиска наилучшей дискриминации данных заключается в нахождении такой канонической дискриминантной функции d , которая максимизировала бы отношение межгрупповой вариации к внутригрупповой:

$$\lambda = B(d) / W(d),$$

где B межгрупповая и W внутригрупповая матрицы рассеяния наблюдаемых переменных от средних [5, 6].

Классификация объектов в дискриминантном методе производится с помощью функции расстояния. В качестве меры сходства можно использовать евклидово расстояние между объектами. Чем меньше расстояние между объектами, тем больше сходство. Однако в тех случаях, когда переменные коррелированы, измерены в разных единицах и имеют различные стандартные отклонения, полезнее применить не евклидово расстояние, а выборочное расстояние Махаланобиса

$$D^2 = \sum_{k=1}^p n_k \times (x_k - \bar{x}) \times C^{-1} \times (x_k - \bar{x}).$$

При использовании функции расстояния, объект относят к той группе, для которой расстояние D^2 наименьшее. На содержательном уровне его можно интерпретировать как взвешенную сумму расстояний от вектора средних каждой группы x_k до общего вектора средних \bar{x} . Если гипотеза H_0 верна, а объем выборки стремится к ∞ (достаточно большой), то D^2 может быть аппроксимирована F -распределением. Другим, в некоторых случаях более точным способом проверки гипотезы H_0 , является использование U -статистики Уилкса (она же – лямбда Вилкса), которая вычисляется как отношение детерминантов (\det) матрицы внутригрупповой ковариации W и полной ковариационной матрицы T

$$U = \det(W) / \det(T).$$

В некоторых работах для классификации используется обобщенное расстояние Махаланобиса V (обобщение величины D^2). Эта мера, известная как V -статистика Рао (аппроксимация статистики

U-Уилкса с помощью F-распределения), измеряет расстояния от каждого центроида группы до главного центроида с весами, пропорциональными объему выборки соответствующей группы. Она применима при любом количестве классов и может быть использована для проверки гипотезы H0. Отметим, что при включении или исключении переменных U-статистика имеет распределение χ-квадрат с числом степеней свободы, равным количеству групп -1, умноженное на число переменных, включенных (исключенных) на этом шаге. Если изменение статистики не значимо, то переменную можно не включать. Если после включения новой переменной V-статистика оказывается отрицательной, то это означает, что включенная переменная ухудшает разделение центроидов.

В ходе проведения дискриминантного анализа на каждом шаге (максимальное число шагов равно 48), включалась переменная, минимизирующая индикатор “лямбда Уилкса”. Максимальная значимость F-включения переменной составляла 0,05, а минимальная значимость F-исключения – 0,10.

3. Основные результаты

В результате проведения анализа были определены переменные, которые вносят значительный статистический вклад в дискриминацию элементов по диагностическим группам. Таким образом, делаем вывод, что информативными параметрами являются: *M, As, Ex, Mo, AMo, TI, N, IN, P0, SP, SO, wS, wP, deltaNorm, deltaExp*, т.е. 15 из 23 выбранных для анализа параметров.

Полученные нормированные канонические дискриминантные функции имеют вид:

$$d_1 = -1.931 \times M - 1.223 \times As + 2.726 \times Ex + 0.172 \times AMo + 0.183 \times Mo + 2.137 \times TI + 0.931 \times N - 0.644 \times IN - 1.596 \times SP - 0.055 \times SO + 1.542 \times wS - 3.048 \times P0 + 0.295 \times wP - 3.274 \times deltaNorm - 0.304 \times deltaExp;$$

$$d_2 = 0.624 \times M + 0.439 \times As - 1.31 \times Ex - 4.105 \times AMo + 1.431 \times Mo - 0.169 \times TI - 0.654 \times N + 0.588 \times IN + 0.713 \times SP + 0.839 \times SO + 2.391 \times wS - 1.902 \times P0 + 0.861 \times wP + 3.396 \times deltaNorm + 0.64 \times deltaExp;$$

$$d_3 = 1.334 \times M + 0.198 \times As + 0.211 \times Ex + 0.778 \times AMo + 1.062 \times Mo + 0.054 \times TI + 0.669 \times N - 0.226 \times IN + 0.606 \times SP + 0.382 \times SO + 1.315 \times wS - 2.313 \times P0 + 1.909 \times wP - 0.917 \times DeltaNorm + 0.302 \times deltaExp;$$

Из рис. 1 видно, что показатели расположены отдельно по диагнозам. Это говорит о правильном выборе значимых количественных показателей электромиографического сигнала.

На территориальной карте (рис. 2), показано разделение на области, которые означают принадлежность к группе. При этом в пределах границ

соответствующей области вероятность отнесения к данной группе выше, чем для других групп. На границах областей вероятности для граничащих групп одинаковы. Символы, используемые в территориальной карте, приведены в табл. 1.

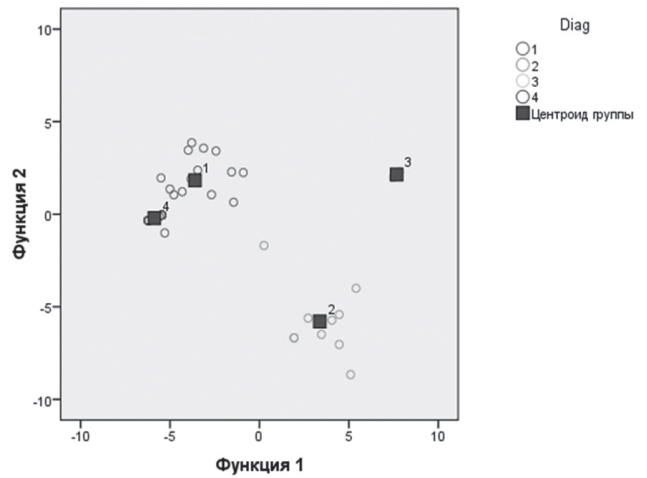


Рис. 1. Размещение центроидов групп и показателей в пространстве признаков

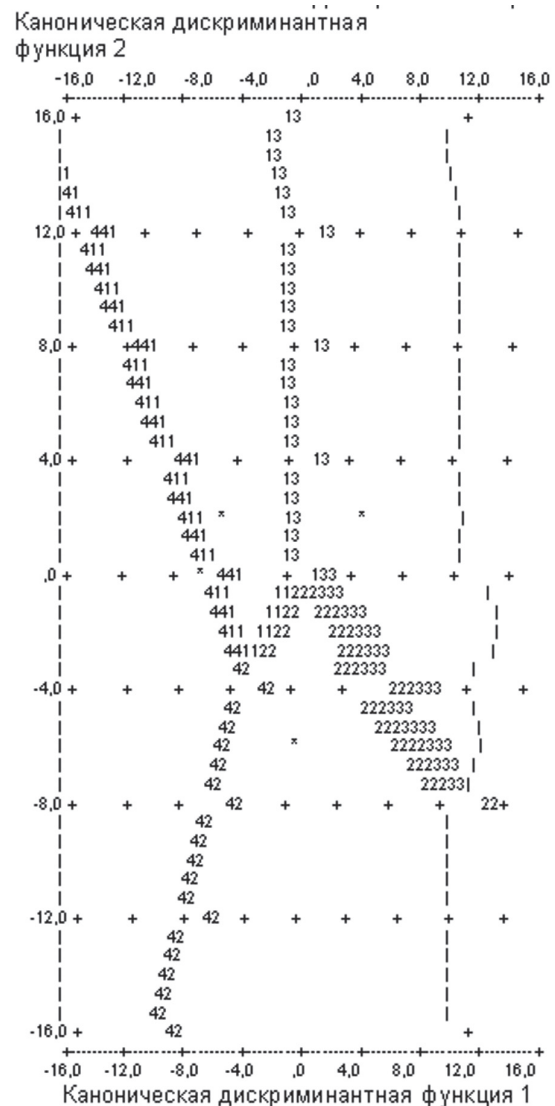


Рис. 2. Территориальная карта

В результате проведения дискриминантного анализа был определен набор диагностически значимых количественных показателей ЭМГ-сигнала, а именно: математическое ожидание, асимметрия, эксцесс, мода, амплитуда моды, триангулярный индекс, индекс наклона гистограммы, индекс напряженности, значение периодограммы в нулевой частоте, средняя спектральная мощность, значение спектра в нулевой частоте, эффективная ширина спектра, эффективная ширина периодограммы, погрешность аппроксимации амплитуд волн ЭМГ нормальным распределением, погрешность аппроксимации длительностей волн ЭМГ нормальным распределением.

Таблица 1

Символы, используемые в территориальной карте

Символ	Группа	Метка
1	1	Диагноз: норма
2	2	Диагноз: слабые нарушения
3	3	Диагноз: сильные нарушения
4	4	Диагноз: денервация
*	центр группы	

В дискриминантном анализе процедура классификации используется для определения принадлежности к той или иной группе случайно выбранных объектов, которые не были включены при выводе дискриминантной и классифицирующих функций. Для проверки точности классификации применим классифицирующие функции к тем объектам, по которым они были получены. По количеству правильно классифицированных объектов можно оценить точность процедуры классификации. Результаты такой классификации представлены в табл. 2.

Таблица 2

Классификационная таблица

		Результаты классификации					
		Диагноз	Предсказанная принадлежность к группе, %				Итого
			1	2	3	4	
Группа	1	87,5	0,0	0,0	12,5	100,0	
	2	9,1	90,9	0,0	0,0	100,0	
	3	0,0	0,0	100,0	0,0	100,0	
	4	0,0	0,0	0,0	100,0	100,0	

Выводы

При помощи найденных дискриминантных функций была проведена классификация ЭМГ-сигналов в норме и при разной степени патологии. Фактическая и прогнозируемая группы не совпали

для 3 из 65 объектов. Таким образом, точность классификации при данном наборе дискриминантных переменных составляет 95,4%.

Высокая эффективность использования выбранного метода обеспечивает возможность широкого применения полученных результатов в практической диагностике, благодаря параметризации и переводу анализа в количественную область, возможности проводить классификацию произвольных объектов по найденным классификационным функциям.

Список литературы: 1. Гехт, Б.М. Теоретическая и клиническая электромиография [Текст] / Б.М. Гехт. – Л.: Наука, 1990. – 229 с. 2. Бых, А.И. Поиск информативных количественных показателей электромиографического сигнала. Сообщение 1 [Текст] / А.И. Бых, Т.В. Жемчужкина, Т.В. Носова // Бионика интеллекта. – 2007. – № 1 (66). – С. 18–125. 3. Бых, А.И., Поиск информативных количественных показателей электромиографического сигнала. Сообщение 2. [Текст] / А.И. Бых, Т.В. Жемчужкина, Т.В. Носова // Бионика интеллекта. – 2008. – № 1 (68). – С. 111–116. 4. Ким, Дж.-О. Факторный, дискриминантный и кластерный анализ [Текст] / Дж.-О. Ким, Ч.У. Мьюллер, У.Р. Клекка и др. М.: 1989. 5. Енюков, И.С. Факторный, дискриминантный и кластерный анализ: сборник работ под ред. Енюкова И. С. – М.: Финансы и статистика, 1989. – 215 с. 6. Айвазян, С.А. Прикладная статистика: Классификация и снижение размерности [Текст] / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин. – М.: Финансы и статистика, 1989. – 607 с.

Поступила в редколлегию 20.01.2012

УДК 615.47:617-089

Пошук інформативних кількісних показників електроміографічного сигналу. Повідомлення 3 / А.І. Бих, Т.В. Жемчужкіна, Т.В. Носова, В.А. Сухенький // Біоніка інтелекту: наук.-техн. журнал. – 2012. – № 1 (78). – С. 68-71.

Робота присвячена пошуку інформативних кількісних показників ЕМГ-сигналів для диференціальної діагностики стану нервово-м'язової системи. Для відбору діагностично значущих статистичних параметрів застосовується метод дискримінантного аналізу.

Табл. 2. Іл. 2. Бібліогр.: 6 найм.

UDC 615.47:617-089

Searching for informative quantitative parameters of electromyographic signal / A.I. Bykh, T.V. Zhemchuzhkina, T.V. Nosova, V.A. Suchenkyi // Bionics of Intelligense: Sci. Mag. – 2012. – № 1 (78). – P. 68-71.

The work is dedicated to the searching informative quantitative parameters of EMG signals for the differential diagnostics of the condition of the neuromuscular system. The method of discriminant analysis has been used for selection of diagnostically significant statistical parameters.

Tab. 2. Fig. 2. Ref.: 6 items.

УДК 004.932.72:004.931.4



В.В. Авраменко¹, Р.С. Волков²

¹СумГУ, г. Сумы, Украина, avr@sumdu.edu.ua

²СумГУ, г. Сумы, Украина, roman-email@yandex.ru

РАСПОЗНАВАНИЕ КОНТУРНЫХ ИЗОБРАЖЕНИЙ И ИХ ФРАГМЕНТОВ ПРИ ПЕРЕСЕЧЕНИИ

В статье описан метод локального распознавания фрагментов эталонных контуров на анализируемом изображении. Такие фрагменты, в частности, образуются при пересечении нескольких контурных изображений. Данный метод является инвариантным к масштабу, углу поворота и размещению этих фрагментов на распознаваемом изображении.

КОНТУРНОЕ ИЗОБРАЖЕНИЕ, ФРАГМЕНТ, ФУНКЦИИ НЕПРОПОРЦИОНАЛЬНОСТЕЙ,
ПОЛЯРНАЯ СИСТЕМА КООРДИНАТ, РАДИУС-ВЕКТОР

Введение

Распознавание контурных изображений является одной из основных задач при создании систем искусственного зрения. Обычно в системе распознавания имеется конечное множество эталонных контуров, которые она должна выявить на входном изображении. Однако на практике анализируемое изображение часто содержит только лишь фрагменты этих контуров, как, например, в случае пересечения нескольких контурных изображений. Данное обстоятельство значительно усложняет задачу их распознавания.

Существуют различные подходы и методы распознавания изображений [1]. Для решения рассматриваемой задачи необходим метод локального распознавания. Это вызвано тем обстоятельством, что довольно сложно применять интегральные оценки, когда эталонный контур на анализируемом изображении представлен не полностью. К тому же, для анализа может быть доступно не всё изображение, а лишь его фрагмент. Это обстоятельство также усложняет нормализацию изображения.

В [2] описан метод локального распознавания фрагментов эталонных контуров на выпуклом анализируемом контуре. Эти фрагменты могут иметь произвольный масштаб, угол поворота и положение.

В данной работе предлагается к рассмотрению метод, который решает более общую, нежели в [2], задачу, а именно: распознавание контурных изображений произвольной формы, а также фрагментов, которые возникают при их пересечении.

Система распознавания, реализующая данный метод, представляется как система поддержки принятия решений оператором.

1. Постановка задачи

Дано конечное множество эталонных изображений и анализируемое контурное изображение. Оно состоит из фрагментов эталонных контуров. Эти фрагменты в анализируемом изображении могут отличаться от эталонных масштабом, положением

и углом поворота. Кроме того, они могут пересекаться.

Допускается, что в общем случае для анализа может быть доступным не всё изображение, а лишь его фрагмент, а также то, что эталонные контуры могут иметь одинаковые между собой фрагменты.

Для каждой точки анализируемого контура необходимо определить, принадлежит ли она фрагменту какого-либо из эталонных изображений.

2. Описание метода

В соответствии с постановкой задачи метод должен быть инвариантным к углу поворота фрагмента эталона на анализируемом изображении. Для этого контурное изображение предлагается описывать в полярной системе координат. Это позволяет описать контур как зависимость длины радиус-вектора, который соединяет начало координат с точкой на контуре, от угла θ наклона касательной в точке начала координат, как показано на рис. 1 (для удобства восприятия на всех приведенных рисунках цифрами возле контурного изображения и графика функции его параметрического описания обозначены соответствующие точки).

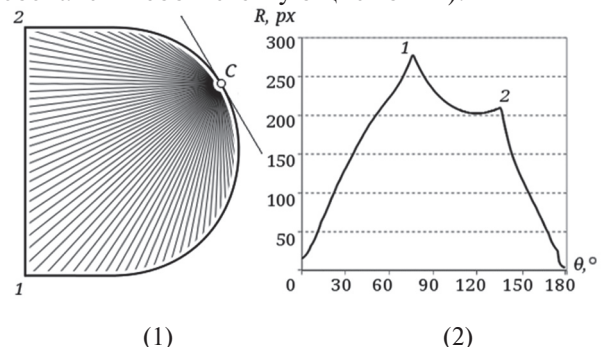


Рис. 1. Пример выпуклого анализируемого контура (1) и его представления в полярной системе координат (2)

Как видно из рис. 1, функция зависимости длины радиус-вектора от угла θ его наклона к касательной в точке начала координат задана однозначно. Однако в общем случае эта функция может быть неоднозначной, как показано на рис. 2 и рис. 3.

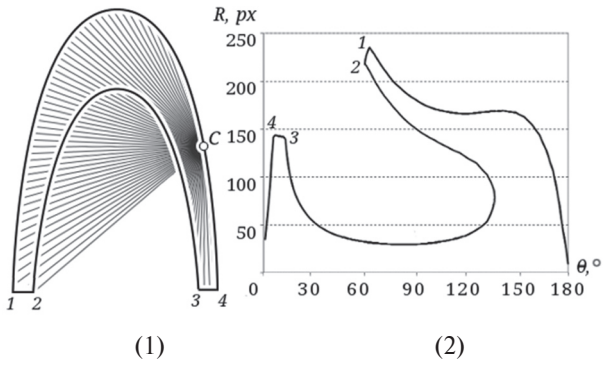


Рис. 2. Пример невыпуклого контурного изображения

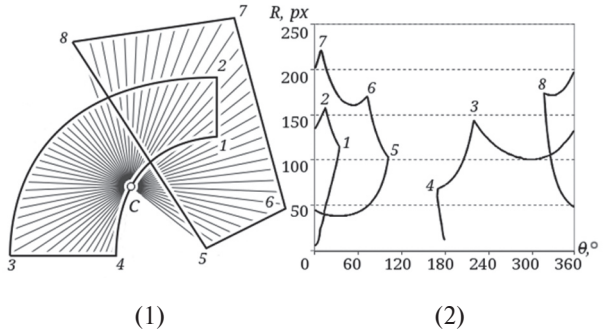


Рис. 3. Пример пересекающихся контурных изображений

Для устранения неоднозначности предлагается всё множество радиус-векторов разбить на подмножества в зависимости от количества пересечённых ими линий контура. Каждому подмножеству поставить в соответствие свою функцию зависимости длины радиус-вектора от угла наклона к касательной.

Количество таких функций будет равно максимальному числу пересечённых радиус-векторами линий контура. В результате того, что параметрическое описание контура может состоять не из одной, а одновременно из нескольких функций, каждая из них должна рассматриваться отдельно и независимо от других. Однако в процессе принятия решения о принадлежности фрагмента анализируемого контура какому-либо из эталонов, необходимо учитывать результаты распознавания по каждой из них.

Следует отметить, что инвариантность к углу поворота в [2] достигается благодаря учёту угла наклона касательной в точке начала полярной системы координат. Касательная, проведенная к любой точке контура, делит плоскость изображения на две полуплоскости. В случае, когда рассматривается выпуклый контур, он будет целиком находиться в одной из этих полуплоскостей. Вследствие этого диапазон рассматриваемых значений параметра θ лежит в пределах $0 \leq \theta \leq \pi$. Однако в случае, когда анализируются контуры произвольной формы или их пересечения, они могут находиться по обе стороны от касательной. Вследствие этого диапазон рассматриваемых значений параметра θ должен

быть $0 \leq \theta \leq 2\pi$. При этом необходимо рассматривать уже два значения угла поворота эталона относительно касательной к своему фрагменту на изображении: $\theta = 0$ и $\theta = \pi$, как показано на рис. 4.

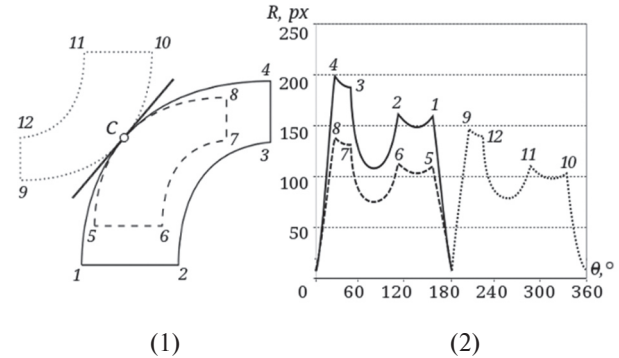


Рис. 4. Возможные положения эталонного контура относительно начала полярной системы координат и касательной в этой точке

Как видно из рис. 4, в случае, когда начала координат распознаваемого изображения и эталонного находятся в соответствующих точках эталона и его фрагмента на изображении, а с помощью касательных определён угол их поворота относительно друг друга, параметрические описания соответствующих фрагментов изображения и эталона отличаются на заранее неизвестный масштабный множитель. В этом случае имеет место уравнение вида:

$$R_{nq}(x_{nj}, y_{nj}, \theta) = k \cdot R_d(x_i, y_i, \theta), \quad (1)$$

где: $R_{nq}(x_{nj}, y_{nj}, \theta)$ – q -ая функция зависимости длины радиус-вектора с началом координат в j -ой точке (x_{nj}, y_{nj}) n -ого эталонного контура от угла наклона θ ; $R_d(x_i, y_i, \theta)$ – d -ая функция зависимости длины радиус-вектора с началом координат в i -ой точке (x_i, y_i) распознаваемого контура от угла наклона θ ; i – порядковый номер точки в анализируемом контуре, $1 \leq i \leq N$, N – количество точек анализируемого контура; j – порядковый номер точки в n -ом эталонном контуре, $1 \leq j \leq M_n$, M_n – количество точек в n -ом эталонном контуре; n – порядковый номер эталона в списке эталонов, $1 \leq n \leq G$, G – количество эталонов; d – порядковый номер функции параметрического описания анализируемого контура, $1 \leq d \leq D_i$, D_i – максимальное количество пересечённых радиус-вектором линий контура, при нахождении начала координат в точке (x_i, y_i) ; q – порядковый номер функции параметрического описания эталонного контура, $1 \leq q \leq Q_{nj}$, Q_{nj} – максимальное количество пересечённых радиус-вектором линий контура, при нахождении начала координат в точке (x_{nj}, y_{nj}) ; θ – угол наклона радиус-вектора к касательной в точке начала координат, $0 \leq \theta \leq 2\pi$; k – заранее неизвестный масштабный множитель.

В качестве меры, инвариантной к масштабному множителю, предлагается использовать функцию непропорциональности по производной первого

порядка для функций заданных параметрически [3,4]. Она равна нулю, если две функции пропорциональны, независимо от значения коэффициента пропорциональности.

$$\textcircled{d} R_{nq}^{(1)} R_d = \frac{R_d(x_i, y_i, \theta)}{R_{nq}(x_{nj}, y_{nj}, \theta)} - \frac{d R_d(x_i, y_i, \theta)/d\theta}{d R_{nq}(x_{nj}, y_{nj}, \theta)/d\theta}, \quad (2)$$

где символ \textcircled{d} – обозначение операции определения непропорциональности, d – от английского derivative (производная). Целиком $\textcircled{d} R_{nq}^{(1)} R_d$ – означает операцию взятия непропорциональности по производной первого порядка функции $R_d(x_i, y_i, \theta)$ по функции $R_{nq}(x_{nj}, y_{nj}, \theta)$, которые зависят от параметра θ .

В случае, если имеет место пропорциональная связь (1), что говорит о наличии в анализируемом изображении фрагмента рассматриваемого эталона, выражение (2) равно нулю при соответствующих значениях параметра θ .

Для i -ой точки (x_i, y_i) анализируемого контура соответствующая точка (x_{nj}, y_{nj}) на эталоне заранее неизвестна, поэтому производится последовательное перемещение начала полярной системы координат по контуру эталона.

3. Алгоритм распознавания

Входными данными для системы распознавания являются анализируемое и эталонные контурные изображения. Для их параметрического описания необходимо определить точки пересечения радиус-вектора, проведенного из начала координат, с линиями этих контуров.

Предлагается для увеличения быстродействия системы и точности нахождения точек пересечения все контурные изображения вначале преобразовывать из растрового формата в векторный, например с помощью линейной аппроксимации. Такое представление контурных изображений в то же время требует и меньший объем памяти для хранения.

Начало координат анализируемого изображения помещается в требуемую точку на контуре, для того чтобы установить её принадлежность какому-либо из эталонов. В свою очередь, начало полярной системы координат эталонного изображения последовательно перемещается по контуру, для того чтобы определить соответствующую пару точек эталонного и распознаваемого контуров. Вследствие того, что для каждой точки анализируемого контура необходимо каждый раз совершать обход всего эталонного контура и при этом в каждой точке нахождения начала координат также заново строить параметрическое описание контура, предлагается в начале процесса распознавания совершить некоторую подготовительную работу. Она заключается в том, что при первом проходе начала полярной системы координат по контуру

эталона в каждой точке его нахождения строится параметрическое описание контура и сохраняется в памяти. При последующих обходах эталонного контура вычисления, связанные с построением его параметрического описания, производиться не будут, а необходимые данные будут браться непосредственно из памяти.

Поскольку, в общем случае, эталонные контуры могут иметь одинаковые фрагменты, не исключается ситуация, когда одной и той же точке распознаваемого контура может быть поставлено в соответствие сразу несколько эталонов. В таком случае, требуется дополнительный критерий для получения оценки принадлежности, например по максимальному количеству значений параметра θ , в которых выражение (2) равно нулю, или по максимальной длине последовательности таких значений.

В результате, алгоритм распознавания представляет собой следующую последовательность действий:

1. Считывание всех эталонных изображений.
2. Аппроксимация линий контуров с помощью линейной интерполяции либо интерполяцией сплайнами [5].
3. Создание массива функций $R_{nq}(x_{nj}, y_{nj}, \theta)$ – параметрического описания эталонных контуров, где n – порядковый номер эталона, q – количество функций, описывающих контур, когда начало полярной системы координат помещено в точку (x_{nj}, y_{nj}) . Для создания этого массива выполняются следующие действия:
 - а) выбирается первый эталонный контур, $n = 1$;
 - б) начало полярной системы координат устанавливается в первую точку контура, $j = 1$;
 - в) начало координат соединяется радиус-векторами с каждой точкой контура;
 - г) производится подсчет количества пересеченных линий контура каждым радиус-вектором;
 - д) весь массив радиус-векторов разбивается на Q_{nj} подмассивов, Q_{nj} – максимальное число пересеченных линий контура каким-либо из радиус-векторов;
 - е) вычисляются длины всех радиус-векторов;
 - ж) подмассивы длин радиус-векторов сортируются в порядке возрастания значений параметра θ – угла наклона радиус-вектора к касательной в точке начала полярной системы координат;
 - з) начало полярной системы координат устанавливается в следующую точку контура, $j = j + 1$ и если $j \leq M_n$, то необходимо возвратиться к п. 3 в), иначе – к п. 3 и);
 - и) выбирается следующий эталонный контур, $n = n + 1$. Если выполняется условие $1 \leq n \leq G$, то переход к п. 3 б), иначе – к п. 4.
4. Считывание распознаваемого изображения. Допускается случай, когда для анализа доступно не всё изображение, а только его фрагмент.

5. Создание массивов функций $R_d(x_i, y_i, \theta)$ – параметрического описания анализируемого контура. Для этого выполняются действия, приведенные п. 2 и п. 3, но только как для одного эталона.

6. Выбор первой точки на распознаваемом контуре, $i = 1$.

7. Выбор первого эталонного контура, $n = 1$.

8. Выбор первой точки на эталонном контуре, $j = 1$.

9. Выбор первой функции из $R_d(x_i, y_i, \theta)$, $d = 1$.

10. Выбор первой функции из $R_{nq}(x_{nj}, y_{nj}, \theta)$, $q = 1$.

11. Вычисление непропорциональности (2) функции $R_d(x_i, y_i, \theta)$ по $R_{nq}(x_{nj}, y_{nj}, \theta)$.

12. Вычисление непропорциональности (2) функции $R_d(x_i, y_i, \theta)$ по $R_{nq}(x_{nj}, y_{nj}, \theta)$ со сдвигом $\Delta\theta = \pi$.

13. Если в п. 11 или п. 12 функция (2) принимает значения, равные нулю, то анализируемая точка распознаваемого контура (x_i, y_i) является фрагментом n -го эталонного контура и производится переход к п. 17, иначе – переход к п. 14.

14. Выбор следующей функции из $R_{nq}(x_{nj}, y_{nj}, \theta)$, $q = q + 1$. Если выполняется условие $q \leq Q_{nj}$, то переход к п. 11, иначе – к п. 15.

15. Выбор следующей функции из $R_d(x_i, y_i, \theta)$, $d = d + 1$. Если выполняется условие $d \leq D_i$, то переход к п. 11, иначе – к п. 16.

16. Выбор следующей точки на эталонном контуре, $j = j + 1$. Если выполняется условие $j \leq M_n$, то переход к п. 9, иначе – к п. 17.

17. Выбор следующего эталонного контура, $n = n + 1$. Если выполняется условие $n \leq G$, то переход к п. 8, иначе – к п. 18.

18. Выбор следующей точки на распознаваемом контуре, $i = i + 1$. Если выполняется условие $i \leq N$, то переход к п. 7, иначе – к п. 19.

19. Сохранение и вывод результатов распознавания.

4. Пример распознавания

Для иллюстрации работы метода на рис. 5 представлено распознаваемое изображение и эталонный контур. Необходимо определить, какие из линий анализируемого контура являются фрагментами эталона. Для наглядности рассматривается случай, когда начало полярной системы координат на распознаваемом изображении и на эталонном находятся в соответствующих точках (точки C и C' на рис. 5).

Как видно из рис. 5 б, представление распознаваемого контура в полярной системе координат задано неоднозначно. Для исключения этой неоднозначности анализируемый контур представлен одновременно двумя функциями зависимости

длины радиус-вектора от угла наклона к касательной, рис. 5 в и г.

Результаты вычисления непропорциональности функций параметрического описания распознаваемого контура и эталона представлены на рис. 5 ж и рис. 5 з. Из этих графиков видно, что фрагмент эталона на первой функции параметрического описания контура находится в диапазоне $0 \leq \theta \leq 6$ и $63 \leq \theta \leq 180$, а на втором – соответственно $7 \leq \theta \leq 62$. Этим значениям параметра θ соответствуют фрагменты изображения, представленные на рис. 5 и и рис. 5 к.

При принятии решения о принадлежности фрагментов изображения рассматриваемому эталону берутся во внимание результаты распознавания по каждой из функций параметрического описания контура, рис. 5 л.

Следует также отметить, что в точках перегиба функции параметрического описания контура функция непропорциональности (2) не вычисляется, т.к. в этих точках отсутствует первая производная.

Согласно описанному алгоритму точка C на контуре рис. 5 а, в которую помещено начало полярной системы координат, считается фрагментом рассматриваемого эталона, рис. 5 в. Далее начало координат переносится в другую точку распознаваемого контура, после чего принимается решение о принадлежности уже следующей точки рассматриваемому эталону, и т.д. Процесс распознавания заканчивается после рассмотрения всех точек распознаваемого контура.

Приведенный алгоритм можно оптимизировать, а именно: принимать решение о принадлежности не только той точки, в которой находится начало полярной системы координат, а считать фрагментами эталона и те точки, которым соответствуют нулевые значения функции непропорциональности, как это можно видеть из рис. 5 ж – к). В то же время, предложенный ранее алгоритм более надежный, т.к. менее подвержен погрешностям при вычислениях.

Система поддержки принятия решений для исключения случайных совпадений может учитывать как общее число нулевых значений функции непропорциональности, так и длины цепочек таких значений. Например, точка, в которой находится начало координат, может быть фрагментом рассматриваемого эталона в случае, если функция непропорциональности принимает последовательно как минимум три нулевых значения.

Процесс распознавания фрагментов эталонов на анализируемом контуре может проходить одновременно для многих точек. Поэтому для решения задачи можно использовать системы параллельных вычислений, что позволит существенно увеличить быстродействие системы распознавания.

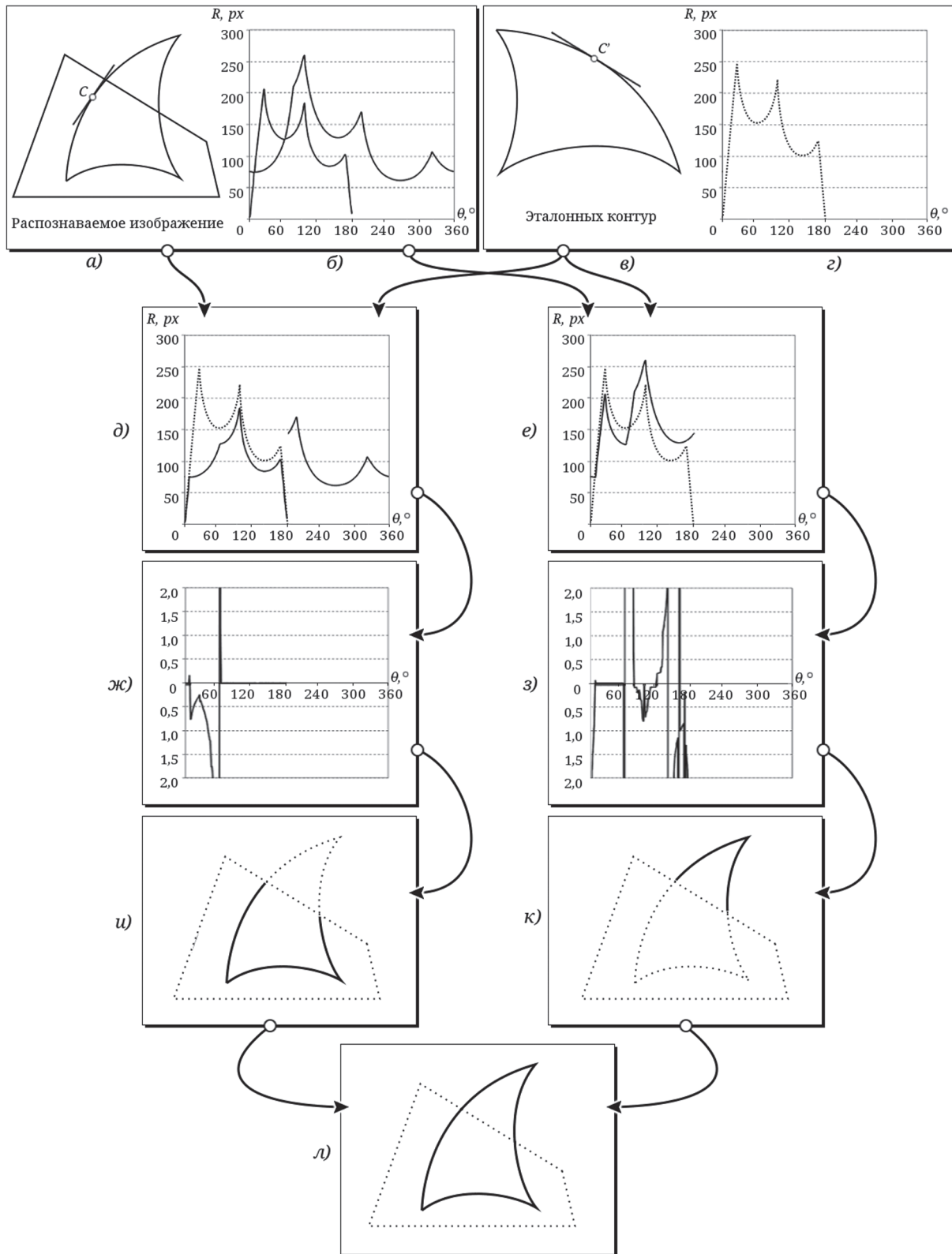


Рис. 5. Пример распознавания фрагментов эталонного контура (e) на анализируемом контуре (a) при нахождении начала полярной системы координат в соответствующих точках C и C'

Выводы

Предложен метод локального распознавания фрагментов эталонных контуров на анализируемом изображении, которые могут иметь произвольный угол поворота, положение и масштаб относительно друг друга, а также пересекаться. Допускается

также случай, когда для анализа доступно не всё изображение, а только его часть. Такие свойства метода достигаются благодаря:

- параметрическому описанию контуров в полярной системе координат как зависимости длины радиус-вектора от угла наклона к касательной

в точке начала координат. Радиус-вектор соединяет начало координат, которое помещается в точку на контуре, поочередно со всеми остальными точками этого контура;

– использованию функций непропорциональностей [3,4] в качестве меры, инвариантной к масштабному множителю.

Программная реализация предложенного алгоритма показала свою эффективность при распознавании фрагментов эталонов на анализируемом изображении, которые образуются при пересечении нескольких контурных изображений.

Список литературы: 1. *Путятин, Е.П.* Нормализация и распознавание изображений [Электронный ресурс] / Сумский гос. ун-т, летняя научно-практическая школа «Интеллектуальные системы». – Режим доступа: [www/URL: http://sumschool.sumdu.edu.ua/is-02/rus/lectures/putyatin/putyatin.htm/](http://sumschool.sumdu.edu.ua/is-02/rus/lectures/putyatin/putyatin.htm/) – 14.12.2011 г. – Загл. с экрана. 2. *Авраменко, В.В.* Распознавание фрагментов выпуклых контурных изображений [Текст] / В. В. Авраменко, Р. С. Волков // Вестник сумского гос. ун-та. – 2011. – № 2. – С. 7 – 12. 3. *Авраменко, В.В.* Характеристики непропорциональности числовых функций и их применение. – Деп. в ГНТБ Украины 19.01.98, № 59 – Ук 98. 4. *Авраменко, В.В.* Характеристики непропорциональности числовых функций и их применение при решении задач диагностики [Текст] / В. В. Авраменко // Вісник СумДУ. – 2000. № 16. – С.12 – 20. 5. *Авраменко, В.В.*

Аппроксимация контуров сплайнами в процессе их распознавания [Текст] / В. В. Авраменко, Р. С. Волков: тез. докл. научн.-техн. конф. ИММ-2011 / отв. ред. Т. В. Лютий. – Сумы : СумГУ, 2011. – С. 38.

Поступила в редколлегию 31.01.2012

УДК 004.932.72:004.931.4

Розпізнавання контурних зображень та їх фрагментів при перетині / В.В. Авраменко, Р.С. Волков // Біоніка інтелекту : наук.-техн. журнал. – 2012. – № 1 (78). – С. 72-77.

Запропоновано метод локального розпізнавання фрагментів еталонних контурів на зображенні, що аналізується. Такі фрагменти зазвичай утворюються при перетині декількох контурних зображень. Описаний у статті метод являється інваріантним до масштабу, куту повороту та розміщенню цих фрагментів на зображенні, що розпізнається.

Л. 5. Бібліогр.: 5 найм.

UDC 004.932.72:004.931.4

Crossing Contour Images and its Fragments Recognition / V.V. Avramenko, R.S. Volkov // Bionics of Intelligence: Sci. Mag. – 2012. – № 1 (78). – P. 72-77.

A novel method for recognition of contour fragments in an image being analyzed is proposed. Such fragments usually appear when contour images overlap each other. Using proposed method it is possible to recognize contour fragments which has arbitrary size, position and angle of rotation with respect to their corresponding fragments in etalons.

Fig. 5. Ref.: 5 items.



А.А. Гришко, С.Г. Удовенко, Л.Э. Чалая

ХНУРЭ, г. Харьков, Украина, udovenko@kture.kharkov.ua

ГИБРИДНЫЕ МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ В СИСТЕМАХ УПРАВЛЕНИЯ ДИНАМИЧЕСКИМИ ОБЪЕКТАМИ

В работе проведен анализ проблемы машинного обучения применительно к цифровым системам управления динамическими объектами. Предложен и протестирован модифицированный метод формирования стратегий управления, основанный на использовании Q-обучения с подкреплением. Новый подход позволяет пользователю системы управления непрерывно получать, анализировать и использовать сигналы подкрепления для выработки управляющих воздействий с применением нейросетевой модели.

СИСТЕМА УПРАВЛЕНИЯ, МАШИННОЕ ОБУЧЕНИЕ, НЕЙРОННАЯ СЕТЬ, Q-ФУНКЦИЯ, СТРАТЕГИЯ

Введение

В последнее время получили распространение управляемые стохастические системы, в основе функционирования которых лежит метод обучения с подкреплением, также называемый методом подкрепляемого обучения. Метод подкрепляемого обучения является достаточно новым методом в группе методов машинного обучения и занимает промежуточное положение между методами обучения с учителем и без учителя. В основе метода обучения с подкреплением лежат те основополагающие принципы адаптивного поведения, которые позволяют живым организмам приспосабливаться к изменяющимся или неизвестным условиям обитания. Метод обучения с подкреплением (Reinforcement Learning) был представлен и подробно изложен в [1]. В данном методе в обобщенном виде рассматривается взаимодействие агента с внешней средой, в результате которого агент путем проб и ошибок самостоятельно определяет наиболее оптимальное поведение для достижения максимума некоторого критерия. Отличительной чертой метода обучения с подкреплением является наличие сигнала подкрепления, который получает агент в процессе взаимодействия с внешней средой и который является скалярной величиной, характеризующей, насколько удачно функционирует агент в данный момент времени. Целью функционирования агента является максимизация суммарного сигнала подкрепления, которое получит агент при взаимодействии с внешней средой. В исходном виде метод обучения с подкреплением предполагает конечное количество состояний внешней среды и возможных воздействий агента на внешнюю среду, а также взаимодействие агента с внешней средой в дискретные моменты времени. Обучение с подкреплением является методом, который позволяет находить оперативное решение, являющееся оптимальным в смысле получения максимального дохода в каждом из состояний. При этом он позволяет в процессе обучения допускать возможность кратковременных потерь, с тем,

чтобы впоследствии максимизировать суммарный доход на длительном интервале. Вследствие этого, обучение с подкреплением является методом, наиболее приспособленным для эффективной работы в системах, характеризующихся высоким уровнем изменения внешних и внутренних воздействий (например, в трейдинговых системах электронной биржевой торговли) [2]. В соответствии с методом обучения с подкреплением сигналы подкрепления и состояния внешней среды должны обладать свойством марковости. Однако в [1] показано, что метод может быть успешно применен и в том случае, когда сигналы подкрепления и состояния внешней среды не обладают свойством марковости. Представляется целесообразным рассмотреть возможность применения обучения с подкреплением для управления динамическими объектами и предложить эффективный подход к реализации задачи такого управления.

1. Формализация задачи

В задаче обучения с подкреплением для агента управляемой системы используются следующие элементы:

- набор состояний системы S , возникающих при получении агентом информации из внешней среды;
- набор возможных управляющих воздействий A ;
- функция выигрыша R_t , получаемого в состоянии s_t при выборе в текущем такте управляющего воздействия a_t .

Задача базового алгоритма Q-обучения с подкреплением – определить и реализовать стратегию $\pi: S \rightarrow A$, основанную на текущем состоянии s_t , т.е. $\pi(s_t) = a_t$. Обычно требуется найти стратегию, соответствующую максимальному значению длительной суммы сигналов подкрепления.

Долгосрочный выигрыш R_t может задаваться различными типами функций в зависимости от поставленных задач. Если задача состоит в выработке фиксированной стратегии на протяжении

заданного числа шагов, то выигрыш можно определять как сумму прогнозируемых сигналов подкреплений:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \quad (1)$$

где $\gamma \in [0,1]$ – весовой коэффициент, задающий значимость будущих подкреплений.

В большинстве известных алгоритмов обучения с подкреплением используются обобщенные функции выигрыша V^π , которые позволяют для заданной стратегии π оценить средний выигрыш для одного состояния:

$$V^\pi(s) = E_\pi(R_t | s_t = s). \quad (2)$$

Эти функции могут также определяться не только состояниями, но и действиями агента в этих состояниях:

$$Q^\pi(s, a) = E_\pi(R_t | s_t = s, a_t = a). \quad (3)$$

Обобщенная функция (2) для состояния связана с $Q^\pi(s, a)$ соотношением вида:

$$V^\pi(s) = \sum_a \pi(s, a) Q^\pi(s, a). \quad (4)$$

Особенности функции (4) позволяют создавать различные модификации алгоритмов обучения с подкреплением, основанные на рекуррентном уравнении Беллмана следующего вида:

$$V^\pi(s) = \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [\mathfrak{R}_{ss'}^a + \gamma V^\pi(s')], \quad (5)$$

где $P_{ss'}^a$ – функция перехода, определяющая вероятность перехода системы в состояние s' при применении агентом в состоянии s действия a ; $\mathfrak{R}_{ss'}^a$ – соответствующая усредненная функция выигрыша.

Это уравнение, связывающее выигрыши для одного состояния с выигрышами для последующих состояний, можно получить следующим образом:

$$\begin{aligned} V^\pi(s) &= E_\pi(R_t | s_t = s) = \\ &= E_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s\right) = \\ &= E_\pi\left(r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \middle| s_t = s\right) = \\ &= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [\mathfrak{R}_{ss'}^a + \gamma E_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \middle| s_{t+1} = s'\right)] = \\ &= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [\mathfrak{R}_{ss'}^a + \gamma V^\pi(s')]. \end{aligned}$$

Функция (5) позволяет оценивать качество стратегии по выигрышам в будущих состояниях системы и, соответственно, сравнивать различные стратегии:

$$\pi \geq \pi' \Leftrightarrow \forall s, V^\pi(s) \geq V^{\pi'}(s). \quad (6)$$

В соответствии с неравенством (6) зададим функцию оценивания оптимальности стратегии,

которая позволила бы реализовывать обучение с подкреплением обучения с подкреплением:

$$V^*(s) = \max_\pi V^\pi(s). \quad (7)$$

Каждой паре «состояние-действие» с учетом (7) поставим в соответствие функцию вида:

$$Q^*(s, a) = \max_\pi Q^\pi(s, a), \quad (8)$$

определяемую последующей рекурсией:

$$Q^*(s, a) = E(r_{t+1} + \gamma V^*(s_{t+1}) | s_t = a, a_t = a). \quad (9)$$

С учетом уравнений (5) и (9) можно определить уравнения оптимальности стратегий для функций V^* и Q^* соответственно:

$$Q^*(s, a) = E(r_{t+1} + \gamma V^*(s_{t+1}) | s_t = a, a_t = a); \quad (10)$$

$$\begin{aligned} V^*(s) &= \max_a E(r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a) \\ &= \max_a \sum_{s'} P_{ss'}^a [\mathfrak{R}_{ss'}^a + \gamma V^*(s')] \end{aligned} \quad (11)$$

Очевидно, что применение стратегий в соответствии с (10) и (11) позволит агенту выбирать действия, максимизирующие обобщенный выигрыш.

Если динамику управляемой системы можно описать конечномерным управляемым марковским процессом, то уравнения (10) и (11) дают единственное решение, которое позволяет определить для каждого состояния максимальный возможный выигрыш. По известным значениям V^* нетрудно найти оптимальную стратегию, которая ставит в соответствие каждому состоянию действие a' , максимизирующее уравнения оптимальности стратегий для функций V^* и Q^* соответственно:

$$a' = \arg \max_a E(r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a); \quad (12)$$

$$a' = \arg \max_a Q^*(s, a). \quad (13)$$

Следует отметить, что для определения стратегий по V^* необходимо иметь модель среды, чтобы оценивать r_{t+1} и s_{t+1} . В случае применения функции Q^* наличие такой модели не является необходимым.

Таким образом, проблема обучения с подкреплением может быть сведена к выбору оптимальных стратегий по оценкам V^* и Q^* . Отметим, что для известных $P_{ss'}^a$ и $\mathfrak{R}_{ss'}^a$ можно непосредственно определять V^* и Q^* , решая систему уравнений оптимальности стратегий для каждого состояния методами динамического программирования. Однако такое решение трудно применить к системам управления динамическими объектами (в частности, роботами), так как зачастую полная модель управляемой системы (среды) является неизвестной, а ее большая размерность не позволяет агенту использовать метод проб и ошибок. Более приемлемыми для решения задач оценивания рассмотренных функций

выигрыша и последующего улучшения стратегия являются методы Монте-Карло, не требующие обязательного наличия такой модели. Базовый метод Монте-Карло основан на независимой оценке выигрыша для каждого состояния и позволяет выделить наиболее важные сегменты управляемого пространства, в то время как использование динамического программирования предполагает необходимость одновременного оценивания всех состояний. Этот метод применим для функции $Q(s,a)$, не требующей наличия полной модели среды и формирующейся в процессе получения сигналов подкрепления (текущих оценок выигрыша) после реализации действий a в состоянии s . При этом предполагается необходимость ограничения бесконечного в общем случае количества пар (s,a) , что является серьезной проблемой, так как снижает качество получаемых решений. Решить эту проблему можно добавлением к процедуре определения стратегии специальной процедуры, гарантирующей возможность реализации всех допустимых действий с некоторой вероятностью. Существует два метода задания такой процедуры. Первый состоит в условном объединении всех стратегий с низкой вероятностью, близкой к некоторой величине ϵ , и их сопоставлении всем возможным действиям. Процедура обучения сходится при этом к оптимальной стратегии в окрестности этого объединенного класса стратегий. Это позволяет гарантировать субоптимальную работу системы, а значение вероятности ϵ может быть уменьшено, если в процессе работы системы агент получит достаточные для оценивания стратегий данные. Такой метод называют методом «on policy» (в пределах стратегии), так как он лишь модифицирует при необходимости стратегию, используемую агентом. Второй метод называют методом «off policy» (вне стратегии), так как он оценивает одну стратегию, в то время как агент использует другую. Эта другая стратегия выбирает, как правило, оригинальную стратегию с вероятностью $1 - \epsilon$ и случайное действие с вероятностью ϵ . Для оценивания оригинальной стратегии метод Монте-Карло использует конечные фрагменты последовательностей, для которых выбор действий соответствует выбору, применяемому для оригинальной стратегии, но модифицирует взвешивание выигрышей для компенсации разности вероятностей выбора действий по двум стратегиям. Обучение, использующее метод Монте-Карло, осуществляется, таким образом, на основе чередования оценок одной стратегии и ее модификации, вырабатывая действия, которые максимизируют выигрыш для каждого состояния. Такое чередование может быть реализовано различными способами, как и для динамического программирования. Как динамическое программирование, так и метод Монте-Карло имеют свои преимущества и недостатки. В частности, динамическое

программирование обладает интересным свойством оценивания каждого из состояний, именуемым «загрузчиком». Это свойство позволяет достичь более высокой сходимости (в единицах сочетаний «состояние-действие-сигнал подкрепления») по сравнению с методом Монте-Карло.

Рассмотрим теперь процесс обучения по методу временных разностей (ТД), объединяющему в определенном смысле преимущества двух описанных выше методов и являющемуся наиболее приемлемым для практического применения в системах управления динамическими объектами. Оценивание стратегии ТД – метода осуществим, как и для метода Монте-Карло, использующего усредненные оценки для одного состояния. Представление такого оценивания в итеративной форме приводит к следующему уравнению:

$$V(s_t) \leftarrow V(s_t) + \alpha [R_t - V(s_t)], \quad (14)$$

где

$$R_t = r_{t+1} + \gamma r_{t+2} + \dots + \gamma^p r_{t+p+1}. \quad (15)$$

Уравнения (14) и (15) отражают идею использования оценки выигрыша, начиная со следующего сигнала подкрепления и следующего значения состояния, вместо использования сигналов подкрепления экспериментальной последовательности. ТД – метод использует при этом следующее преобразование:

$$V(s_t) \leftarrow V(s_t) + \alpha [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]. \quad (16)$$

Как правило, применяется инкрементная форма преобразования (16), учитывающая действия агента. Такая форма для функции Q имеет следующий вид:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]. \quad (17)$$

Метод, основанный на преобразовании (17), называют иногда SARSA-методом (State, Action, Reward, State, Action – состояние, действие, выигрыш, состояние, действие). Он реализует оценивание «on policy», так как используемое здесь действие a_{t+1} зависит от выбираемой стратегии.

Наиболее распространенный метод обучения с подкреплением – метод Q-обучения, являющийся вариантом «off policy» SARSA-метода, в котором используется максимум функции от прогнозируемых действий вместо реализуемого в текущий момент действия:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]. \quad (18)$$

Этот алгоритм осуществляет сходимость Q к Q^* независимо от реализуемой стратегии, причем эта стратегия гарантирует приемлемые результаты (если вероятности отличны от нуля для всех действий во всех состояниях).

2. Гибридный метод машинного обучения

Методы, использующие временные разности, позволяют заменить конечные значения используемых в алгоритме оценок оценками, полученными по методу Монте-Карло для следующего состояния. При этом в процессе обучения такие прогнозируемые оценки могут оказаться недостаточно корректными и тогда их необходимо заменить другими значениями оценок.

При применении TD – метода n раз получим выигрыш, определяемый следующим уравнением:

$$R_t = R_t^n = r_{t+1} + \gamma r_{t+2} + \dots + \gamma^n (r_{t+n+1} + \gamma V(s_{t+n+1})). \quad (19)$$

Использование оценки (19) позволяет получить процедуру вида:

$$V(s_t) \leftarrow V(s_t) + \alpha [R_t^n - V(s_t)]. \quad (20)$$

Введем в (20) усредненные взвешенные оценки R_t :

$$R_t = \sum_i a_i R_t^i; \quad \sum a_i = 1. \quad (21)$$

Частным случаем взвешенных оценок здесь являются экспоненциально взвешенные оценки, которые приводят к повышению значимости будущих результатов и их измерений по мере возрастания аргумента времени:

$$R_t = (1 - \lambda) \sum_{i=1}^{\infty} \lambda^i R_t^i. \quad (22)$$

Это позволяет упростить реализацию TD – метода, используя предыдущие значения сигналов подкрепления вместо оценок будущих значений. Введем для этого вспомогательный параметр (назовем его коэффициентом приемлемости), определяемый по следующей рекурсии:

$$e_t(s) = \begin{cases} \gamma \lambda e_{t-1}(s), & s \neq s_t \\ \gamma \lambda e_{t-1}(s) + 1, & s = s_t. \end{cases} \quad (23)$$

Перерасчет обобщенных функций выигрыша будет теперь производиться для каждого состояния пропорционально значению его коэффициента приемлемости:

$$V(s_t) \leftarrow V(s_t) + \alpha e_t(s_t) [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]. \quad (24)$$

Суть предложенного подхода состоит в замене состояний, использующих прогнозируемые значения сигналов подкрепления предыдущими взвешенными состояниями, использующими текущие значения сигналов подкрепления. Соответствующий алгоритм, который является расширением SARSA-алгоритма и $Q(\lambda)$ -алгоритма, может быть представлен следующей последовательностью вычислительных операций:

1. Инициализировать $V(s)$ и принять $e(s) = 0$ для всех s .
2. **for all** итераций **do**
3. Инициализировать значение s .
4. **for all** шагов текущей итерации **do**

5. $a \leftarrow \pi(s)$.

6. Реализовать a , получить r и значение следующего состояния s' .

7. $\delta \leftarrow r + \gamma V(s') - V(s)$.

8. $e(s) \leftarrow e(s) + \delta$.

9. **for all** s **do**

10. $V(s) \leftarrow V(s) + \alpha \delta e(s)$.

11. $e(s) \leftarrow \gamma \lambda e(s)$.

12. **end for**

13. $s \leftarrow s'$.

14. **end for**

15. **end for**

3. Практическая реализация гибридного метода

В реальных системах робототехники и цифрового управления динамическими процессами датчики и исполнительные механизмы редко бывают дискретными или же, если они все-таки используются, число состояний должно быть очень большим. Описанные выше алгоритмы обучения с подкреплением (и, в частности, предложенный гибридный алгоритм) предполагают применение множеств состояний и управляющих воздействий такой размерности, чтобы соответствующие вычислительные процедуры могли сходиться в реальном масштабе времени. Для практической реализации обучения с подкреплением при управлении динамическими объектами можно использовать различные подходы. Наиболее простой из них заключается в обычной дискретизации непрерывного пространства состояний и выделении нескольких сотен наиболее вероятных состояний, которые впоследствии будут использоваться в упрощенных вариантах алгоритмов обучения (например, с непосредственным использованием таблиц значений $Q[s][a]$).

Второй подход предполагает возможность работы алгоритма обучения непосредственно в непрерывном пространстве состояний, фиксируемых датчиками, используя методы аппроксимации функций. В общем случае, для практической реализации алгоритмов обучения с подкреплением необходимо корректно оценивать значения функции $Q(s, a)$. Аппроксимация такой функции может быть успешно реализована с применением искусственных нейронных сетей (ИНС). Нейросетевые методы реализации алгоритмов обучения зачастую называют коннекционистскими. Один из вариантов ИНС типа «многослойный персептрон», предназначенных для аппроксимации функции $Q(s, a)$, приведен на рис. 1.

Сравнивая обычное использование персептрона для задачи аппроксимации и его использование в качестве составной части алгоритма с подкреплением, можно выделить два основных момента. Во-первых, в задачах обычной аппроксимации обучение производится на некотором обучающем множестве, элементы которого постоянно

повторяются. При обучении с подкреплением предварительно заданного обучающего множества нет, а входные образцы формируются при взаимодействии автономного агента со средой. Таким образом, в процессе обучения некоторые образцы встречаются чаще, другие реже, а при работе с непрерывной средой велика вероятность того, что входной образец встретится лишь однажды. Поэтому для задач обучения с подкреплением задача переобучения является неактуальной.

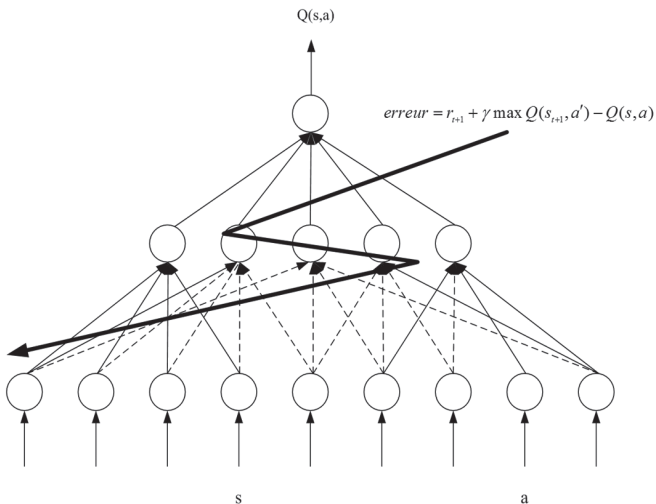


Рис. 1. Структура искусственной нейронной сети, аппроксимирующей функцию $Q(s, a)$

Во-вторых, в задачах обычной аппроксимации обучение производится на известных результатах, т.е. известны истинные значения аппроксимируемой функции в определенных точках. При обучении с подкреплением истинные значения аппроксимируемой функции заранее неизвестны, и обучение происходит на оценках Q-значений, которые постепенно изменяются в процессе обучения.

При использовании Q-обучения коррекция Q-функции производится на основе оценок текущего и последующего состояний, т.е. на оценках смежных состояний. Гибридный алгоритм, описанный выше, позволяет учитывать оценки состояний, удаленных на большее расстояние друг от друга. Формула для алгоритма TD(λ) может быть представлена как в инкрементной, так и в рекурсивной форме. При использовании рекурсивной формулы обновление Q-функции производится только после того, как агент достигнет поглощающего состояния (когда задача агента будет выполнена). При использовании инкрементной формулы обновления производятся после каждого шага агента. Данные методы ускорения обучения могут использоваться как для табличного представления Q-функции, так и при использовании нейросетевой аппроксимации.

Для обучения многослойного персептрона используется алгоритм обратного распространения ошибки. Следует отметить, что при использовании

такой архитектуры необходимо обновлять только веса, связанные с выходным узлом, действие которого было выбрано. Обновление производится после того, как агент совершил переход. Чтобы при обратном распространении ошибки не возникало конфликтных ситуаций с весами скрытого слоя, можно разделить единую нейронную сеть на несколько сетей с единственным выходом, каждая из которых относилась бы к отдельному действию. Конечно, такой подход может быть применен только при относительно малом количестве действий. При использовании множества сетей для вычисления оценок-значений на каждой итерации алгоритма на входы всех сетей подается состояние, а коррекция весов производится только для той сети, действие которой было выбрано.

Рассмотрим два примера, иллюстрирующих возможность применения гибридного метода обучения с подкреплением при управлении динамическими объектами.

Пример 1. Проведем анализ решения задачи управления мобильным роботом по гибридному алгоритму обучения с подкреплением на основе Q-таблиц и ИНС.

Перед роботом ставится задача – добраться до цели, избежав столкновения с препятствиями. Критерием оценки эффективности функционирования робота служит среднее значение выигрыша, полученного за время взаимодействия со средой. Информацию об окружающей среде робот получает при помощи 7 сенсоров: 1-5 сенсоры располагаются под углом 15° относительно друг друга и представляют информацию о расстоянии от робота до препятствия; 6-ой – информацию о расстоянии до цели; 7-ой – информацию об угле между направлением робота и целью. Схема дискретизации данных от сенсорных датчиков представлена на рис. 2. Для каждого сектора (от 0 до 6) заданы значения функции Θ . Значение Θ_i определяется наличием или отсутствием препятствий в зонах **a**, **b** и **c** (0, если препятствие в зоне **a**; 1, если в зоне **b** и 2 если в зоне **c**). Робот использует действия 3 видов: идти прямо, повернуть направо, повернуть налево. Текущий выигрыш (сигнал подкрепления) составляет +3, если робот выбрал действие продвижения вперед, и -10, если робот сталкивается с препятствиями. Состояние s определяется сочетанием 3 значений Θ_i : $\Theta_i : \Theta_i = \sum_k 3^k \Theta_k$.

В рассмотренном примере $\Theta_0 = 2$, $\Theta_1 = 2$, $\Theta_2 = 1$, $\Theta_3 = 1$, $\Theta_4 = 2$, $\Theta_5 = 2$, $\Theta_6 = 2$, следовательно, $s = 2150$.

При проведении экспериментов с классическим Q-обучением было установлено, что размер таблицы Q-значений по окончании обучения колеблется от 2800 до 3200 состояний. А это означает, что при использовании 2-х байт для хранения одного Q-значения потребуется 32000 байт для

всей таблицы. В то время как при использовании коннекционистского Q -обучения (по гибридно-му алгоритму) с 3 нейронами скрытого слоя и 2-мя байтами для каждого веса нейронной сети потребовалось 810 байт.

Пример 2. Базовый метод обучения с подкреплением предполагает конечное количество состояний внешней среды и возможных воздействий агента на внешнюю среду, а также взаимодействие агента с внешней средой в дискретные моменты времени. Эти ограничения не позволяют широко использовать метод обучения с подкреплением в задачах управления техническими системами, т. к. сигналы в системах управления могут быть как дискретными, так и непрерывными по уровню и во времени. В то же время известны примеры успешного применения обучения с подкреплением в технических системах (например, для управления тележкой с шестом и плавающим роботом) [3].

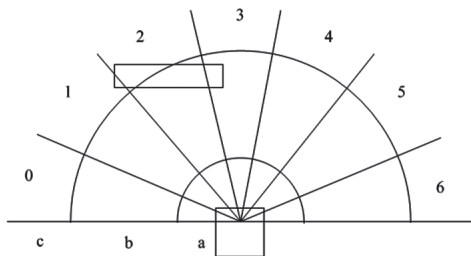


Рис. 2. Дискретизация пространства состояний мобильного робота

На основе метода подкрепляемого обучения в [4] предложена структурная схема обобщенной системы автоматического управления, функционирующей на основе метода обучения с подкреплением (МОП-САУ), используемая для моделирования SISO-систем, и алгоритмы работы структурных блоков. Приведенный выше гибридный алгоритм позволяет расширить возможности применения такой схемы. Входящий в состав модифицированной МОП-САУ объект управления (ОУ) должен удовлетворять следующим условиям: ОУ имеет один или несколько входов и один выход; в моменты дискретизации можно измерить сигналы, которые вместе с вектором управляющих воздействий u однозначно определяют значение выходной величины y в будущие моменты времени; транспортное запаздывание по каналам управления является незначительным. В результате обработки вектора входных сигналов адаптивный регулятор (АР) формирует векторное управляющее воздействие, значение которого является одним из элементов заранее определенного дискретного множества возможных воздействий. Под действием управляющего воздействия ОУ изменяет свое состояние. Вектор входных сигналов поступает на вход аналого-цифрового преобразователя (АЦП), который осуществляет дискретизацию по времени

входных сигналов, необходимую для реализации гибридного метода обучения с подкреплением, который предполагает взаимодействие агента с внешней средой в дискретные моменты времени. На выходе АЦП формируется вектор дискретных сигналов, который поступает на анализирующее устройство (АУ) и на квантователь. АУ определяет значение сигнала подкрепления r , а квантователь определяет значение сигнала состояния внешней среды s , которое является одним из элементов заранее определенного множества возможных состояний внешней среды. Экстраполятор (ЭК) переводит дискретный сигнал, сформированный блоком «Агент-М» как воздействие на внешнюю среду, в непрерывное по времени управляющее воздействие на ОУ. Наличие в векторе входных сигналов производной входного воздействия и вектора переменных состояния ОУ позволяет частично учесть то, что в соответствии с методом обучения с подкреплением сигналы подкрепления и состояния внешней среды должны обладать свойством марковости. Блок «Агент-М» функционирует на основе метода обучения с подкреплением. Целью функционирования этого блока является максимизация суммарной оценки выигрыша для управления. Блок «Агент-М» состоит из устройства управления объектом (УУО) и устройства управления адаптацией (УУА). УУО формирует воздействие на основе информации о текущем состоянии внешней среды с использованием функции оценки воздействия (Q -функции). УУА осуществляет коррекцию Q -функции на основе анализа текущего состояния внешней среды и значения сигнала подкрепления как результата воздействия на внешнюю среду на предыдущем такте. В модифицированной МОП-САУ Q -функция представляется в виде таблицы соответствия, то есть для каждого возможного состояния внешней среды и для каждого возможного воздействия выделяется ячейка памяти, в которой хранится значение функции для данных значений аргументов. Экспериментальные исследования дискретных систем с различными ОУ показали, что Q -функции являются гладкими и непрерывными, что позволяет использовать для их представления функциональные аппроксиматоры. Проблему экспоненциального роста объема требуемой памяти предлагается устранить за счет представления Q -функции на основе трехслойной искусственной нейронной сети (ИНС) прямого распространения. Так как для хранения значений параметров ИНС не требуется больших объемов памяти, их применение позволит решить указанную проблему. Кроме того, входные и выходные сигналы ИНС могут быть непрерывными, что позволяет перейти от ограниченного множества возможных состояний ОУ к непрерывному пространству состояний ОУ. Первый слой ИНС является входным и

содержит столько нейронов, сколько сигналов содержится в векторе входных дискретных сигналов. Третий слой состоит из одного нейрона с линейной активационной функцией. Количество нейронов в среднем слое выбирается в зависимости от количества нейронов во входном слое. Изменение Q -функции осуществляется методом обратного распространения ошибки. Рассмотренная МОП-САУ была использована для моделирования систем цифрового управления некоторыми техническими ОУ, в частности процессом управления газоперекачивающим агрегатом (ГПА). Математическая модель ГПА представляется в виде системы дифференциальных уравнений третьего порядка. С помощью дискретной Q -функции в математическом пакете Scilab была обучена соответствующая трехслойная ИНС. Результаты моделирования показали, что среднеквадратическое отклонение значений исходной и аппроксимирующей функций друг от друга не превышает 1,5%, что свидетельствует о возможности использования ИНС для представления Q -функций. Применение ИНС позволяет не только устранить экспоненциальную зависимость объема требуемой памяти от порядка ОУ, но также подтверждает возможность синтеза МОП-САУ для SISO и MISO-ОУ. Использование нейронных сетей в МОП-САУ затрудняется тем, что коррекция значения Q -функции на основе ИНС в одной точке приводит к изменению значений функции в других точках. Это связано с тем, что применение метода обратного распространения ошибки приводит к изменению параметров связей между нейронами, которые участвуют в формировании значений функции при любых значениях входных сигналов. С целью уменьшения влияния изменения значения Q -функции в одной точке на значения функции в других точках был применен следующий способ обучения ИНС: совместно с изменением значения Q -функции в этой точке осуществляется закрепление значений Q функции в нескольких точках из окрестности этой точки. Закрепление осуществляется за счет применения метода обратного распространения ошибки с нулевой ошибкой. Результаты экспериментов показали, что применение такого способа итерационного обучения ИНС позволяет уменьшить величину среднеквадратического отклонения значений функции в окрестности изменяемой точки от первоначальных значений.

Выводы

Результаты, полученные при моделировании систем управления роботом и газоперекачивающим агрегатом с применением предложенного гибридного метода определения управляющих стратегий,

подтверждают работоспособность и перспективность применения методов машинного обучения с подкреплением в технических системах. Повышение эффективности практической реализации принципов интеллектуального управления динамическими объектами с использованием рассмотренных в настоящей процедур, может быть достигнуто в результате проведения дальнейших исследований применения методов машинного обучения с подкреплением для различных классов задач идентификации и управления в условиях неопределенности. Представляется целесообразным дальнейшее развитие функциональных возможностей рассмотренной в статье системы МОП-САУ с целью ее эффективного применения для синтеза реальных систем управления и в учебном процессе.

Список литературы: 1. Sutton, R.S., Barto, A.G. Reinforcement learning: An introduction. / R.S. Sutton, A.G. Barto// Cambridge, MA: MIT Press. —1998. — 432 p. 2. Hryshko A. An Implementation of Genetic Algorithms as a Basis for a Trading System on the Foreign Exchange Market./ A.Hryshko, T. Downs// Proceedings of the 2003 Congress on Evolutionary Computation. — 2003. — P.1695-1701. 3. Filliat, David. Robotique Mobile. / D. Filliat// Paris: ENSTA. — 2001. — 175p. 4. Вичугов, В.Н. Нейросетевой метод подкрепляемого обучения в задачах автоматического управления [Текст] / В.Н. Вичугов // Известия Томского политехнического университета. — 2006. — Т. 309. № 7. — С. 92-96.

Поступила в редколлегию 11.11.2011

УДК 519.62

Гібридні методи машинного навчання в системах керування динамічними об'єктами / А.О. Гришко, С.Г. Удовенко, Л.Е. Чала // Біоніка інтелекту : наук.-техн. журнал. — 2012. — № 1 (78). — С. 78-84.

У статті розглядаються оптимальні стратегії у системах керування динамічними об'єктами, що використовують методи машинного навчання з підкріпленням. Запропонований підхід дозволяє отримати високу якість апроксимації функцій оцінювання оптимальності стратегій за допомогою багатопарових штучних нейронних мереж. Розглянуто приклади використання розроблених гібридних методів у задачах керування. Методи реалізовано програмно та протестовано.

Л. 2. Бібліогр.: 04 найм.

УДК 519.62

Hybrid machine learning methods in dynamic objects control systems / А.А. Hryshko, S.G. Udovenko, L.E. Chalaya. // Bionics of Intelligense: Sci. Mag. — 2012. — № 1 (78). — P. 78-84.

The paper considers the optimal strategies in control systems for dynamic objects using machine learning methods for reinforcement. The proposed approach allows to obtain high-quality approximation of the optimal strategies for evaluating functions by using multi-layer artificial neural networks. Examples of the use of hybrid methods developed in control. Methods implemented in software and tested.

Fig. 2. Ref.: 04 items.

УДК 519.87



В.А. Дяченко, О.Ф. Михаль

ХНУРЭ, г. Харьков, Украина, fuzzy16@pisem.net

АДАПТИВНОЕ ПАРАЛЛЕЛЬНОЕ ОБУЧЕНИЕ МОДИФИЦИРОВАННОЙ САМООРГАНИЗУЮЩЕЙСЯ КАРТЫ КОХОНЕНА

Рассмотрены и сопоставлены по параметрам классический и модифицированный варианты реализации обучения самоорганизующихся карт Кохонена. Проведен сравнительный анализ скорости обучения при различных типах входных данных. Полученные результаты интересны в аспекте выработки рекомендаций по распараллеливанию отдельных фаз обучения при реализации модифицированного варианта карты Кохонена на вычислительной системе, поддерживающей распределённую обработку данных.

РАСПРЕДЕЛЁННАЯ ОБРАБОТКА ДАННЫХ, САМООРГАНИЗУЮЩИЕСЯ КАРТЫ КОХОНЕНА

Введение

Задачи *распределённой обработки данных* (РОД) имеют ряд особенностей при реализации на небольших сетевых распределённых *вычислительных системах* (ВС), составленных из компьютеров общего назначения. В таких ВС процессы, которые могли бы реализовываться параллельно во времени, вынужденно частично исполняются последовательно, в меру наличия вычислительных ресурсов и с учётом сосредоточенности ресурсов по узлам ВС. Для подобных систем со смешанной «сосредоточенно-параллельной» (сосредоточенной в рамках локальной сетевой ВС) аппаратной РОД актуальным является нахождение эффективных сочетаний алгоритмических решений с конкретными структурно-аппаратными возможностями ВС. В особенности это представляет интерес в связи с использованием ВС общего назначения для моделирования самоорганизующихся и самонастраивающихся систем. В качестве объектов моделирования перспективны, в частности, *самоорганизующиеся карты Кохонена* (СКК), реализующие режим самообучения. Модифицированный вариант СКК [1, 2] предполагает распараллеливание отдельных фаз обучения [2–4]. В частности, распараллеливание может быть аппаратно реализовано на указанных ВС со смешанной «сосредоточенно-параллельной» РОД. На данный момент использование модифицированных СКК с поддержкой средствами ВС со смешанной «сосредоточенно-параллельной» РОД исследовано не в полной мере.

Целью данной работы является сравнение возможностей и оценка эффективности классического и модифицированного вариантов СКК в аспекте поддержки распараллеливания процедур обучения.

1. Основные представления

Радикальным способом повышения эффективности работы ВС является применение РОД, при которой предполагается декомпозиция вычислительного процесса, вычленение независимых (независимых) фрагментов и реализация соответствующих вычислений в виде независимых потоков на отдельных узлах имеющейся ВС. ВС

может быть, в частности, локально-сетевой либо аппаратно сосредоточенной, реализованной в виде компьютера общего назначения с многоядерным процессором. В указанных вариантах, а также в комбинациях подобных случаев, представляет интерес эффективное сочетание алгоритмических решений с конкретными структурно-аппаратными возможностями ВС. Наличие параллельных ветвей и независимых блоков выполняемого алгоритма должно приниматься в расчёт и реализовываться с учётом наличных возможностей и структурных особенностей РОД.

Широкий класс подобных задач связан с развитием самонастраивающихся и самоорганизующихся систем. В частности, перспективны самоорганизующиеся *нейросетевые структуры* (НС). Одним из достоинств аппарата НС является возможность получения обоснованного результата на основании данных, ранее не встречавшихся в процессе обучения. Это позволяет радикально сокращать объёмы обучающих выборок, а следовательно, сроки обучения системы. Количество вычислительных операций существенно возрастает с увеличением объёма обрабатываемой информации. Таким образом, всякое сокращение объёма обучающей выборки автоматически обеспечивает более экономное расходование системных ресурсов.

Одним из видов НС являются СКК. Их отличительной особенностью является автономное обучение (обучение по схеме «без учителя»). Внешне, с точки зрения пользователя, это выглядит как процесс, в котором сама СКК «вырабатывает понимания» структуры данных. Результатом работы СКК является разделение данных на кластеры, которые могут далее непосредственно распознаваться пользователем. Соответственно, при поступлении входного образа ранее не предъявлявшегося сети, СКК относит его к одному из имеющихся кластеров или определяет новый кластер.

Другим достоинством СКК является возможность визуализации многомерных данных – снижение размерности набора данных. Совокупность кластеров представима в пространстве с меньшей размерностью. Если размерность является

«антропоморфной» (соответствует практике обычных человеческих представлений), результаты работы СКК могут использоваться человеком непосредственно, без дополнительного обобщающего анализа. Таким образом, СКК удобны как интеллектуальный человеко-машинный интерфейс – усилитель определённых аспектов человеческого интеллекта.

Принципы работы СКК в классическом варианте реализации достаточно подробно изучены [5], как один из видов НС с обучением «без учителя». Ограничением классических СКК является «поштучная» обработка обучающих образов: поэлементное (повекторное) предъявление образов и коррекция сенсорного поля после обработки каждого из предъявлений. Модифицированные СКК [1-3] предполагают «пакетную» обработку обучающих образов. Образы обучающей выборки предъявляются последовательно, и если их воздействия на сенсорное поле не взаимоперекрываются, то результирующая коррекция сенсорного поля осуществляется после (и по результатам) предъявления всего пакета. При этом матрица результирующей коррекции строится так, что включает в себя все отдельные коррекции, соответствующие отдельным предъявленным образам, проявившимся на разных непересекающихся участках сенсорного поля. Достижимый выигрыш эффективности обеспечивается за счёт сокращения обработок сенсорного поля и обусловлен объёмом независимых (не-взаимовлияющих) обучающих образов в «пакете» (обучающей выборке).

Представляет интерес рассмотрение возможностей реализации классического и модифицированного вариантов СКК и сопоставление методов их обучения. В частности, сравнительный анализ скорости обучения СКК при различных типах входных данных может быть интересен в аспекте выработки рекомендаций по распараллеливанию отдельных фаз обучения СКК при реализации модифицированного варианта на ВС, которые поддерживают РОД.

2. Обучение классических самоорганизующихся карт Кохонена

Рассмотрим вариант СКК с двумерным сенсорным полем. Работа СКК заключается в следующем. Сеть состоит из $p \times q$ элементов (нейронов), которые расположены в узлах двумерной решетки – сенсорного поля. Каждый (i, j) -й, где: $i \in (1, 2, \dots, p)$, $j \in (1, 2, \dots, q)$ нейрон имеет вектор весовых коэффициентов из n элементов: $A_{ij}: \{a_1, a_2, a_3, \dots, a_s, \dots, a_n\}$. Из входного набора данных $B: \{B_1, B_2, \dots, B_k, \dots, B_m\}$ (обучающей последовательности из m образов) векторы того же размера n $B_k: \{b_1, b_2, b_3, \dots, b_s, \dots, b_n\}$ выбираются в случайном порядке и последовательно подаются на элементы сенсорного поля для сравнения. Каждый подаваемый обучающий

вектор параллельно (независимо) сопоставляется с каждым из векторов сенсорного поля $B_k \leftrightarrow A_{ij}$.

Для каждого B_k отыскивается весовой вектор сенсорного поля A_{ij} , евклидово расстояние до которого от входного вектора минимально: $\min \left(\sqrt{\sum_{s=1}^n (a_s - b_s)^2} \right)$. Найденный весовой вектор и весовые векторы нейронов-соседей сенсорного поля подстраиваются по правилу Кохонена, с учётом функции соседства (с убыванием по отстоянию от нейрона-победителя) и функции влияния (с убыванием по номеру обучающего образа) [5]. После того, как обучающий набор данных B предъявлен достаточное число раз, сенсорное поле прошло обучение на наборе B . При этом основная масса весовых векторов элементов сенсорного поля центрируется вокруг малого числа векторов – центров кластеризации.

За счёт того, что сенсорное поле изначально представляет собой малоразмерную (двумерную) решетку (A_{ij} – два индекса), совокупности нейронов СКК и значения их весовых векторов могут быть представлены, например, в виде специальных карт-раскрасок, содержащих цветные области. Подобное представление является «антропоморфным», то есть удобным для непосредственного человеческого восприятия. Таким образом, при этом проявляется снижение размерности исходного многомерного набора данных до уровня, удобного для визуализации.

Укрупнённая блок-схема алгоритма обучения классического варианта СКК представлена на рис. 1.

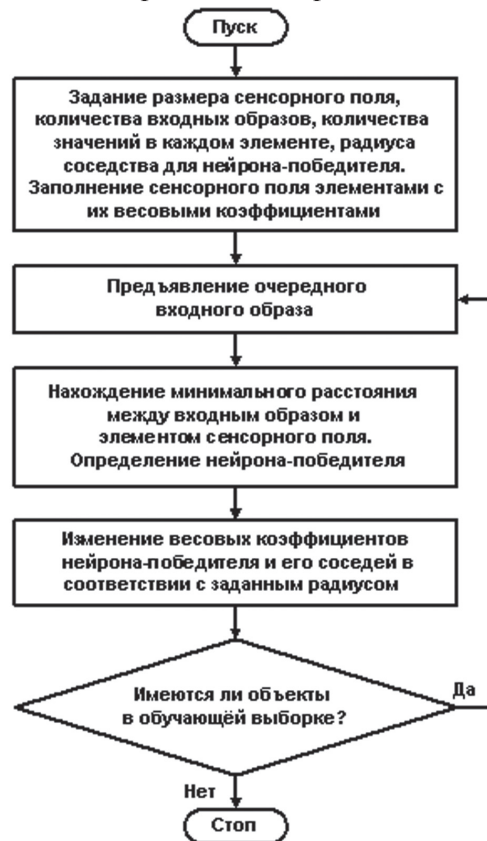


Рис. 1. Блок-схема алгоритма обучения классической самоорганизующейся карты Кохонена

На блок-схеме не детализированы процедуры начального заполнения сенсорного поля (с использованием генератора случайных чисел), ввода векторов обучающей последовательности V , выборки-предъявления образов из V и определения нейрона-победителя. Также не детализированы производимые математические операции (сравнение, нахождение минимального расстояния, изменение весовых коэффициентов). В остальном блок-схема иллюстрирует ключевые моменты работы классического варианта СКК.

В рамках представленной блок-схемы (классического подхода) могут быть реализованы различные варианты стратегий обучения СКК. В частности, различия могут быть связаны с кратностью, последовательностью предъявления и предварительной сортировкой (группировкой) образов обучающей выборки.

Также эффективность процесса обучения существенно варьируется при изменении параметров функций соседства и влияния.

Существенно больше влияет на эффективность обучения СКК то, что предъявление входного образа, определение нейрона-победителя, изменение весовых векторов сенсорного поля (в соответствии с заданными функциями соседства и влияния) имеют *последовательный* характер. Значительное ускорение обучения возможно за счет модификации СКК с введением элементов *параллельности* (рис. 2).

3. Обучение модифицированных карт Кохонена

Параллельность обработки информации сенсорным полем СКК фундаментально противоречит последовательному характеру процедуры предъявления образов и обучения СКК. Сопоставление вектора обучающего образа с каждым из векторов сенсорного поля $B_k \leftrightarrow A_{ij}$ в принципе реализуемо в виде $p \times q$ параллельных независимых процедур (для каждого из $p \times q$ векторов A_{ij}). Однако перед сопоставлением предъявление элементов B_k производится последовательно. После сопоставления также последовательно для каждого B_k производится коррекция сенсорного поля. Снятие данного противоречия посредством модифицирования структуры связей и процедуры обучения СКК потенциально обеспечивает положительный эффект: повышает скорость и эффективность обучения и последующей эксплуатации СКК. Соответствующим образом изменяется структура организации информации, приобретаемой и хранимой СКК, а также эксплуатационные характеристики СКК после обучения. В [1] предложен вариант сокращения времени обучения СКК за счет распараллеливания обработки данных.

На рис. 2 представлен алгоритм обучения модифицированной сети Кохонена. Изначально задаются размеры двумерного массива (сенсорного поля), количество входных образов (обучающая выборка), а также радиус удалённости нейронов

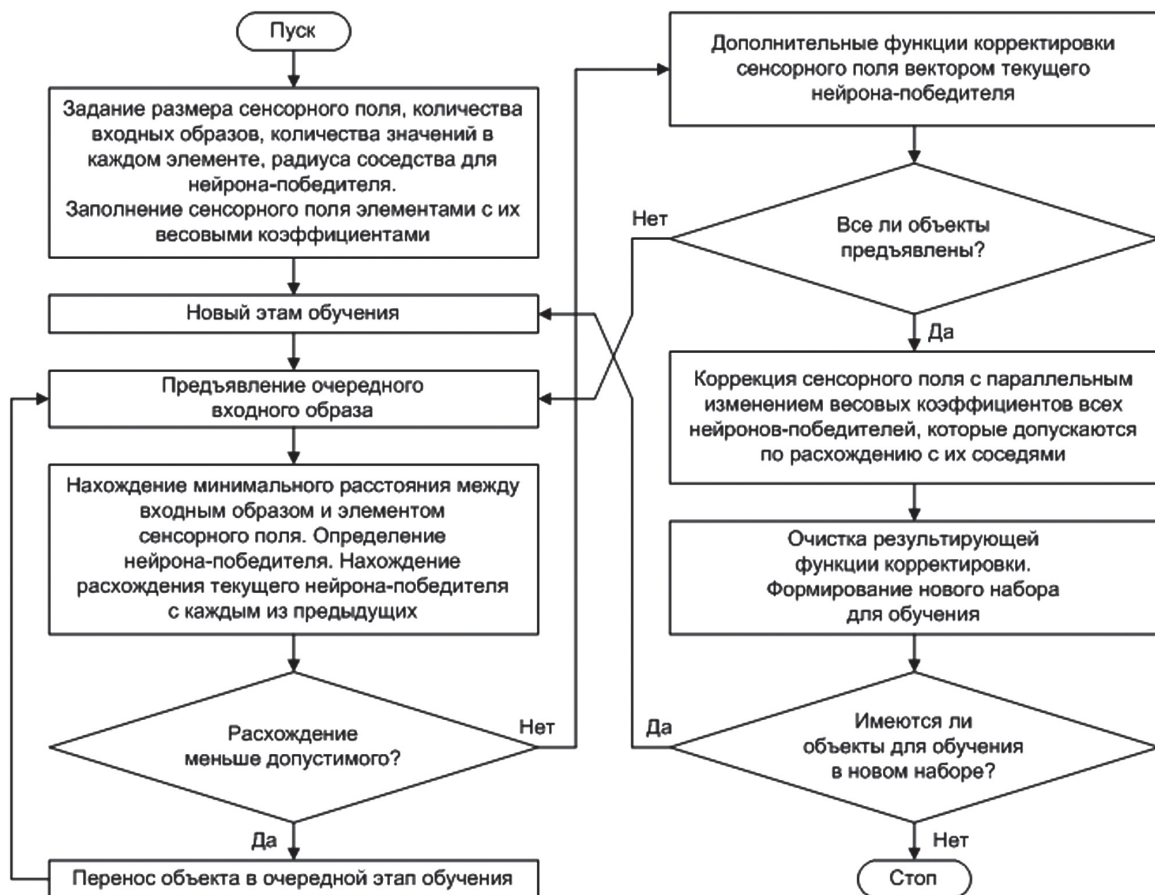


Рис. 2. Блок-схема алгоритма обучения модифицированной самоорганизующейся карты Кохонена

– соседей нейрона-победителя (упрощённый вариант представления функции соседства).

Массив сенсорного поля заполняется низкими (начальными, фоновыми) значениями. Порядок величины значений начального заполнения выбирается в соответствии с параметрами применяемой в дальнейшем функции влияния. Далее следует первый (начальный) этап обучения. Случайным образом выбирается и подается первый входной образ, определяется нейрон-победитель, рассчитываются его соседи (радиус окружения). Координаты нейрона-победителя и его рассчитанного окружения заносятся в массив, предназначенный для коррекции на данном этапе обучения.

Далее, в отличие от классического алгоритма, изменение весовых коэффициентов (коррекция сенсорного поля) не производится. Подается следующий (очередной) входной образ и определяется нейрон-победитель. Затем определяется расстояние до предыдущего или нескольких (всех предшествовавших) предыдущих нейронов-победителей. Если оно меньше допустимого, то данный входной образ заносится в новый массив обучающих образов, который будет предъявляться на следующем этапе обучения. Иначе – координаты нейрона-победителя заносятся в коррекционный массив. Обучение на данном этапе происходит до тех пор, пока из первоначальной обучающей выборки не будут предъявлены все входные образы.

После завершения предъявления всех входных обучающих образов, сенсорное поле корректируется на основе информации, собранной в коррекционном массиве. Коррекция происходит параллельным изменением значений весовых коэффициентов [1]. Далее начинается следующий (очередной) этап обучения. При этом в качестве обучающей выборки выступает массив из элементов, которые в предыдущем цикле обучения попадали в недопустимую область. Т.е. те элементы, для которых расстояния от текущего нейрона-победителя до предыдущих нейронов-победителей было меньше допустимого. Данный этап обучения проходит до тех пор, пока в обучающей выборке не останется входных образов и т.д. При этом формируется новый массив обучающих образов, состоящий из тех образов, которые на текущем этапе попали в недопустимую область.

Обучение СКК считается законченным в том случае, когда в новом (формируемом) обучающем массиве не оказывается ни одного элемента (образа). Это значит, что на текущем этапе обучения все образы были однократно предъявлены сенсорному полю. По поводу каждого из образов был определён нейрон-победитель, и на основе этой информации в конечном счёте были проведены необходимые коррекции сенсорного поля. Повторное (кратное) предъявление обучающей выборки с целью повышения степени обученности СКК (качества обучения) – вопрос стратегии обучения, непосредственно

не затрагиваемый рассмотренным алгоритмом работы модифицированной СКК (рис. 2).

4. Реализация процедур обучения

Ранее [2] было проделано статистическое моделирование описанной модифицированной процедуры обучения с вероятностным оцениванием динамики формирования повторных обучающих выборок. Моделированием [2] был продемонстрирован выигрыш в эффективности (по временным затратам) по сравнению с классическим вариантом СКК. Вместе с тем, реализованная статистическая модель [2] была ограниченной по числу параметров. В частности, на ней не были проведены исследования зависимостей эффективности от размера сенсорного поля и длины вектора элемента (нейрона) сенсорного поля.

Эти зависимости сняты на другой разработанной модели. В отличие от статистической модели [2], новая разработанная модель является имитационной. Она воспроизводит не только подачу тестового образа – снятие реакции СК, но и внутреннюю логику работы СКК – обработку сигнала. Этим открывается возможность реально оценить вклад модифицированной части алгоритма работы СКК в выигрыш по эффективности. Таким образом, модель предназначена для сравнительного анализа быстродействия классической и модифицированной процедур обучения СКК.

Модельное программное обеспечение разработано на языке C++ в интегрированной среде разработки приложений Microsoft Visual Studio 2010. Модель реализована в потоковом варианте обработки: данные вводятся из входного текстового документа, результаты выводятся в текстовый протокол. Анализ и визуализация результатов производятся отдельно от вычислительной части обработкой протокола. При работе с программой во входной текстовый документ заносятся исходные данные: тип алгоритма (классический или модифицированный), размер двумерного массива сенсорного поля, радиус соседства для нейрона-победителя на прямоугольной решетке, количество входных образов, количество значений (весовой вектор) для каждого элемента в двумерном массиве. В выходном текстовом протоколе накапливаются результаты работы программы – время обучения системы при заданных входных параметрах.

Для реализации модифицированного варианта подключены библиотеки MPI (Message Passing Interface), в частности, блок MPICH.NT. Замеры продолжительности работы программ проведены на следующей программно-аппаратной конфигурации: Intel® Pentium® 4 CPU (2 CPUs) 3.00GHz, 2048MB RAM, ATI Radeon HD 5600 Series, Windows 7 Ultimate (x64).

На рис. 3 а, б представлены некоторые из результатов работы программы для обучения традиционного (а) и модифицированного (б) вариантов СКК

в зависимости от размера двумерного массива сенсорного поля, и количества значений (компонентов вектора) каждого элемента (нейрона) массива.

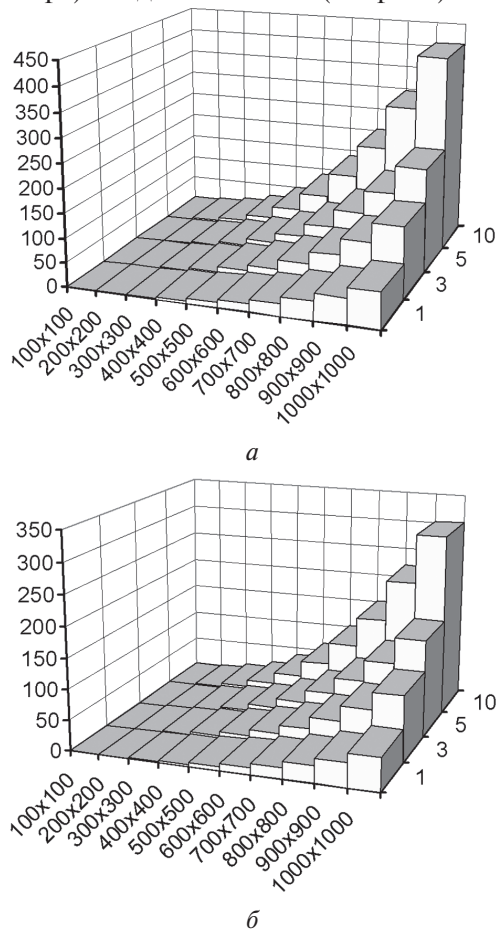


Рис. 3. Результаты работы программы для обучения традиционного (а) и модифицированного (б) вариантов самоорганизующейся карты Кохонена

Рассмотрены сенсорные поля с октагональной топологией (8 соседей) только квадратной формы (равенство сторон). Размеры массива задавались в диапазоне от 100×100 элементов до 1000×1000 с шагом 100×100 . Количество весовых коэффициентов в каждом элементе массива (нейроне сенсорного поля) задавалось равным 1, 3, 5 и 10.

Количество входных образов (объём обучающей выборки) определялось на уровне 75% от размера стороны массива сенсорного поля. Таким образом, введена нормировка числа тестовых образов в зависимости от размера сенсорного поля и эффективной ширины функции влияния нейрона-победителя (величины радиуса соседства). Этим обеспечено неплотное ограниченное заполнение сенсорного поля. Предварительные прогоны машинного эксперимента показали, что при введённой нормировке количество входных образов (объём заданной обучающей выборки) не оказывает ощутимого влияния на скорость обучения. Непосредственно сказываются только размер сенсорного поля, а также количество весовых коэффициентов (длины векторов образов). Поэтому основной (полноразмерный) машинный эксперимент спланирован по

этим двум параметрам при стандартных (нормированных указанным способом) объёмах обучающих выборок. Разумеется, при этом автоматически соблюдается сопоставимость традиционного и модифицированного вариантов СКК для каждой пары значений параметров. То есть графики а и б (рис 3) – взаимно-сопоставимы. Для каждой из комбинаций параметров замерено время выполнения всей процедуры обучения СКК (вертикальные оси на графиках рис. 3, значения в микросекундах).

Полученные двумерные зависимости качественно (по характеру поверхностей) одинаковы. Различие может быть охарактеризовано вертикальным масштабным коэффициентом порядка 1,3 – 1,4, иллюстрирующим выигрыш в производительности модифицированного варианта СКК по сравнению с традиционным. В остальном, как и предполагалось изначально, зависимости приблизительно являются квадратичными от размера поля и линейными от длины вектора.

5. Обсуждение результатов

Выигрыш в производительности модифицированного варианта СКК по сравнению с традиционным (порядка 1,3 – 1,4), повидимому, может быть уточнён. Как отмечалось, при построении модели были применены библиотеки MPI, реализующие определённый «внутренний интерфейс» (собственный порядок) взаимодействия с данными. Оба алгоритма (традиционный и модифицированный) были реализованы и продемонстрировали функциональность, но не был специально исследован вопрос о сопоставимости этих реализаций с учётом использования библиотек MPI.

Вместе с тем, из общих соображений, выигрыш порядка 1,3 – 1,4 может вполне соответствовать той доле вычислительного времени, которая экономится за счёт «объединения» операций корректировки сенсорного поля (см. рис. 2) в модифицированном варианте алгоритма СКК. Поэтому возможно и вероятно, что дополнительное исследование корректности применения библиотек MPI в данной задаче не опровергнет данного результата.

Проведенные исследования и полученные результаты иллюстрируют работу СКК при не слишком плотном заполнении сенсорного поля. Как отмечалось, при проведении машинного эксперимента была выбрана (принята) специальная нормировка по плотности размещения нейронов-победителей. При данной стандартной плотности заполнения графики рис. 3 демонстрируют пропорциональность выигрыша по времени для всего массива вариантов машинного эксперимента, а также отмеченные квадратичный рост с повышением размера сенсорного поля и линейный рост пропорционально длине вектора каждого элемента.

Снижение плотности имеет низкую прикладную ценность: соответствует слабой нагруженности (избыточной информационной ёмкости) СКК.

Повышение плотности размещения нейронов более перспективно, т.к. позволяет приблизиться к оценке предельных возможностей СКК с сенсорным полем заданного размера. В плане сравнения эффективности традиционного и модифицированного вариантов обучения СКК, повышение плотностей предположительно связано со снижением выигрыша в эффективности, приближением модифицированного варианта к традиционному. Можно ожидать, что с ростом плотности алгоритм рис. 2 будет формировать всё большее число коротких обучающих выборок. Т.о. модифицированный вариант СКК асимптотически приближается к традиционному. Безусловно, характер изменения эффективности модифицированного алгоритма СКК при указанном асимптотическом приближении чрезвычайно интересен как объект отдельного исследования.

Также отдельного рассмотрения требует общий случай прямоугольных (не квадратных) сенсорных полей. Не очевидно, что прямоугольные поля обеспечат более высокую эффективность модифицированного варианта СКК. Более вероятно снижение эффективности. Однако в плане практического (инженерного) использования может представить интерес вопрос о допустимости и пределах прямоугольности сенсорного поля при небольших потерях эффективности алгоритма. Подобная проблема может иметь смысл, в частности, в технических системах с частичным выходом элементов из строя и реконфигурированием сенсорного поля.

6. Перспективы использования

Дилемма «сосредоточенные системы» – «распределённая обработка» в действительности является не противоречием, а диалектическим моментом развития. Продуктивность сочетания данных противоположностей уже реализована в живой природе (естественные нейронные системы).

Как отмечалось, в предложенном модифицированном алгоритме СКК выигрыш эффективности достигается за счёт сокращения обработок сенсорного поля и обусловлен объёмом независимых (не взаимодействующих) обучающих образов в «пакете» (обучающей выборке). Подобный подход повсеместно используется в живой природе, в частности, в структурах реализующих интеллектуальные процессы, в том числе у человека.

Известен феномен человеческой психики – способность запомнить (воспринять) за краткое время ограниченное число объектов (правило 5 ± 2) [6]. Если интерпретировать этот феномен как аналог пакетной обработки информации, то одним из вариантов алгоритмической реализации может быть рассмотренный модифицированный вариант СКК.

Выводы

Рассмотрены (сопоставлены) реализации классического и модифицированного вариантов обучения самоорганизующейся карты Кохонена.

Проведен сравнительный анализ скорости обучения при различных типах входных данных. Сделаны оценки выигрыша эффективности. Полученные результаты интересны для выработки рекомендаций по распараллеливанию отдельных фаз обучения карт Кохонена при реализации модифицированного варианта на вычислительных системах, поддерживающих распределённую обработку данных.

Список литературы: 1. Дяченко, В.А. Сеть Кохонена с параллельным обучением [Текст] / В.А. Дяченко, О.Ф. Михаль, О.Г. Руденко // Управляющие системы и машины. – 2009, № 5. – С. 14-18. 2. Дяченко, В.А. Параллельная процедура обучения в модифицированной сети Кохонена [Текст] / В.А. Дяченко, А.А. Кошкин, О.Ф. Михаль // Вестник ХНТУ. – 2011. – № 2 (41). – С. 146-149. 3. Дяченко, В.А. Повышение эффективности обучения модифицированной сети Кохонена обучением [Текст] / В.А. Дяченко, О.Ф. Михаль // Информатика, математическое моделирование, экономика: Сб. науч. статей по итогам Междунар. науч.-практич. конф., г. Смоленск, 22 апреля 2011 г. В 2-х томах. – 2011. Том 2. – С. 90-96. 4. Дяченко, В.А., Адаптивное параллельное обучение модифицированной самоорганизующейся карты Кохонена [Текст] / В.А. Дяченко, О.Ф. Михаль // Сучасні напрямки розвитку інформаційно-комунікаційних технологій та засобів управління : Матеріали другої міжнар. наук.-техн. конф. – К.: ДП «ЦНДІ НІУ», Х.: «ХНДІ ТМ»; К.: КДАВТ, 2011. – С. 44. 5. Руденко, О.Г. Основы теории искусственных нейронных сетей [Текст] / О.Г. Руденко, Е.В. Бодянский. – Харьков: ТЕЛЕТЕХ, 2002. – 317 с. 6. Милнер, П. Физиологическая психология [Текст] / П. Милнер. – М.: Мир, 1973. – 648 с.

Поступила в редакцию 29.11.2011

УДК 519.87

Адаптивне паралельне навчання модифікованої самоорганізованої карти Кохонена / В.О. Дяченко, О.П. Міхаль // Біоніка інтелекту : наук.-техн. журнал. – 2012. – № 1 (78). – С. 85-90.

Зроблена оцінка ефективності класичного і модифікованого варіантів навчання самоорганізованих мереж Кохонена. При використанні різних типів навчальних даних зроблено порівняльний аналіз швидкості навчання. Запропоновано програмне забезпечення, що реалізовує модифікований варіант навчання мереж Кохонена. Отримано результати роботи карт Кохонена при незадатно щільному заповненні сенсорного поля.

Л. 3. Бібліогр: 6 найм.

UDK 519.87

Adaptive parallel teaching of modified self-organizing Kohonena card / V.A. Djachenko, O.Ph. Mikhal // Bionics of Intelligense: Sci. Mag. – 2012. – № 1 (78). – P. 85-90.

The estimation of classical and modified variants of algorithms of teaching of self-organizing Kohonen cards was done. The benchmark analysis is made as to velocities of the teaching of system by using different types of input data. The software of modified variant of educating of networks of Kohonen cards was realized and offered. The results of work Kohonen cards with not too dense filling was offered.

Fig. 3. Ref.: 6 items.

УДК 681.513



И.А. Осотов, О.Ф. Михаль

ХНУРЕ, г. Харьков, Украина, fuzzy16@pisem.net

ОЦЕНКА ЭФФЕКТИВНОСТИ ПРОВЕДЕНИЯ ЛОКАЛЬНО-ПАРALLELНЫХ ВЫЧИСЛЕНИЙ НА МНОГОЯДЕРНЫХ ПРОЦЕССОРАХ

На основе анализа тенденций развития процессорной техники обоснована перспективность расширенного использования локально-параллельных методов обработки. В машинном эксперименте реализован выигрыш в эффективности ~15% для локально-параллельного алгоритма с потоком обработки данных внутри распараллеленного участка.

ЛОКАЛЬНО-ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ИНФОРМАЦИИ, МНОГОЯДЕРНЫЕ ПРОЦЕССОРЫ

Введение

Объекты живой природы, в том числе *живые системы* (ЖС), наделённые интеллектом (в различной степени), по комплексу характеристик надёжности, устойчивости, способности к самообучению и др. существенно превосходят образцы *технических систем* (ТС), разработанных человеком. При этом сами ЖС собраны из неустойчивых и ненадёжных компонентов, то есть высокие *эксплуатационные характеристики* (ЭХ), включая интеллектуальный аспект, достигнуты в них за счёт системных и структурных решений. В связи с этим актуален круг проблем по системному исследованию интеллектуальных ЖС для использования соответствующих структурных решений в ТС. В [1, 2] рассмотрены иерархические параллельно-организованные ЖС с ретроспективой на тенденции развития *вычислительных устройств* (ВУ). Показано, что перспективно использовать ВУ общего назначения с программной реализацией на них *локально-параллельных* (ЛП) методов обработки информации. Новый аспект применения принципа ЛП связан с многоядерностью процессоров ВУ. На настоящий момент ЛП обработка на многоядерных процессорах недостаточно изучена. В частности для известных [1, 2] ЛП алгоритмов не разработаны версии их выполнения на многоядерных процессорах; ввиду чего уже разработанные [3, 4] методы исследования для многоядерных процессоров не апробированы на известных ЛП алгоритмах. Между тем, ожидаемый прирост эффективности составляет десятки процентов, чем определяется целесообразность проведения разработок в данном направлении.

Целью настоящей работы является оценка эффективности проведения ЛП вычислений для конкретных типов алгоритмов на многоядерных процессорах.

1. Иерархичность и параллеленность

В плане обеспечения надёжности и устойчивости функционирования ЖС (объекты живой природы) являются неоспоримыми (не теряющими актуальность) прототипами для ТС. Свойства ЖС по самовосстановлению, авторепродуцированию,

обучаемости, приспособляемости к агрессивным внешним условиям и др. по настоящее время в технических устройствах не воспроизведены. Базовые биологические (высокомолекулярные) материалы, из которых собраны ЖС, имеют малую устойчивость и критичны к условиям внешней среды. Тем не менее, вследствие иерархической структурной организации ЖС демонстрируют долговременное устойчивое функционирование и приспособляемость – сохранение *эксплуатационных характеристик* (ЭХ) в широких диапазонах по ряду параметров ($P_1, P_2, \dots, P_j, \dots$), задаваемых внешней средой. В этом они значительно превосходят менее сложно организованные, и потому казалось бы более надёжные, ТС. Важнейшими особенностями обеспечения высоких ЭХ в ЖС, являются многоуровневость (иерархичность) и параллельная организация в каждом из уровней (внутриуровневый локальный параллелизм).

Упрощённо иерархичность и распараллеленность ЖС выглядят следующим образом. Имеются объекты ЖС и внешняя среда, в которой находятся ЖС. Каждый из объектов ЖС является многоуровневым по сложности. На i -м уровне сложности ЖС состоит из функционально однотипных объектов, каждый из которых состоит из объектов $(i-1)$ -го уровня. Понятие «внешняя среда» – относительное: объект $(i+1)$ -го уровня является внешней средой для объектов $(i-1)$ -го уровня. Поэтому группы объектов $(i-1)$ -го уровня объединяются в объекты i -го уровня сложности с целью противостоять воздействиям внешней среды. Для этого на i -м уровне однотипные элементы $(i-1)$ -го уровня взаимоорганизуются так, что в целом на i -м уровне обеспечиваются обратные связи, поддерживающие объект i -го уровня в рамках параметров $(p_{min,j,i}; p_{max,j,i})$, где в индексах: j – номер параметра внешней среды P_j ; i – номер уровня иерархической организации. При этом для ЖС в целом по j -му параметру внешней среды P_j обеспечивается вложенность интервалов значений параметров:

$$p_{min,j,1} \geq p_{min,j,2} \geq p_{min,j,3} \geq \dots \geq p_{min,j,(i-1)} \geq p_{min,j,i} \geq \dots; \\ p_{max,j,1} \leq p_{max,j,2} \leq p_{max,j,3} \leq \dots \leq p_{max,j,(i-1)} \leq p_{max,j,i} \leq \dots \quad (1)$$

В результате оказывается, что имеется последовательная вложенность «сред обитания» объектов

«друг в друге», подобная игрушке «Матрёшка». При этом объект иерархии ЖС, находящийся на верхнем уровне, эффективно функционирует в «самой внешней» «истинно окружающей» среде, разрушительной для исходных материалов (базовый уровень), из которых собрана ЖС.

Человек является частным случаем (одним из видов) ЖС, поэтому приведённое описание (системный генезис) плюс продолжение на социальный аспект развития относятся также и к нему. Деятельность человека в системном плане направлена на расширение своих возможностей и совершенствование выполняемых функций. Поэтому разрабатываемое человеком искусственное (рукотворное) окружение (от примитивных орудий труда до ВУ и средств телекоммуникации) является его антропоморфным продолжением: усилением возможностей, улучшением системных ЭХ. Исходя из этого, целесообразны разработки с соблюдением принципов иерархичности и параллельности в каждом из уровней. Этим, в частности, обеспечиваются надёжность, взаимозаменяемость, возможность реконфигурирования при частичных отказах, а также дополнительная функциональность за счёт параллельного (фоновое) использования резервных возможностей.

Такова общая картина, часто остающаяся вне поля зрения за техническими подробностями. В аспекте разработки ВУ, как интеллектуального элемента человеческого окружения, рассмотрим некоторый частный случай: «внедрения» (проникновение) нового программного распараллеливания внутрь существующего аппаратного уровня. Особенностью является то, что ВУ многоядерное и рассматривается внутриуровневая *программная* распараллеленность на уровне, который сам уже является *аппаратно* распараллеленным.

2. Локальная параллельность

Принцип ЛП обработки информации рассмотрен в [1, 2]. Эффективность ЛП обработки иллюстрируется следующим алгоритмическим примером. Пусть имеется n выражений $c_i = a_i + b_i$; $i = 1, 2, \dots, n$. Значения a_i и b_i заданы. Требуется найти значения c_i . На одноядерном ВУ данная задача разрешима за $4n$ шагов: по каждому из выражений загрузить a_i , загрузить b_i , произвести суммирование, выгрузить результат c_i . Согласно ЛП принципу значения a_i и b_i конкатенируются:

$$A = a_1 \oplus 0 \oplus \dots \oplus 0 \oplus a_2 \oplus 0 \oplus \dots \oplus 0 \oplus a_n;$$

$$B = b_1 \oplus 0 \oplus b_2 \oplus 0 \oplus \dots \oplus 0 \oplus b_i \oplus 0 \oplus \dots \oplus 0 \oplus b_n; \quad (2)$$

а совокупность всех n результатов c_i получается как конкатенация

$$C = c_1 \oplus 0 \oplus c_2 \oplus 0 \oplus \dots \oplus 0 \oplus c_i \oplus 0 \oplus \dots \oplus 0 \oplus c_n \quad (3)$$

в ходе выполнения 4-шаговой операции

$$C = A + B. \quad (4)$$

Разделительные нули в (2) и (3) вставлены между конкатенантами для реализации переноса

разряда при суммировании. Числа a_i и b_i должны быть положительными и иметь ограниченную разрядность, так чтобы конкатенации (2) и (3) не превышали по длине в битах разрядность процессора.

Разумеется, n -кратный выигрыш достигнут без учёта затрат времени на операции конкатенации-деконкатенации, окаймляющие вычислительный блок. Если речь идёт о короткой вычислительной процедуре типа (4), то ЛП вариант реализации скорее всего будет проигрышным по времени ввиду наличия вспомогательных операций формирования (2) и извлечения результатов c_i из (3). Но в реальных расчётных задачах, когда вычислительный блок существенно сложнее (4) и включает более значительный набор разнообразных операций, выполняемых над наборами данных без промежуточных конкатенации-деконкатенации, выигрыш в производительности может быть значительным. Выигрыш растёт с ростом разрядности процессора, а также со снижением точности представления данных («загрублением» системы) при сокращении размеров конкатенируемых сегментов. Оба указанных направления имеют конкретные технические приложения.

Рост разрядности процессоров является долговременной устойчивой тенденцией развития процессорной техники. Разрядность регистра процессора – это объём непосредственно адресуемой оперативной памяти. В конечном счёте, объёмом памяти определяется предельно допустимая размерность задач, которые могут быть разрешены на данном ВУ за приемлемое время. Развитие техники ВУ стимулирует появление новых вычислительных задач, которые в свою очередь стимулируют рост требований к техническим характеристикам вновь разрабатываемых ВУ. Данная петля обратной связи исправно функционирует уже полвека и нет принципиальных причин отхода от данной парадигмы.

Снижение точности представления данных до приемлемых пределов допустимо и целесообразно, в частности, в системах, базирующихся на использовании экспертных знаний. Носителями исходных (первоначальных) экспертных знаний являются люди. В связи с этим должен приниматься во внимание известный из психологии факт [5]: человек может устойчиво различать лишь малое число объектов: 5 ± 2 шт. При работе с большим числом объектов различение идёт с разделением на подуровни: предъявляемые объекты сначала группируются в блоки по 3–7 шт., затем последовательно осуществляется обработка блоков. Таким образом, для кодирования одного из реально различаемых экспертом состояний требуется до 3 бит информации. Использование более высокой разрядности может предполагать округление, сглаживание. То есть, возможно искажение исходных экспертных знаний.

Разумеется, различение 5 ± 2 объектов есть антропоморфная ЭХ. В дальнейшем в ходе эволюции ТС могут быть востребованы системы с

ЭХ, превосходящими антропоморфные. Но по-видимому это будет уже *чисто техническая* эволюция (эволюция ТС, реализуемая не людьми, а самими ТС), которая, возможно, уже не будет относиться к человеческой (*чисто человеческой*) истории. В тот отдалённый период человек скорее всего не будет являться единственным носителем экспертных знаний, и сама проблематика обработки информации приобретёт другие очертания. Текущий же этап развития ТС был и остаётся антропоморфным, и для него остаются справедливыми две сформулированные мотивации по расширенному использованию принципа ЛП:

- рост разрядности процессоров;
- ограниченность разрядности представления экспертных знаний.

Далее рассмотрим ЛП в плане практической реализации на многоядерных процессорах.

3. Многоядерные процессоры

В настоящее время разрядность процессоров общего назначения возросла до 64 бит. Имеются также специализированные графические процессоры, применяемые, в частности, в игровых приставках, имеющие разрядность 128 и 256 бит. При этом тактовая частота ядра процессора достигла единиц ГГц. Дальнейшая конъюнктура рынка процессоров, по-видимому, складывается так, что конкурентоспособный рост производительности ВУ не обеспечивается ростом разрядности регистров процессора. Для основной массы задач общего назначения разрядность 64 бита является достаточной для представления данных, а аппаратное ЛП-представление данных, по видимому, не обеспечивается существующими инженерными (схемными и конфигурационными) решениями в гигагерцевом диапазоне. В техническом отношении подобная задача, разумеется, является нетривиальной: речь идёт о микроминиатюризации в области СВЧ, где помимо чисто схемных решений существенны также и конфигурационные факторы. Вместе с тем процессорная техника должна развиваться, и характеристики должны наращиваться, поскольку развёрнуто и профинансировано несколько конкурирующих организационных структур, занимающихся разработкой и производством процессоров. В сложившейся ситуации, по-видимому, оптимальным по сложной группе критериев «капиталовложения – сроки разработки – выигрыш в производительности» является курс на создание многоядерных однокристалльных процессоров. Как известно [6], данное направление было впервые (прорывно, но преждевременно) реализовано специалистами фирмы Digital Equipment Corporation ещё в 1992 г. в виде 64-разрядного RISC процессора DEC-Alpha.

Перспективность однокристалльных многоядерных процессоров в техническом отношении состоит в том, что каждое из ядер может быть

высокочастотным, а взаимодействие между ядрами реализуется на более низких частотах через кеш-память 2-го, 3-го и т.д. уровней. Таким образом, иерархические уровни образуются здесь по вложенности, объёмам памяти и убыванию рабочих частот. Вложенность уровней кеш-памяти соответствует (аналогична) изложенной выше (п. 1) концепции об иерархичности ЖС, а последовательность частот, убывающая по уровням кеш-памяти, – вложенности диапазонов значений параметров (1).

Могут также реализовываться различные иерархии дисциплин по автономному или совместному использованию различных уровней кеш-памяти. Представляется достаточно вероятным, что прогресс по числу ядер будет сопровождаться прогрессом по организации многоуровневой кэш-памяти, направленным на повышение рабочих частот и усиление обобществления памяти для эффективного взаимодействия между отдельными ядрами.

В целом переход от одноядерной модели процессора к многоядерной определился, по-видимому, технологическими ограничениями по освоению СВЧ-диапазона на микроэлектронном уровне, т.е. отсутствием инженерного опыта и разработанных технических решений. Поэтому в конечном счёте путь развития в дальнейшем будет определяться накоплением инженерного опыта при разработке процессорных систем по мере освоения СВЧ-диапазона.

В рамках описанного (возможно, ограниченно) понимания перспектив развития процессорной техники сегодняшние многоядерные однокристалльные процессоры предстают как промежуточный этап перед полностью СВЧ-частотными процессорами с большой разрядностью регистров и реконфигурируемостью по числу ядер (вычислительных узлов). В самом деле: n -ядерный однокристалльный процессор с разрядностью каждого ядра m может быть эквивалентен nm -разрядному одноядерному процессору, если обеспечена связь (передача данных) на тактовой частоте ядер между старшим разрядом i -го и младшим разрядом $(i+1)$ -го ядер. В этом случае регистры отдельных ядер процессора конкатенированы в гиперрегистры виртуального единого гиперядра.

В связи со сказанным перспективна реализация ЛП алгоритмов на многоядерных структурах. Технически и программно – ЛП и многоядерность не противоречат друг другу, а в идейном отношении они направлены на реализацию единой цели: внутриуровневой распараллеленности системы.

Рассмотрим далее комбинированное ЛП многоядерное алгоритмическое решение на примере ЛП алгоритма операции нахождения нечёткого теоретико-множественного пересечения двух наборов значений функций принадлежности [1, 2].

4. Алгоритмы

В табл. 1 приведено пошаговое описание ЛП алгоритма нечёткой теоретико-множественной

операции пересечения. Операнды A и B являются конкатенациями масштабированных значений функции принадлежности [2, 3]. Количество и длина сегментов (на примере 10-сегментного 3-битового варианта, удобного для реализации на 32-разрядных ВУ) определяются константами

$$E_1=119304647_{10}=000111000111000111000111000111_2,$$

$$E_2=954437176_{10}=111000111000111000111000111000_2,$$

с использованием которых выделяются множества нечётных и чётных (соответственно) сегментов. Числа 10 и 2 в индексах при значениях E_1 и E_2 обозначают системы счисления. Константы L_1 и L_2 предназначены для помещения «сигнальных» единиц в младшие разряды сегментов. В 10-сегментном 3-битовом варианте – $L_1=1090785344_{10}$; $L_2=653667127_{10}$.

Таблица 1

Пошаговое описание последовательного варианта локально-параллельной операции нечёткого теоретико-множественного пересечения

Шаг	Метод реализации $min(A, B)$
1	Прореживание: $A_1=A$ and E_1 ; $A_2=A$ and E_2 ; $B_1=B$ and E_1 ; $B_2=B$ and E_2 .
2	Установка меток и арифметическое вычитание: $C_1=(A_1 or L_2) - B_1$; $C_2=(A_2 or L_1) - B_2$.
3	Сопряженное прореживание: $C_1=C_1$ and L_2 ; $C_2=C_2$ and L_1 ; или $C_1=C_1$ and E_2 ; $C_2=C_2$ and E_1 .
4	Склеивание: $C=C_1$ "or" C_2 .
5	Формирование маски: $M=C - (C >> n)$.
6	Формирование инверсной маски: $\bar{M} = M xor E_0$.
7	Результат:
	$(A and \bar{M}) or (B and M)$;

Смысл остальных обозначений в табл. 1 следующий: $E_0=E_1+E_2$; $L_0=L_1+L_2$; and, or и xor – битовые логические операции; $(C >> n)$ – битовый сдвиг регистра процессора, содержащего C , на n бит в сторону младших разрядов с потерей содержимого n младших разрядов; при 3-битовых сегментах $n = 3$.

После склеивания (Шаг 4) формируются прямая и инверсная маски, с использованием которых собирается результат (Шаг 7).

Легко видеть, что Шаги 1-3 содержат по две группы действий: производимые с чётными и нечётными сегментами. Эти группы являются алгоритмически распараллеливаемыми. Затем в Шагах 4-7 результаты объединяются и вычисления идут последовательно.

Двухъядерный вариант алгоритма представлен в табл. 2. Отличие от варианта табл. 1 – разнесение параллельных операций в параллельно выполняемые Шаги. Таким образом, на одноядерных процессорах весь алгоритм может выполняться последовательно (таблица 1), а на многоядерных – Шаги

1.1, 2.1, 3.1 выполняются параллельно с Шагами 1.2, 2.2, 3.2, затем остальная часть – последовательно (табл. 2). В результате, на многоядерных процессорах потенциально достигим дополнительный прирост эффективности – сокращение времени обработки.

Таблица 2

Пошаговое описание варианта локально-параллельной операции нечёткого теоретико-множественного пересечения с распараллеливанием

Шаг	Метод реализации $min(A, B)$
1.1	Прореживание: $A_1=A$ and E_1 ; $B_1=B$ and E_1 .
1.2	Прореживание: $A_2=A$ and E_2 ; $B_2=B$ and E_2 .
2.1	Установка меток и арифметическое вычитание: $C_1=(A_1 or L_2) - B_1$.
2.2	Установка меток и арифметическое вычитание: $C_2=(A_2 or L_1) - B_2$.
3.1	Сопряженное прореживание: $C_1=C_1$ and L_2 ; или $C_1=C_1$ and E_2 .
3.2	Сопряженное прореживание: $C_2=C_2$ and L_1 ; или $C_2=C_2$ and E_1 .
4	Склеивание: $C=C_1$ "or" C_2 .
5	Формирование маски: $M=C - (C >> n)$.
6	Инверсная маска: $\bar{M} = M xor E_0$.
7	Результат:
	$(A and \bar{M}) or (B and M)$;

Разработанная ранее [3, 4] методика машинного эксперимента для анализа эффективности была адаптирована с учётом структурной схемы ЛП алгоритма (табл. 1, 2). Согласно плану эксперимента замеры затрат времени на выполнение проводились по фиксациям моментов времени для одно- и двухъядерного вариантов реализации алгоритма при одинаковом числе итераций во внутреннем цикле.

Коэффициент эффективности определяется как отношение времён выполнения одноядерного варианта (аппаратно-последовательного) к двухъядерному (с внутрипроцессорным аппаратным распараллеливанием). То есть, если коэффициент превышает единицу – имеется выигрыш по эффективности.

Программная реализация механизма внутрипроцессорного распараллеливания рассмотрена в [3, 4]. Применены стандартные языковые конструкции инструментария WinAPI: функции SetThreadAffinity-Mask и SetThreadIdeal-Processor.

На рис. 1 представлена блок-схема программы для исследования роста эффективности распараллеленного варианта алгоритма (табл. 2) по сравнению с последовательным (табл. 1). В левом столбце представлены преимущественно обслуживающие процедуры: ввод исходных данных, генерация массивов значений, организация внешнего цикла накопления результатов, статобработка, протоколирование. Собственно исследование (получение

данных) происходит внутри цикла по набору статистики (рис. 1, правый столбец). Внутри цикла выполняются:

- пустая прокрутка (без обработки);
- одноядерная реализация алгоритма (табл. 1);
- многоядерная (в данном случае двухядерная) реализация алгоритма (табл. 2).

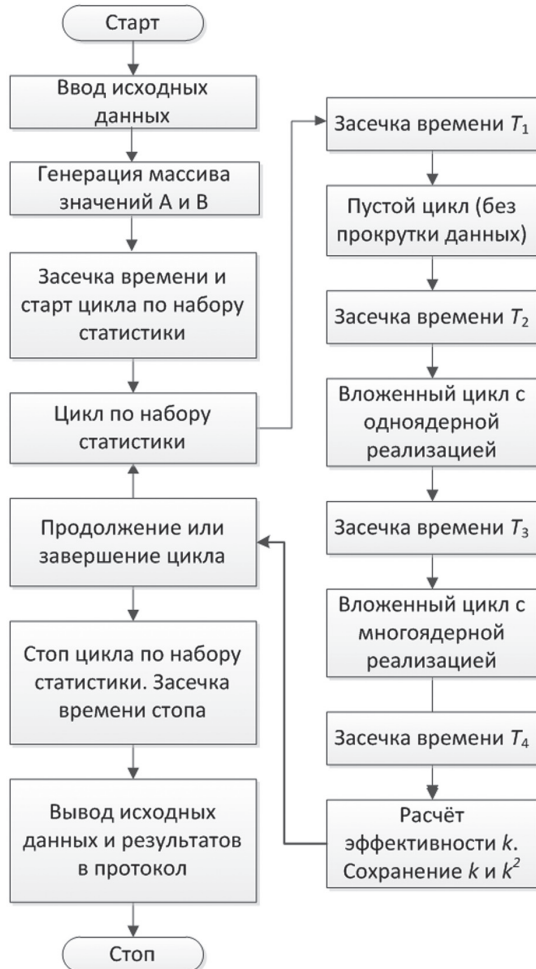


Рис. 1. Блок-схема программы исследования роста эффективности распараллеленного алгоритма по сравнению с последовательным

Число итераций по выполнению указанных трёх процедур является одним из параметров исследования. В частности, в группе данных, представленной ниже на рис. 2, число итераций варьируется в пределах 2 порядков: $10^4 - 10^6$. Перед, после и между процедурами снимаются засечки времени T_1, T_2, T_3, T_4 . Время выполнения пустого цикла вычитается из времён выполнения одноядерного и многоядерного циклов. Коэффициент эффективности

$$k = \frac{(T_3 - T_2) - (T_2 - T_1)}{(T_4 - T_3) - (T_2 - T_1)}$$

накапливается в первой степени и в квадрате, для расчёта моментов статистического распределения.

В ходе *предварительных* экспериментов установлено, что распараллеливание одинарной ЛП вычислительной операции не приводит к выигрышу в производительности. Одноядерная реализация ЛП

алгоритма выполняется даже несколько быстрее многоядерной. Это обусловлено тем, что выигрыш от распараллеливания теряется на фоне затрат на процедуры выделения конкретных ядер под конкретные операции. Выигрыш появляется, если используются массивы входных данных: распараллеливание вычисления происходит на потоках, привязанных к конкретным ядрам процессора.

С учётом обнаруженной особенности, *окончательный* план эксперимента был скорректирован: в двухъядерном варианте итерации по Шагам 1-3 и 4-7 были выделены в две отдельные группы с сохранением одинакового числа итераций в каждой группе. В исправленном варианте структура распределения ядер процессора задаётся однократно для всей группы данных, а не переназначается всякий раз заново. Таким образом, исчезают потери времени на переназначение, которые в действительности непосредственно не относятся к проведению вычислений в алгоритме. Как и ожидалось, с ростом числа итераций выигрыш от распараллеливания начинает превалировать над потерями времени на выделение ядер.

Исследования проведены на аппаратно-программной платформе Intel Pentium Dual-Core E5200; 2.5 GHz; 4 Gb RAM; OS Windows 7 x86. Типовые полученные результаты иллюстрируются графиками (рис. 2 а и б). На графиках по оси абсцисс отложено число итераций в логарифмических координатах. На рис. 2 а представлены зависимости времени выполнения циклов итераций в микросекундах для одноядерной (верхняя кривая) и многоядерной (нижняя кривая) реализаций. Согласно числовым данным на участке до ~ 30000 итераций график одноядерной кривой проходит ниже графика многоядерной, что отображено зависимостью коэффициента эффективности (рис. 2 б). Таким образом, представленный участок изменения числа итераций – в пределах двух порядков – отображает переход от проигрышности к выигрышности по эффективности в точке ~ 30000 итераций.

Полученные результаты демонстрируют выигрыш эффективности $\sim 15\%$ при циклической (поточной) организации вычислений внутри распараллеленной части алгоритма, что характерно, в частности, при обработке массивов однотипных данных. Как следует из представленного на рис. 2 б, при объёме итераций свыше 100000 график эффективности «выходит на полку».

Наличие «полки» на графике указывает на «стабилизацию» зависимости, приближение к асимптоте, являющейся оценкой выигрыша от распараллеливания по сравнению с потерями на однократное распределение потоков по ядрам процессора.

Переход от проигрышности алгоритма к выигрышности может быть охарактеризован как «плавный» или «резкий», в зависимости от масштаба итераций, применительно к конкретной решаемой

задаче. График рис. 2б иллюстрирует изменение числа итераций на 2 порядка и охватывает переход между двумя «полками»: превалирование затрат времени на выделение ядра и выигрыш от распараллеливания вычислений. Таким образом, выигрышность алгоритма критична к динамическому диапазону по числу итераций. Если алгоритм должен применяться к выборкам разного объёма в районе «перехода между двумя полками», то в вопросе эффективности (по выбору между одноядерным и двухядерным вариантами) требуется дополнительное исследование.

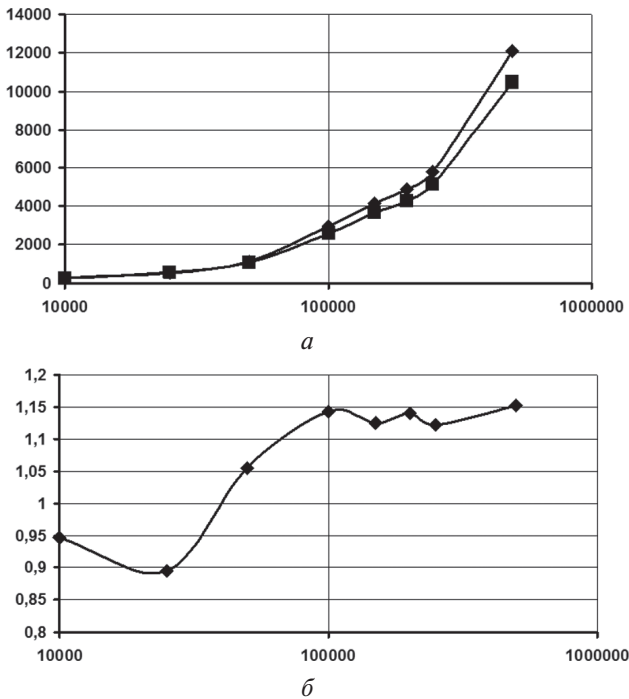


Рис. 2. Зависимости времени выполнения (а) и коэффициента эффективности (б) от числа итераций в машинном эксперименте

ЛП алгоритм нечёткого теоретико-множественного пересечения (табл. 1 и 2), взятый за основу для оценки эффективности проведения многоядерных ЛП вычислений, является типовым по сложности для алгоритмов нечёткой логики [1, 2]. Как отмечено выше (п. 2), обрабатываются неотрицательные ограниченные величины, интерпретируемые как значения функции принадлежности. Аналогичную сложность имеют ЛП алгоритмы алгебраических нечётких операций. Поэтому полученный результат может рассматриваться как типовой для наработанного класса ЛП алгоритмов операций нечёткой логики.

Интересна так же оценка эффективности для данного класса алгоритмов на матричных операциях. При этом появляются дополнительные параметры машинного эксперимента, связанные с тем, что матрица есть двумерный объект. Соответственно, обработка элементов матрицы в комбинации с ЛП представлением данных может распараллеливаться на большее число ядер.

Выводы

На примере алгоритма локально-параллельной операции нечёткого теоретико-множественного

пересечения методом машинного эксперимента показано, что выигрыш в эффективности от использования внутрипроцессорной многоядерности может составлять десятки процентов. С применением стандартного инструментария WinAPI реализован выигрыш ~15% при потоковой обработке данных внутри распараллеленного участка. Потоковая структура приемлема, в частности, при обработке крупных массивов данных, для повышения производительности обработки которых и предназначены локально-параллельные алгоритмы.

Список литературы: 1. Михаль, О.Ф. Моделирование распределенных информационно-управляющих систем средствами локально-параллельных алгоритмов обработки нечеткой информации [Текст] / О.Ф. Михаль // Проблемы бионики. Всеукраинский межведомственный научно-технический сборник. – Харьков: ХНУРЭ. – 2001. – Вып. 54. – С. 28-34. 2. Михаль, О.Ф. Принципы организации систем нечеткого регулирования на однородных локально-параллельных алгоритмах [Текст] / О.Ф. Михаль, О.Г. Руденко // Управляющие системы и машины. – 2001. – № 3. – С. 3-10. 3. Осотов, И.А. Применение локально-параллельных алгоритмов с процессорно-зависимой многопоточностью [Текст] / И.А. Осотов, О.Ф. Михаль // Информационные технологии в навигации и управлении: состояние и перспективы развития. Материалы первой международной научно-технической конференции. – К.: ДП «ЦНДІ НіУ», 2010. – С. 53. 4. Осотов, И.А. Реализация локально-параллельных алгоритмов с использованием процессорно-зависимой многопоточности [Текст] / И.А. Осотов, О.Ф. Михаль // Информатика, математическое моделирование, экономика: Сборник научных статей по итогам Международной научно-практической конференции, г. Смоленск, 22 апреля 2011 г. В 2-х томах. Том 1 – Смоленск: Смоленский филиал АНО ВПО ЦС РФ «Российский университет кооперации», 2011. – С. 14-20. 5. Милнер, П. Физиологическая психология [Текст] / П. Милнер. – М.: Мир, 1973. – 648 с. 6. Википедия. DEC Alpha // (http://ru.wikipedia.org/wiki/DEC_Alpha).

Поступила в редакцию 5.12.2011

УДК 681.513

Оцінка ефективності проведення локально-паралельних обчислень на багатоядерних процесорах / І.А. Осотов, О.Ф. Михаль // Біоніка інтелекту : наук.-техн. журнал. – 2012. – № 1 (78). – С. 91-96.

Проведено машинний експеримент щодо оцінки ефективності застосування внутрішньо-процесорної багатоядерної обробки в алгоритмі локально-паралельної операції нечіткого теоретико-множинного перетину. За потокової обробки даних всередині паралельної частки з застосуванням стандартного інструментарію WinAPI реалізовано вигреш ефективності порядку 15%.

Табл. 2. Л. 2. Бібліогр.: 6 найм.

UDK 681.513

Estimation of efficiency of local-parallel calculations based on multikernel processors / I.A. Osotov, O.Ph. Mikhal // Bionics of Intelligense: Sci. Mag. – 2012. – № 1 (78). – P. 91-96.

The machine experiment is organized on estimation of efficiency of multikernel processing of local-parallel algorithm of fuzzy sets intersection operation. Advantage to efficiency is found of the order 15% under stream data processing in parallel area with using standard toolbox WinAPI.

Tabl. 2. Fig. 2. Ref.: 6 items.

УДК 004.89:528:519.816



А.В. Бельчева¹, Н.О. Манакова²

¹ ХНУРЕ, м. Харків, Україна annija@yandex.ru

² ХНУРЕ, м. Харків, Україна natalym@rambler.ru

АНАЛИТИЧЕСКИЙ ВЫБОР ИСХОДНЫХ ДАННЫХ НА ОСНОВЕ МЕТОДА ИЕРАРХИЙ С УЧЕТОМ ТЕМАТИКИ ГЕОИНФОРМАЦИОННОГО ПРОЕКТА

В статье продемонстрирован метод выбора исходных данных для ГИС-проекта с помощью иерархической модели. На основе предложенной классификации определены приоритеты критериев всех уровней и значения матрицы попарных сравнений. Такой способ выбора данных основан на значениях приоритетов критериев, которые зависят только от тематики проекта.

ГИС, МЕТОД ИЕРАРХИЙ, КАЧЕСТВО ДАННЫХ, КЛАССИФИКАЦИЯ

Введение

ГИС (геоинформационные системы) обеспечивают ввод, анализ, прогноз и вывод пространственно-связанных данных о территориях, объектах, социальных и природных явлениях. ГИС объединяют различных предметные области и технологии. Это привело к тому, что ГИС на сегодняшний день решают очень большой круг прикладных задач из различных сфер человеческой деятельности. Такое многообразие задач и активное развитие ГИС-технологий предполагает наличие современной классификации ГИС-приложений.

В настоящее время существует несколько классификаций ГИС по различным признакам. Основные из них: территориальный охват, круг решаемых задач, целевое использование, области применения, тип обработки или представления географических данных, архитектурное построение и т.д. Однако данные классификации устарели и не отражают современного состояния развития ГИС технологий. В настоящее время очень активно развиваются такие отрасли как WEB-картография, электронное правительство, туризм и т.д. что не отражено в подобных классификациях.

На первых этапах разработки ГИС-приложения необходимо выбрать набор исходных данных, от чего во многом зависит и срок и цена проекта. На

начальной стадии проектирования критерии выбора картографического материала определяются, в основном, тематикой приложения. На основе метода иерархий можно сделать аналитический выбор исходных данных проекта, учитывая его направленность.

1. Классификация ГИС-приложений по тематике проекта

На сегодняшний момент по тематике ГИС-приложения можно классифицировать следующим образом (рис. 1):

1. Картография и инженерная геодезия.

Создание и актуализация карт и планов. Является пространственной основой для построения ГИС-приложений.

2. Управление инженерными сетями и коммуникациями.

Система автоматизированного управления инженерными сетями и коммуникациями активно используется организациями, которые обеспечивают коммунальные услуги:

Управление сетями инженерно-технического обеспечения:

- системы электро-, тепло-, водо-, газо- снабжения,
- системы вентиляции и кондиционирования,
- системы наружного освещения,



Рис. 1. Классификация ГИС-приложений

– системы связи (телефон, кабельные системы, контроля доступа, диспетчерского управления).

Управление транспортом (воздушные водные и наземные транспортные сети).

3. *Управление охраной (экология) и разработкой природных ресурсов. Экологическая безопасность.*

Системы, направленные на исследование, охрану и мониторинг природных ресурсов, обеспечение экологической безопасности:

- системы геологоразведки,
- системы природопользования,
- системы планирования заповедных зон,
- системы обеспечения экологической безопасности (активно используются аппараты построения буферных зон для оценки ожидаемого или причиненного ущерба от различных катастроф и стихийных бедствий).

4. *E-Government – Электронное правительство.*

Система, обеспечивающая принятия решений на государственном уровне.

Контроль условий жизни населения, здравоохранения, социальное обслуживание, трудовая занятость [3].

Курс на создание системы «информационного общества» в нашей стране был взят еще в 1996 году. На сегодняшний день создан правительственный web-портал, информационно-поисковая система Верховной Рады Украины, системы электронного документооборота ведомств, информационно-аналитические системы по работе с обращениями граждан, электронные системы государственной службы занятости, разработка правительственной информационно-аналитической системы Украины по чрезвычайным ситуациям [2].

5. *Управление предприятиями и бизнесом.*

Системы территориально-экономический анализа:

- система инвентаризации и учета экономических ресурсов,
- система анализа рынка,
- система создания и анализа территории сбыта,
- система маршрутов доставки,
- система анализа распределения ресурсов,
- система нахождения оптимального места для размещения предприятий, которые относятся к сфере обслуживания населения [3].

6. *Управление земельными ресурсами, земельный кадастр.*

Система, обеспечивающая оценку возможности строительства объектов, зон отдыха, стоимости продажи жилья [1]. Анализ, инвентаризация и учет природного, хозяйственного и правового положения земель:

- система управления земельными ресурсами,
- система учета и анализа регистрации прав на недвижимое имущество и сделок с ним,

– анализ деятельности связанной с пользованием и распоряжением земельными участками,

– система контроля по использованию земельных ресурсов.

7. *WEB-картография.*

Пространственная навигация и информационная коммуникация, сферы, доступные сегодня любому желающему. Web-картография в связи с активным развитием интернет-технологий может оперативно доставлять пользователю необходимую информацию, позволяет использовать карту для добавления сервисов (различной информации: реклама, фотографии, комментарии). Этот класс ГИС-приложений привлекателен еще тем, что в основном использует открытый тип данных [1]. ГИС-приложения, которые используют WEB-картографию, можно классифицировать по инструментам разработки:

- виртуальные глобусы (Google Maps, Google Earth, Virtual Earth, ArcGIS Explorer),
- пользовательские ГИС (ArcGIS, Mapinfo, QGIS, gvSIG),
- картографические веб-серверы (MapServer, GeoServer, OpenLayers).

8. *Безопасность, военное дело и разведка.*

В военном деле необходимо постоянно проводить анализ местности, планирование и управление боевыми операциями, учет военного хозяйства, размещение объектов, ведение разведки:

- системы командирования и контроля. Анализ и принятие решений командира любого уровня в военной сфере всегда связаны с пространственной характеристикой,
- система выбора наилучшего размещения техники, материальных объектов, военных гарнизонов,
- система мониторинга текущего состояния территорий на предмет возникновения ЧС,
- система координации деятельности функциональных и территориальных систем МЧС, мероприятий по ГО и защите населения и работе экстренных служб: милиции, пожарных, газовых служб [4].

9. *Human Resource.*

На успех любого проекта, прежде всего, влияет человеческий фактор. Поэтому управление и исследование различных факторов, влияющих на HR, необходимо для организации бизнеса:

- системы анализа демографической ситуации (такие показатели как возраст, пол, социальный класс населения могут повлиять на пенсионные, страховые предложения и оплату труда),
- система мониторинга квалификации и навыков сотрудников (может изменить политику и методы мотивации сотрудников),
- система анализа транспортных связей работодателем (разработка транспортных маршрутов для доставки сотрудников).

10. Туризм.

На сегодняшний день при росте численности населения и его благосостояния возрастает интерес к отдыху, поэтому оценка туристического потенциала территорий, планирование и развитие инфраструктуры, рекреации и туризма является очень актуальным вопросом:

- система анализа туристического потенциала территорий,
- система анализа и прогноза перспективных направлений развития туризма,
- система разработки туристических маршрутов.

Пользуясь данной классификацией, нужно учитывать, что многие ГИС-приложения можно отнести одновременно к нескольким классам. Это связано с тем, что ГИС очень гибкие системы, они объединяют разные предметные области, а иногда создаются для решения целого набора различных по характеру задач.

Данную классификацию можно использовать для решения различных практических задач. Например, на этапе проектирования ГИС-приложений разработчику необходимо подобрать набор исходных данных проекта. Требования к такому набору формируются исходя из задач и тематики разрабатываемой системы.

2. Определение исходного набора данных проекта на основе классификации

Данную классификацию можно использовать для выбора набора исходных данных.

Такая задача относится к классу задач принятия решений. Метод анализа иерархий позволит разработчику в интерактивном режиме выбрать вариант решения, который является наилучшей альтернативой. Процесс построения иерархической модели отражает способ мышления человека во время принятия решения.

Наборы данных – начальное множество альтернатив h , среди которых нужно провести ранжирование (выбор) по целевой функции F , которая

представляет собой набор частных критериев f_1, f_2, \dots, f_n . Критерии выбора данных могут быть такими: f_1 – качество набора данных, f_2 – скорость получения, f_3 – стоимость. В вершине иерархической модели – главная цель, «выбор набора данных», на втором уровне располагаются три критерия оценки качества альтернатив, приоритет которого $f = (f_1 + f_2 + f_3)$. На третьем – подкритерии качества: k_1 – разрешение, k_2 – тип съемки, k_3 – всепогодность, k_4 – оперативность. Тогда вектор приоритетов второго уровня $k = (k_1 + k_2 + k_3 + k_4)$.

При вычислении целевой функции необходимо определить значимость критериев, используя метод попарных сравнений. Приоритет критериев первого уровня задает пользователь, тем самым определяя стратегию выбора данных. Критерии второго уровня ранжируются исходя из первого, «аккумулируя» свойства выбранной стратегии. Приоритет критериев четвертого уровня $h = (h_1, \dots, h_n)$ – отражают свойства набора данных заявленные разработчиком, что позволит получить вектор приоритетов для третьего уровня иерархии с учетом веса $k_{1..4}$ каждого подкритерия качества. Сформируем из полученных векторов h матрицу H . Тогда вектор приоритетов верхнего уровня f ранжирует набор данных, с точки зрения выбранной пользователем стратегии и учетом подкритериев качества $f = H \cdot k$.

3. Построение иерархической модели выбора оптимального набора данных

Построим модель выбора данных для разработки ГИС-приложения в сфере безопасности, военного дела и разведки. Иерархическая модель данной системы (рис. 2) реализована в программном приложении «Mpriority». Данная диалоговая система предназначена для поддержки принятия решений в различных сферах человеческой деятельности. В ее основе лежит многокритериальный отбор рациональных альтернатив, кроме этого данное приложение «Mpriority» распространяется бесплатно и имеет русскоязычный интерфейс.

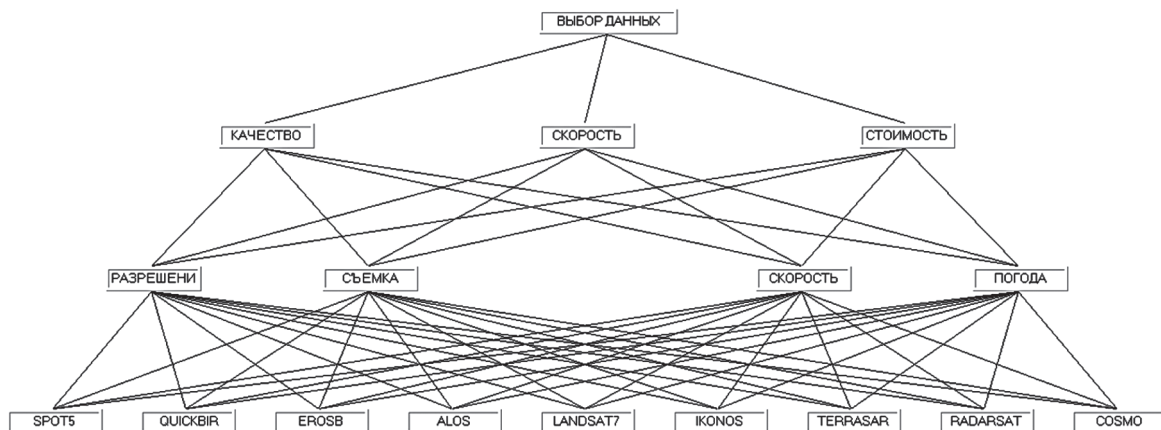


Рис. 2. Иерархическая модель системы принятия решений

Следующим этапом является попарное сравнение отдельных критериев иерархической модели, в результате чего формируется матрица попарных сравнений, по которой и производится вычисление целевой функции. Матрица попарных сравнений первого уровня относительно главной цели модели «Выбор данных» (рис.3) демонстрирует приоритет критериев качества «оперативность» и «качество» над критерием «цена».

		1.	2.	3.	
1.	КАЧЕСТВО	1	1	5	0,4352
2.	СКОРОСТЬ	1	1	7	0,4869
3.	СТОИМОСТЬ	1/5	1/7	1	0,0778

Additional values: 3,0125, 0,0062, 0,0108. Buttons: Применить, Закрывать, Исследовать, Отмена.

Рис. 3. Матрица попарных сравнений первого уровня

Каждая ячейка таблицы матрицы содержит результат попарного сравнения приоритета двух критериев между собой. В программном приложении «Mpriority» для заполнения матрицы используется качественная шкала (рис. 4).

Степень важности	Определение
1	A и B одинаково важны
3	A незначительно важнее B
5	A значительно важнее B
7	A явно важнее B
9	A по значимости абсолютно превосходит B
2, 4, 6, 8	промежуточные значения между соседними значениями шкалы

Рис. 4. Качественная шкала сравнений

После заполнения матриц сравнений второго и третьего уровней произведена математическая обработка данных иерархической модели и сделан выбор в пользу наилучшей альтернативы (рис. 5).

Полученные значения целевой функции демонстрируют явное преимущество таких типов данных, которые характеризуются всепогодностью и оперативностью получения. Для построения ГИС-приложений в сфере деятельности экстренных служб данные критерии являются наиболее важными, поэтому именно радарный тип съемки более предпочтительный. Однако не каждую задачу можно решить на основе радарных снимков, в этом случае, пользователь может выбрать спутники с комбинированным видом съемки.

Чем более полно будет сформулирована задача и требования к набору, тем точнее можно построить

модель иерархий и матрицы попарных сравнений. Что в свою очередь поможет разработчику сделать выбор в пользу наилучшей альтернативы, которая полностью согласуется с его пониманием целей и задач ГИС-приложения.

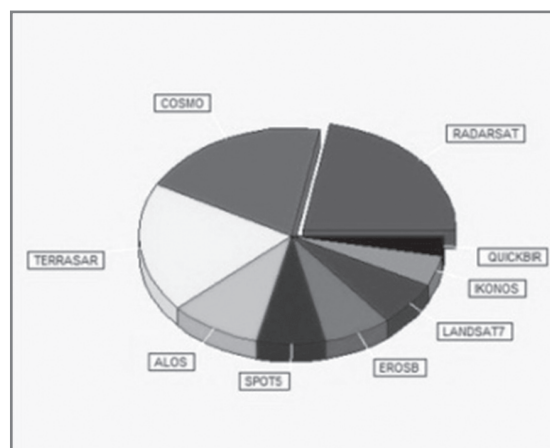


Рис. 5. Значение целевой функции

Выводы

ГИС-системы разрабатываются для решения научных и прикладных задач: мониторинг, планирование, проектирование, принятие решений и т.д. Такое многообразие и активное развитие ГИС технологий предполагает наличие современной классификации. Предложенную классификацию можно использовать для решения различных задач. В работе продемонстрирован метод выбора исходных данных для проекта с помощью иерархической модели. На основе предложенной классификации определены приоритеты критериев всех уровней и значения матрицы попарных сравнений. По значению целевой функции выбран оптимальный набор данных. Иерархическая модель построена в бесплатном программном приложении «Mpriority», в котором есть возможность актуализировать требования к критериям в зависимости от тематики проекта и добавлять новые наборы данных, увеличивая количество элементов на нижнем уровне. Данная модель поможет разработчику на первых этапах проектирования, ведь рынок гео-данных с каждым днем становится больше, а процесс выбора все более сложным и трудоемким. Даже при наличии подробной информации о снимках сложно спрогнозировать дальнейший метод обработки, необходимый тип съемки, разрешение и т.д., когда речь идет о первых этапах работы над проектом. Предложенный способ выбора набора данных основан на значениях приоритетов критериев, которые зависят только от тематики проекта. Разработчик может дополнять или изменять требования по ходу проекта. Наличие полного набора требований обеспечит выбор наилучшей альтернативы, которая полностью согласуется с целями и задачами ГИС-приложения.

Список литературы: 1. Турлапов, В. Е. Геоинформационные технологии в экономике. Учебно-методическое пособие [Текст] / В. Е. Турлапов. – Нижний Новгород: НФ ГУ-ВШЭ, 2007. – 118с. 2. Ищук, А. А. “G-Government” – мечта или ближайшее будущее Украины? [Текст] / А. А. Ищук // Ученые записки Таврического национального университета имени В. И. Вернадского. Серия «География». – 2008. – Т. 21, №1. – С. 29-38. 3. Самардак, А. С. Геоинформационные системы: Учебник. [текст] / А. С. Самардак – Владивосток: ДВГУ, 2005. – 124 с. 4. Самодумкин, С.А. Управление данными в геоинформационных системах: Учеб. пособие для студ. спец. «Искусственный интеллект» [Текст] / С. А. Самодумкин, М. Д. Степанова, Н. А. Гулякина – Мн.: БГУИР, 2006. – 111 с.

Поступила в редколлегию 11.01.2012

УДК 004.89:528:519.816

Аналітичний вибір початкових даних на основі методу ієрархій, враховуючи тематику ГІС-проекту / А.В. Бельчева, Н.О. Манакова // Біоніка інтелекту : наук.-техн. журнал. – 2012. – № 1 (78). – С. 97-101.

У статті запропоновано метод вибору вхідних даних ГІС-проекту на основі ієрархічної моделі. Користуючись класифікацією, визначено пріоритети критеріїв на всіх рівнях та значення коефіцієнтів матриці попарних порівнянь. Такий спосіб вибору даних заснований на значеннях пріоритетів критеріїв, які залежать тільки від тематики проекту.

Лл. б. Бібліогр.: 4 найм.

UDC 004.89:528:519.816

Analytic data selection based on hierarchy process by taking into account the GIS project topic / A.V. Belcheva, N.O. Manakova // Bionics of Intelligense: Sci. Mag. – 2012. – № 1 (78). –P. 97-101.

The method of selecting the initial data using a hierarchical model has been illustrated in the project. Based on the proposed classification the criteria priorities have been defined at all levels as well as the matrix values of the pairwise comparisons. Such method of selecting the data set is based on the priority criteria values which depend only on the project subject.

Fig. 6 Ref.: 4 items.

И.В. Глушаускайте¹, Т.Н. Заболотняя²¹НТУУ «КПИ», г. Киев, Украина, irina_glushauskaite@gmail.com²НТУУ «КПИ», г. Киев, Украина, tatiana104@yandex.ru

КОМБИНИРОВАННЫЙ МЕТОД АВТОМАТИЗИРОВАННОГО ОПРЕДЕЛЕНИЯ АВТОРСТВА ТЕКСТОВ

В данной статье предложен метод определения авторства текстов, который сочетает в себе классификацию и кластеризацию документов. Предложен алгоритм реализации метода, который может быть реализован на ЭВМ. Дана оценка эффективности предложенного метода. Обосновано улучшение численных характеристик качества классификации благодаря использованию предложенного комбинированного метода.

АВТОМАТИЗИРОВАННОЕ ОПРЕДЕЛЕНИЕ АВТОРСТВА, КЛАССИФИКАЦИЯ, КЛАСТЕРИЗАЦИЯ, КЛАСТЕРНЫЙ АНАЛИЗ, КЛАССИФИКАЦИЯ С УЧИТЕЛЕМ, КАЧЕСТВО КЛАССИФИКАЦИИ.

Введение

Определение авторства текста уже много лет является актуальной проблемой как в теоретической лингвистике, так и в области решения прикладных задач. Проведение анализа и сравнения стилей широко применяется в филологических дисциплинах, изучающих особенности организации текстов, в психологии и теории искусственного интеллекта, в судебной практике и криминалистике. Также реализация алгоритмов определения и сравнения стилей текстов широко используется при решении задач информационного поиска, распространенных в связи с увеличением объемов информации, в том числе электронной документации, и ростом сети Интернет.

Сегодняшняя актуальность решения данной проблемы обусловлена повсеместным переходом от рукописного текста к печатному (электронному). Если раньше в выявлении плагиата, определении автора литературного произведения или анонимного текста могла применяться почерковедческая экспертиза, то для печатного текста это невозможно. Автоматизированное определение авторства позволяет также избавиться от двух важных недостатков определения авторства экспертом: существенных временных затрат и возможной необъективности точки зрения эксперта.

На данный момент существует множество средств автоматического определения авторства текста, однако в большей их части под определением авторства понимается выбор автора некоторого текста из коллекции известных программистов авторов [3, 6]. Также широко применимы средства, которые могут разделить коллекцию текстов на несколько коллекций меньшего объема, содержащих похожие тексты, но не обязательно соответствующих определенным авторам [4, 11].

Выбор автора из списка авторов, известных программе, позволяет достаточно точно определять авторство в случае тестирования программы на текстах известных ей авторов, однако результаты работы программы на текстах неизвестных

авторов могут быть некорректными. Очевидным решением является снабжение программы очень большой коллекцией текстов разных авторов в качестве входных данных, однако это существенно замедляет работу программы и все равно не исключает возможности появления во входных данных программы текста автора, который программе не известен.

В качестве примера некорректного определения авторства текста можно привести ситуацию, в которой тексты неизвестных программистов распределяются по категориям уже существующих авторов. Следует отметить, что в случае позднего обнаружения подобной ошибки затраты на ее устранение могут оказаться критическими. Для иллюстрации приведем систему для компьютерной разметки большого и постоянно пополняющегося текстового корпуса, например – корпуса книг в большой библиотеке, которая автоматически определяет авторов поступающих в нее текстов. Для корректного поиска по корпусу важным критерием разметки является авторство каждого текста. Если данные обрабатываются неправильно, то тексты, авторы которых не были известны системе в начале ее работы, будут помечены как принадлежащие уже наличествующим авторам. Это сделает невозможным эффективный поиск в корпусе с использованием созданной разметки. Когда ошибка обнаружится, потребуется перезапускать систему и заново производить разметку корпуса, предварительно расширив множество известных системе обучающих текстов.

1. Постановка задачи

Для повышения вероятности корректной обработки текстов необходимо минимизировать возможность отнесения текста нового автора к категории уже существующего автора, а также ввести механизмы для обработки схожих между собой текстов, не принадлежащих известным программистам авторам.

Таким образом, **целью** данной работы стало повышение точности автоматизированного определения авторства текстов соответствующими

программными средствами за счет разработки нового метода обработки текстовых данных, которым предусматривается добавление текстов к категориям известных программ авторов и автоматическое выделение категорий для текстов авторов, неизвестных программе.

В данном исследовании не берутся во внимание такие характеристики текста, как его смысловая и коммуникативная структура, а также принадлежность текста к какому-либо литературному стилю. То, как эти и другие подобные характеристики влияют на качество автоматического анализа текста, описано в [8]. В данной же работе используются формальные показатели принадлежности текста тому или иному автору, прежде всего – набор употребляемых конкретным автором слов. Также рассматривается применение комбинированных признаков, выделение составных термов и стоп-слов, методов уменьшения пространства признаков и поиска наиболее информативных признаков.

В соответствии с поставленной целью **задачами исследования** являются:

- изучение методов обработки текстовых данных (в том числе методов классификации и кластеризации), применяемых для определения авторства последних;
- разработка нового метода определения авторства текстов для повышения эффективности работы соответствующего программного обеспечения по показателю корректности отнесения заданного текста к работам того или иного автора;
- формулировка алгоритма, реализующего предложенный метод;
- анализ результатов исследования эффективности нового метода определения авторства текстов.

2. Обзор методов определения авторства текстов

На данный момент существует множество разнообразных подходов к решению задачи определения авторства текстов. Большинство таких подходов базируется на использовании методов классификации [1, 3, 4] или кластеризации [6, 7] текстовых данных. Рассмотрим особенности применения этих методов к задаче определения авторства текстов, их преимущества и недостатки, различия, некоторые детали реализации. Цель данного этапа – определить, могут ли средства классификации и кластеризации быть применены к поставленной задаче, и найти специфические особенности применения.

2.1. Применение методов классификации к задаче определения авторства

Классификация документов, или классификация с учителем, — одна из задач информационного поиска, заключающаяся в отнесении документа к одной из нескольких заранее заданных категорий на основании содержания документа [5].

Автоматическая классификация может осуществляться на основе правил (созданных вручную) или с помощью обучения алгоритма по примерам.

Недостатки применения методов классификации для задачи определения авторства:

1. Первый недостаток всех методов классификации кроется в самом определении: количество и параметры категорий, на которые делятся тексты (в случае определения авторства – количество и список авторов), задаются заранее, и определяется принадлежность текстов только к этим определенным категориям. Таким образом, для любого, даже незначительного изменения списка авторов, необходимо выполнить существенный объем работы. В случае обучения по примерам требуется заново выполнить обучение классификатора, что требует значительных ресурсов (как вычислительных, так и временных). Для методов на основе правил требуется заново сформулировать правила, что требует работы экспертов. В случае сильных изменений категорий может потребоваться критически длительное время на обучение классификатора либо работу экспертов.

2. Невозможность автоматического определения новых категорий. В результате выполнения классификации новый текст будет отнесен к заданным заранее n категориям. Классификатор может оперировать вероятностями отнесения к категории, а значит, текст будет отнесен к n категориям с определенными вероятностями p_i , где $i = 1..n$ – индекс соответствующей категории; n , как и p_i , может быть равно 1. В случае, когда вероятность принадлежности текста к каждой категории ниже некоторого порогового значения, текст может быть не отнесен ни к одной категории (n может быть равно 0) – это зависит от реализации алгоритма. Но методы классификации не предусматривают возможности добавления новой категории в случае наличия большой коллекции похожих текстов во входных данных. Новые тексты могут быть отнесены к одной из существующих категорий, могут – к разным категориям, могут остаться некатегоризированными.

Преимущества применения методов классификации для задачи определения авторства. Безусловным плюсом применения методов классификации является стабильно высокое значение численных оценок качества классификации – к примеру, точности и F -меры, при низком значении ошибки классификации [8, 9]. Точность выше в случае использования классификации на основе задаваемых вручную правил, однако правила никогда не являются универсальными, и для добавления новых авторов правила требуется переписать. В связи с этим средства классификации на основе правил широко применяются в автоматической сортировке/фильтрации новостных сообщений (где категории практически неизменны), однако для задачи определения авторства (набор авторов каждый раз разный) эти средства малоприменимы.

Наибольшая эффективность методов классификации при условии, что наборы категорий могут отличаться, достигается за счет обучения алгоритма по примерам. В этом случае для корректного определения авторства необходимо, чтобы на вход алгоритма подавалась коллекция документов, для которых авторы уже известны. При обучении по примерам классификатору достаточно получить коллекцию документов и значений автора для этих категорий в качестве входных данных.

Численные характеристики качества классификации. Рассмотрим поднятый нами вопрос о численных оценках качества классификации. Приведем основные параметры, по которым производится оценка. Терминология приведена в соответствии с работой В.Г. Васильева и М.П. Кривенко «Методы автоматизированной обработки текстов» [9].

Пусть $C^0 = (c_{ij}^0)_{k \times n}$ – матрица эталонной классификации (может быть получена путем экспертной классификации объектов) размера $k \times n$ объектов из множества X по классам из множества W , где

$$c_{ij}^0 = \begin{cases} 1, x_j \in w_i \\ 0, x_j \notin w_i \end{cases} \quad (1)$$

$C = (c_{ij})_{k \times n}$ – матрица оцениваемой классификации (например, получена путем автоматической классификации объектов) размера $k \times n$, где

$$c_{ij} = \begin{cases} 1, x_j \in w_i \\ 0, x_j \notin w_i \end{cases} \quad (2)$$

Для оценки качества классификации обычно производится вычисление различных мер, характеризующих степень близости оцениваемой классификации объектов C к эталонной классификации C^0 .

При обработке текстовых данных наибольшее распространение получил подход, при котором сначала вводятся показатели качества классификации по отношению к отдельным классам, а затем на их основе уже строятся обобщенные показатели для всей совокупности классов.

Пусть зафиксирован некоторый класс

$$w_i, i = 1, \dots, k.$$

Результаты совместной классификации объектов x_1, \dots, x_n по отношению к данному классу, задаваемые с помощью матриц C^0 и C , можно представить в виде табл. 1.

Таблица 1

	T_i	F_i
T_i^0	n_{TT_i}	n_{TF_i}
F_i^0	n_{FT_i}	n_{FF_i}

В табл. 1 используются следующие сокращения: T_i – множество объектов, отнесенных к классу w_i в классификации C ; F_i – множество объектов, не отнесенных к классу w_i в классификации C ; T_i^0 и F_i^0 – множества объектов, которые отнесены и не

отнесены к классу w_i в классификации C^0 , соответственно:

$$n_{TT_i} = |T_i^0 \cap T_i|, n_{TF_i} = |T_i^0 \cap F_i|, n_{FT_i} = |F_i^0 \cap T_i|, \\ n_{FF_i} = |F_i^0 \cap F_i| \quad (3)$$

Наибольшее распространение получили следующие показатели:

Точность классификации (от англ. Precision)

$$P_i = \frac{n_{TT_i}}{n_{TT_i} + n_{FT_i}} \quad (4)$$

представляет собой процент объектов, правильно отнесенных к классу w_i в классификации C , по отношению к общему количеству объектов, отнесенных к классу w_i в классификации C .

Полнота классификации (от англ. Recall)

$$R_i = \frac{n_{TT_i}}{n_{TT_i} + n_{TF_i}} \quad (5)$$

представляет собой процент объектов, правильно отнесенных к классу w_i в классификации C , по отношению к общему количеству объектов, отнесенных к классу w_i в классификации C^0 .

Точность и полнота классификации характеризуют различные стороны оценки качества и их нельзя использовать независимо. Например, если все объекты отнести к классу w_i , то в этой ситуации полнота будет равна 1, а точность классификации очень низкой. Наоборот, если к классу w_i отнести только один правильный объект, то точность будет равна 1, а полнота будет близкой к 0.

Для возможности сравнения качества работы различных систем друг с другом удобно использовать один показатель, а не несколько. По этой причине при оценке качества классификации часто используется комбинированный показатель, который называется F -мера.

Он определяется следующим образом:

$$F_i(\lambda) = \left[\lambda \frac{1}{P_i} + (1-\lambda) \frac{1}{R_i} \right]^{-1}, \quad (6)$$

где $\lambda \in [0,1]$. Заметим, что при $\lambda = 0$ $F_i(0) = R_i$, при $\lambda = 1$ $F_i(1) = P_i$, при остальных значениях $\lambda \in (0,1)$ показатель $F_i(\lambda)$ является комбинацией точности и полноты. Обычно на практике используется значение $\lambda = 0,5$, в данном случае для F -меры используется обозначение $F_i \equiv F_i(0,5)$.

Помимо точности, полноты и F -меры также используется такой обобщенный показатель как ошибка классификации. Он определяется следующим образом:

$$E = \frac{1}{kn} \sum_{i=1}^k \sum_{j=1}^n |c_{ij}^0 - c_{ij}|. \quad (7)$$

2.2. Применение механизмов кластеризации для определения авторства.

Кластеризация документов, или классификация без учителя, — одна из задач информационного

поиска, целью которой является автоматическое выявление групп семантически похожих документов среди заданного фиксированного множества документов [5]. Следует отметить, что группы формируются только на основе попарной схожести описаний документов, и никакие характеристики этих групп не задаются заранее, в отличие от классификации документов, где категории задаются до начала выполнения алгоритма.

Средства кластеризации позволяют выделить подмножества похожих текстов из общего множества обучающих текстов. Количество подмножеств может быть как фиксированным, так и вычисляемым в зависимости от близости текстов. Безусловным плюсом применения средств кластеризации является то, что количество кластеров может меняться по ходу добавления в систему новых текстов. При пересчете кластеров на основании множества имеющихся в системе текстов новые кластеры могут добавляться к уже существующим, а те, в свою очередь, могут делиться на части или объединяться.

Кластеризация может быть применена для визуализации авторского стиля и схожести стилей авторов, для определения в творчестве автора группы текстов, несхожих с остальными текстами автора [11, 12].

Кластеризация применяется и для определения авторства, но в этом случае речь идет не об определении автора текстов из группы заданных авторов, а о выделении групп похожих текстов, которые могут принадлежать одному автору. Кластеризация в чистом виде не очень широко используется в определении авторства из-за невозможности явного (предварительного) задания категорий, как в случае классификации с учителем.

Кластерный анализ также может быть применен для определения принадлежности текста к уже существующим (явно заданным либо определенным классификатором) категориям: категории трактуются как кластеры, и новый документ добавляется в один из этих кластеров.

Однако какие преимущества дает нам применение кластерного анализа, если по определению сам алгоритм классификации занимается определением категорий новопривывших документов, к тому же применение алгоритма кластеризации само по себе не учитывает задание конкретных авторов для текстов? Ответ на этот вопрос будет рассмотрен далее.

3. Комбинированный метод определения авторства текста

Просуммировав вышесказанное, можем сделать выводы о том, что:

1. В большинстве случаев для автоматического определения авторства используются методы классификации.

2. Недостатки применения классификации — работа классификаторов только на заранее заданных категориях, невозможность автоматического задания новых категорий. Преимущество — высокое значение численных оценок качества.

3. Кластеризация применима для определения групп схожих по стилю текстов.

Проанализировав использование методов классификации и кластеризации, можем сделать вывод, что возможным путем добиться корректной обработки текстов, авторы которых заранее не заданы программе, является автоматическое введение новых категорий для авторов, неизвестных программе, с возможностью группировать похожие входные данные и причислять их к новой категории. Опишем, каким образом это реализуется и что позволило нам сделать такие выводы.

Из результатов анализа применения средств классификации видим, что для определения автора текстов эффективными и широко применимыми являются средства классификации, также известные как средства классификации с учителем. Однако методы классификации дают качественные и корректные результаты только в случае, если входные данные принадлежат к заранее заданным категориям, на которых обучался алгоритм. В случае определения авторства невозможным является снабжение алгоритма входными данными для обучения, которые включают тексты всех возможных авторов. В таком случае классификатор, получив на вход тексты неизвестных ему авторов, либо припишет эти тексты одному из существующих авторов, что является ошибкой и может повлечь ошибки в дальнейшем, либо же оставит текст некатегоризованным.

В случае, если классификатор оставляет часть текстов некатегоризованными, логичным является введение в его состав средств, производящих анализ этих текстов и их объединение в новые категории в случае выявления их схожести. В этом случае применимы средства кластеризации (классификации без учителя), которые дают возможность анализировать некатегоризованные тексты и выделять из них кластеры в случае схожести этих текстов.

Таким образом, для достижения поставленной цели данного исследования предлагается создать комбинированный метод определения авторства текста, в котором недостатки использованных методов классификации будут нивелированы путем введения элементов кластеризации.

Предлагаемый метод подразумевает выполнение следующих этапов:

1. Подготовительные этапы:
 - a. Выбор критериев классификации.
 - b. Выбор способа представления входных данных.
 - c. Подготовка обучающего множества текстов, для которых известны и заданы авторы.

d. Выбор метода классификации и его модификация: тексты могут быть не причислены ни к одной категории.

e. Обучение классификатора.

f. Выбор или разработка метода выделения кластеров среди группы текстов.

2. Основные этапы:

a. Проведение классификации группы текстов по категориям известных программе авторов.

b. Применение алгоритма кластеризации к текстам, не причисленным к известным авторам, и формирование новых категорий на основе результатов работы алгоритма кластеризации.

4. Обобщенный алгоритм реализации метода

В предыдущем пункте статьи был представлен новый метод определения авторства текстов. Сформулируем теперь обобщенный алгоритм определения авторства текста, который реализует предложенный метод.

Подготовительные этапы — действия, которые необходимо выполнить, для того чтобы алгоритм мог конкретно определять авторов текстов коллекции.

Основные этапы — это те действия, которые необходимо выполнить над текстом или коллекцией текстов для определения авторов.

4.1. Подготовительные этапы

Первый подготовительный этап — **выбор критериев классификации**. Критерии классификации — те свойства анализируемых документов, которые будут учитываться при анализе [2, 8, 10]. В качестве критериев классификации обычно принимается наличие в документах определенных термов. Однако в литературе существует много других подходов к определению критериев классификации [2]. Также стоит отметить, что для задачи определения авторства критерии классификации могут значительно отличаться от тех критериев, которые используются в общей задаче классификации.

Некоторые подходы к выбору критериев классификации:

1. Критериями классификации могут быть термы документов.

2. Для уменьшения размера пространства критериев и для повышения точности классификации можно пользоваться составными термами. Ими могут быть устойчивые словосочетания. Тогда имеет смысл пользоваться словарями для выявления таких словосочетаний. Также важными индикаторами авторского стиля являются свойственные конкретным авторам словосочетания. Однако создание обобщенного алгоритма выявления авторских словосочетаний представляется нетривиальной задачей.

3. Для уменьшения пространства термов имеет смысл не включать в число термов чересчур часто

используемые слова, а также стоп-слова. Примером часто используемых слов для русского языка являются слова «быть», «делать», «красивый»; наиболее часто используемые стоп-слова — предлоги, союзы, местоимения.

4. Именно для задачи определения автора естественно рассматривать лишь те характеристики текстов, значения которых, переходя от объекта к объекту, выявляют наибольшую изменчивость. То есть, помимо выделения стоп-слов рационально выполнить выделение слов, которые могут встречаться не очень часто, но частота их использования характеризует стиль определенного автора.

5. Вместо исходных, непосредственно измеренных, признаков целесообразно рассматривать меньшее количество новых признаков, которые являются производными от исходных. Создание комбинированных признаков может существенно уменьшить количество критериев классификации, что, в свою очередь, может сделать классификацию как более быстрой из-за небольшого количества критериев, так и более точной, потому что в созданных комбинированных признаках будут учитываться наиболее значимые признаки. На практике эффективные алгоритмы выделения комбинированных признаков недостаточно изучены, потому широко применяются эвристические методы.

6. Если стандартно в алгоритмах классификации имеет значение набор слов каждого текста и его соотношение с набором слов всех текстов, то для определения авторства существует много не менее важных критериев. Например, важными являются знаки препинания и частота их использования, порядок следования слов и предложений, длина предложения.

Выделению критериев классификации посвящено множество научных работ [2, 8, 10]. Наша задача — выбрать такие критерии, которые позволят классифицировать тексты с точностью, не ниже заданной, однако их введение не повлечет значительного усложнения алгоритма.

В качестве критериев классификации предлагается использовать термы документа. Также необходимо провести работу по уменьшению пространства термов — удаление стоп-слов и термов, частота использования которых в тексте выше некоторого экспериментально подобранного порогового значения. Кроме того, предлагается использование словаря, содержащего устойчивые словосочетания, с целью уменьшения пространства термов и увеличения точности классификации за счет использования составных термов. Таким образом, формируется m критериев классификации. Например, i -й критерий при $i = 1..m$ означает наличие в тексте некоторого простого либо составного терма.

Выбор критериев классификации обусловлен, прежде всего, требованиями к быстродействию: весь алгоритм определения авторства должен

выполняться быстро, что позволит оперативно произвести необходимую корректировку параметров. Усложнение критериев классификации замедлит процесс их формирования, но может увеличить точность результирующей классификации; уменьшение пространства термов способно ускорить процесс классификации, но может повлечь потерю точности.

Второй подготовительный этап алгоритма — **представление входных данных**. Этап подразумевает выполнение всех действий, которые необходимо выполнить над текстами естественного языка для преобразования их во входные данные алгоритма классификации. Входные данные должны отображать соответствие текстов критериям, о которых речь шла ранее.

Традиционно входными данными алгоритмов классификации является либо вектор термов документа, либо множество термов (так называемый “мешок слов”, *bag of words*) [8, 9]. Однако, как уже было сказано, может быть проведена работа по уменьшению пространства термов, по выделению многословных термов, по удалению стоп-слов и т.д., результатом которой будет выделение критериев классификации [2, 8, 9].

В рамках предлагаемого алгоритма остановимся на простейшем и наиболее широко применяемом случае: представление документа — вектор документа в пространстве выбранных на предыдущем этапе критериев классификации, значения элементов вектора нормированы. Например, находятся в диапазоне $[0, 1]$ — критерий в документе отсутствует, 1 — критерий присутствует либо частота его применения максимальна.

Для каждого текста его векторным представлением является вектор из m элементов, i -й элемент вектора означает наличие (частоту употребления) или отсутствие в тексте i -й характеристики, или же соответствие i -му критерию.

Третий подготовительный этап — **подготовка обучающего множества текстов**. Для построения классификатора, обучающегося на примерах, в первую очередь необходимо обеспечить наличие коллекции текстов для обучения классификатора. Коллекция должна быть достаточно обширной, и для всех текстов этой коллекции должны быть заданы авторы. Это позволяет достичь двух целей: дает алгоритму возможность собрать сведения про стиль каждого автора, а также дает возможность автоматического тестирования классификатора.

Четвертый подготовительный этап — **выбор метода классификации**. В классификации и кластеризации для определения схожести документов применяются некоторые метрики. Для примера рассмотрим простейшую метрику: близость между текстами — это расстояние между векторными представлениями этих текстов в пространстве выбранных критериев классификации. Чем меньше

это значение, тем более близкими друг к другу мы считаем тексты, а для близких текстов можно говорить об одинаковом авторе. Еще раз подчеркнем, что перед нами стоит задача выделения именно таких характеристик текстов, которые были бы существенными для определения авторства.

В действующих системах классификации для определения схожести документов применяется ряд стандартных методов: например, Байесовский метод классификации (*Naive Bayes classifier*), метод опорных векторов (*Support Vector Machines*), деревья принятия решений (*Decision Trees*), k -ближайших соседей (*k-Nearest Neighbors*) и т.д.

Для реализации этого этапа в алгоритме предлагается использование метода опорных векторов [9], поскольку этот метод имеет большую точность и легко реализуется программно. Метод работает с векторными представлениями текстов, однако определение близости текстов происходит сложнее, чем в примере, описанном выше.

Пятый подготовительный этап — **обучение классификатора**. Алгоритмом обучения классификатора предусматривается деление всего множества входных данных (заранее отобранных текстов с заданными авторами) на обучающее и проверочное множества текстов. Алгоритм обучается с помощью части наличествующих у нас текстов — обучающего множества. В качестве входных данных алгоритм получает векторные представления текстов и информацию об их принадлежности авторам; на основании этих данных формируются правила отнесения документов к определенному автору. Далее алгоритм тестируется (экспертом либо же программно) на проверочном множестве текстов. Отслеживается, правильно ли тексты проверочного множества были отнесены к категориям, в случае большого числа случаев неправильного отнесения алгоритм классификации может быть пересмотрен. Например, могут быть изменены некоторые коэффициенты.

После выполнения этих действий классификатор способен получать на вход новые тексты (в соответствующем представлении) и относить их к какой-либо из заранее заданных рубрик с определенной вероятностью.

Ясно, что в задаче определения авторства алгоритм не должен относить документ более чем к одной категории; введение возможности соавторства чрезмерно усложнит алгоритм. Однако нетрудно ввести в алгоритм модификацию: если документ не относится к числу документов обучающей выборки, и программа относит его к каждой категории с вероятностью менее какого-то порогового значения, документу присваивается “неопределенная” категория. Суть этой модификации в том, что каждый документ теперь не обязан быть отнесен к какой-то категории. Добавляются в существующие категории те документы, которые

явно (то есть, с достаточно большой вероятностью) принадлежат к этим категориям. Те же документы, принадлежность которых к какой-либо категории спорна, остаются в “неопределенной” категории. Они могут быть категоризированы на следующем этапе работы алгоритма. Пороговое значение следует подбирать экспериментально, оно будет зависеть от выбранных критериев классификации и схожести. Для определения принадлежности документа к категории используются критерии схожести, определенные на подготовительном этапе алгоритма.

И, наконец, последний, шестой подготовительный этап – **выбор метода выделения кластеров**. Существует множество алгоритмов кластеризации текстовых данных [8, 9]. На вход алгоритма поступает коллекция текстов, результатом работы алгоритма является эта коллекция с выделенными кластерами. В общем случае алгоритм кластеризации выполняет разделение некоторого множества документов на заранее не заданные категории, анализируя содержимое документов. Количество категорий может быть как фиксированным, так и определяемым алгоритмом. Подбирается метод кластеризации, работающий с нефиксированным количеством кластера, с возможностью задания минимального размера кластера.

4.2. Основные этапы алгоритма определения авторства текста

Проведение классификации группы текстов по категориям известных программ авторов. После проведения подготовительных этапов обученный классификатор может определять автора определенного текста. На этом этапе на вход алгоритма подается группа текстов, на выходе части текста присвоены категории известных программ авторов, часть текстов остается не категоризированной. Из не категоризированных текстов могут быть выделены новые категории в ходе выполнения следующего этапа алгоритма.

Применение алгоритма кластеризации к текстам, не причисленным к известным авторам, и формирование новых категорий.

Для обработки документов, которым не была присвоена категория в процессе классификации, используется механизм кластеризации.

В нашем случае алгоритм определяет схожесть документов по критериям схожести, подобным тем, которыми мы воспользовались при классификации (можно использовать как тот же набор критериев схожести, так и некоторый другой). Кластеризация выполняется после классификации, ее цель – выделить новые категории из документов, которые не могли быть отнесены к существующим категориям из-за недостаточной близости к ним. Нет нужды выполнять кластеризацию после классификации (успешной или нет) каждого документа.

Кластеризация может выполняться после подачи n документов на вход классификатора, после определенного промежутка времени, при наличии определенного количества документов с “не определенной” категорией.

На вход алгоритма кластеризации поступают документы с “не определенной” категорией. Алгоритм анализирует близость документов друг к другу с точки зрения критериев схожести, и создает новые категории в случае нахождения групп близких документов (документов, схожесть которых выше некоторого порогового значения).

Два основных вопроса в данном случае – критерии схожести документов: набор используемых признаков, размер создаваемых категорий. Например, может случиться так, что один текст одного писателя будет очень похож на один текст другого. В случае если каждый из этих текстов будет определен как “не похожий” на остальные тексты писателя, можно ли говорить о том, что эти тексты написаны одним человеком? Очевидным кажется решение ввести минимальный размер создаваемой категории: например, можно говорить о появлении нового автора, если для не менее чем k (например, при $k = 10, 20, 100$) текстов, которые классификатор пометил как тексты неопределенной категории, их попарная близость выше заданного ранее порогового значения. В таком случае этим документам присвоится новая категория, и на следующем шаге алгоритм классификации будет учитывать также эту категорию при категоризации новых документов, поступающих в алгоритм.

Для настройки поведения алгоритма выполняется тестирование. **Тестирование** алгоритма включает наблюдение эксперта за ходом выполнения алгоритма и корректировку параметров алгоритма в случае необходимости. Как при работе алгоритма, так и при тестировании эксперт может подтвердить, что новая категория была создана корректно, и назвать ее именем автора этих текстов. В случае некорректного определения новых категорий следует изменить пороговое значение, критерии схожести документов или признаки, по которым мы производим классификацию. Тестирование подразумевает выполнение алгоритма на некоторых входных данных и корректировку параметров алгоритма – критериев для классификации и кластеризации, размерности пространства критериев, размера создаваемых рубрик, метрик схожести. Также в рамках тестирования оценивается корректность работы алгоритма по некоторым заранее заданным критериям.

5. Исследование эффективности комбинированного метода определения авторства текстов

Алгоритм позволяет повысить значение параметров качества классификации, например точности, полноты, F -меры, и снизить значение

ошибки классификации по сравнению с применением стандартных алгоритмов к задаче определения авторства. Рассмотрим значение численных характеристик качества классификации для предложенного алгоритма.

В табл. 1 были рассмотрены характеристики для оценки качества алгоритмов классификации. Эти характеристики обозначены как n_{TT_i} , n_{TF_i} , n_{FT_i} , n_{FF_i} . В дальнейшем будем считать, что это оценки для алгоритма определения авторства, использующего средства классификации без добавления средств кластеризации.

На данном этапе то, какой алгоритм классификации мы рассматриваем, не принципиально, поскольку мы ставим цель показать улучшение значений оценки качества вследствие введения кластеризации в рассмотренный алгоритм. Будем считать, что алгоритм классификации классифицирует каждый текст из входных текстов, и присваивает ему одно значение автора. Это верно для преимущественного большинства случаев применения алгоритмов классификации для определения авторства [1, 3, 4]

Введем аналогичные рассмотренным ранее обозначения для созданного нами алгоритма: $n_{TT_i}^*$, $n_{TF_i}^*$, $n_{FT_i}^*$, $n_{FF_i}^*$, а также производные от них P_i^* , R_i^* , F_i^* , E_i^* . Параметры P_i^* , R_i^* , $F_i(\lambda)^*$ характеризуют категорию, для характеристики всего классификатора в простейшем случае используется среднее арифметическое параметров категорий.

Рассмотрим значения этих параметров и изменение численных характеристик качества классификации:

1. Для категорий текстов, авторы которых известны программе в начале ее работы.

Введение средств кластеризации позволяет уменьшить число текстов, некорректно отнесенных к категории какого-то автора. В случае использования только средств классификации в категорию, соответствующую определенному автору, попадут и тексты, действительно принадлежащие автору, и часть текстов, которые принадлежат не известным программе авторам и были ошибочно классифицированы. Используя символьные обозначения,

$$n_{FT_i} > n_{FT_i}^* . \quad (8)$$

Это изменение улучшает значение точности и F -меры. Подставив в формулы (4) и (6) нововведенные обозначения и беря во внимание свойство (8), имеем:

$$P_i = \frac{n_{TT_i}}{n_{TT_i} + n_{FT_i}} < \frac{n_{TT_i}^*}{n_{TT_i}^* + n_{FT_i}^*} = P_i^* ; \quad (9)$$

$$F_i \equiv F_i(0,5) = \left[\frac{1}{2^* P_i} + \frac{1}{2^* R_i} \right]^{-1} < \left[\frac{1}{2^* P_i^*} + \frac{1}{2^* R_i^*} \right]^{-1} \\ F_i^* \equiv F_i(0,5)^* . \quad (10)$$

Видим, что введение методов кластеризации позволило увеличить значение точности и F -меры для категорий, автора которых уже известны программе.

2. Для категорий текстов, авторы которых уже известны программе в начале ее работы.

Без применения средств кластеризации n_{TT_i} для всех категорий авторов, заранее не известных программе, было равно 0. Текст не мог быть отнесен к корректной категории, поскольку категория не существовала в начале работы программы и не могла быть создана позднее.

Представленный алгоритм выделяет категории для новых авторов, и часть категорий выделяется корректно. То есть, некоторое число текстов, присвоенных этой категории, в действительности принадлежат одному автору, и новосозданная категория соответствует этому автору. Для этих категорий n_{TT_i} станет больше 0. В таком случае значение оценок классификации, имеющих n_{TT_i} в числителе, возрастет; значение точности, полноты и F -меры станет ненулевым. Для некорректно введенных категорий n_{TT_i} останется равным 0, что не повредит результирующим оценкам качества классификации.

Подставим в формулы (4), (5), (6) характеристики алгоритма классификации и созданного нами алгоритма:

$$P_i = \frac{n_{TT_i}}{n_{TT_i} + n_{FT_i}} = \frac{0}{n_{FT_i}} < \frac{n_{TT_i}^*}{n_{TT_i}^* + n_{FT_i}^*} = P_i^* ; \quad (11)$$

$$P_i = \frac{n_{TT_i}}{n_{TT_i} + n_{TF_i}} = \frac{0}{n_{TF_i}} < \frac{n_{TT_i}^*}{n_{TT_i}^* + n_{TF_i}^*} = R_i^* ; \quad (12)$$

$$F_i \equiv F_i(0,5) = \left[\frac{1}{2^* P_i} + \frac{1}{2^* R_i} \right]^{-1} = 0 ; \quad (13)$$

$$F_i^* \equiv F_i(0,5)^* = \left[\frac{1}{2^* P_i^*} + \frac{1}{2^* R_i^*} \right]^{-1} . \quad (14)$$

Сознательно избегаем деления на 0, потому считаем, что $F_i = 0$, а не ∞ .

Видим, что введение методов кластеризации позволило увеличить ранее равные 0 значения точности, полноты и F -меры для корректно выделенных средствами кластеризации категорий.

Просуммировав вышесказанное, видим, что численные оценки улучшаются, поскольку минимизируется вероятность отнесения текста неизвестного автора к некорректной категории, соответствующей уже известному программе автору. Кроме того, с большой вероятностью средства кластеризации корректно создадут категорию и внесут туда текст, даже если он принадлежит автору, заранее не известному программе. Для стандартных же средств классификации такая вероятность равна нулю, и если автор не известен программе, то его

текст точно не будет отнесен к корректной категории. Из этого делаем вывод, что введение средств кластеризации позволяет улучшить значение численных характеристик качества классификации.

Выводы

В данной статье предложен метод определения авторства текстов, который сочетает в себе классификацию и кластеризацию документов. Данный метод эффективен в случае его использования в системе для установления факта авторства текстов с возможностью определения автора из заданных и автоматического добавления категорий для новых авторов. Первоначально тексты, представляющие входные данные, рассортированы по заданным категориям, но в дальнейшем при добавлении новых текстов средства кластеризации позволяют объединять добавленные тексты в новые категории. Нововведенные категории соответствуют группам текстов авторов, которых не было в первоначальном списке авторов. Этот подход рассмотрен подробнее, также рассмотрены методы представления данных и выбора информативных признаков документа, которые позволяют точнее задавать новые категории. Проведено исследование эффективности созданного метода, которое показало, что метод позволяет достичь улучшения численных оценок качества классификации.

Предложен алгоритм реализации метода определения авторства с использованием средств классификации и кластеризации, который может быть реализован на ЭВМ.

Дальнейшего рассмотрения требуют отдельные параметры классификации. Также планируется создать программную реализацию алгоритма для корректировки параметров классификации и проведения тщательного тестирования.

Список литературы. 1. *Агеев, М.С.* Методы автоматической рубрикации текстов, основанные на машинном обучении и знаниях экспертов [Текст] : дис. ... канд. физ.-мат. наук : 05.13.11 / М.С. Агеев. — М., 2004. — 136 с. 2. Автоматическое определение авторства [Электронный ресурс] : / Л. М. Пивоварова // Режим доступа: <http://www.slideshare.net/lmp/09-9770049> - 20.01.2012 г. — Загл. с экрана. 3. *Романов, А.С.* Методика и программный комплекс для идентификации автора неизвестного текста [Текст] : дис. ... канд. техн. наук : 05.13.18 / А.С. Романов. — Томск, 2009. — 149 с. 4. *Шевелев, О.Г.* Разработка и исследование алгоритмов сравнения стилей текстовых произведений [Текст] : дис. ... канд. техн. наук : 05.13.18 / О.Г. Шевелев. — Томск, 2006. — 176 с. 5. *Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze.* An Introduction to Information Retrieval Draft. Online edition. — Cambridge University Press, 2009. — 544 p.

6. *Сидоров, Ю.В.* Математическая и информационная поддержка методов обработки литературных текстов на основе формально-грамматических параметров [Текст] : дис. ... канд. техн. наук : 05.13.18 / Ю.В. Сидоров Юрий Владимирович. — Петрозаводск, 2002. — 127 с. 7. *Грушников, А.В.* Аутентификация произведений живописи по цифровым изображениям [Текст] / А.В. Грушников // Всероссийский журнал научных публикаций. — 2011. — Июнь. — С. 5-7. 8. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика [Текст] : учеб. пособие / Е.И. Большакова, Э.С. Клышинский, Д.В. Ландэ [и др.]. — М.: МИЭМ, 2011. — 272 с. 9. *Васильев, В.Г.* Методы автоматизированной обработки текстов [Текст] / В. Г. Васильев, М. П. Кривенко. — М. :ИПИ РАН, 2008. — 305 с. 10. *Стадник, А.В.* Использование искусственных нейронных сетей и вейвлет-анализа для повышения эффективности в задачах распознавания и классификации [Текст] : дис. ... канд. техн. наук : 05.13.18 / А.В. Стадник. — Иваново, 2004. — 88 с. 11. *Шмулевич, М.М.* Метод автоматической кластеризации текстов, основанный на извлечении из текстов имен объектов и последующем построении графов совместной встречаемости ключевых термов [Текст] : дис. ... канд. физ.-мат. наук : 05.13.17 / М.М. Шмулевич. — М., 2009. — 120 с. 12. Обзор методов кластеризации текстовой информации [Электронный ресурс] : / К. М. Кириченко, М. Б. Герасимов // Режим доступа http://www.dialog-21.ru/Archive/2001/volume2/2_26.htm - 20.01.2012 г. — Загл. с экрана.

Поступила в редколлегию 18.11.2011

УДК 004.912

Комбінований метод автоматичного визначення авторства текстів / І.В. Глушаускайте, Т.М. Заболотня // Біоніка інтелекту : наук.-техн. журнал. — 2012. — № 1 (78). — С. 102-110.

У даній роботі запропоновано метод автоматизованого визначення авторства текстів, який поєднує в собі застосування засобів класифікації та кластеризації. Розроблений метод дає можливість визначення тих авторів текстів, які не були знайдені після проведення класифікації. Подано узагальнений алгоритм реалізації методу, який може бути реалізований програмно. Дана оцінка ефективності запропонованого методу.

Табл. 1. Бібліогр. : 12 найм.

UDC 004. 912

The combined method of automatic text authorship determination / I. Glushauskaite, T. Zabolotnyaya // Bionics of Intelligense: Sci. Mag. — 2012. — № 1 (78). — P. 102-110.

In this work a method of automated text authorship determination which combines implementations of both the means of classification and clustering is proposed. The developed method enables attribution of the texts that couldn't be attributed to some authors during classification. A generalized algorithm for implementation of the method, which can be implemented programmatically, is presented. An estimation of the efficiency of the proposed method is given.

Tab. 01. Ref.: 12 items.

УДК 519.62

Т.М. Федорова¹¹ХНУРЕ м. Харків, Україна, tanja_fedorova@mail.ru

ПОБУДОВА ЛОГІЧНОЇ МЕРЕЖІ ДЛЯ ФЛЕКТИВНОЇ ОБРОБКИ ДІЄСЛІВ УКРАЇНСЬКОЇ МОВИ

Стаття присвячена побудові моделі флективної обробки логічною мережею для дієслів української мови. Вона характеризується системою бінарних відношень та формулами відповідних предикатів. Мережа призначена для рішення системи рівнянь, що задаються відповідною моделлю.

МОВА, ДІЄСЛОВО, ГРАМАТИЧНІ ОЗНАКИ, ПАРАГМАТИЧНА ТАБЛИЦЯ, ДИЗ'ЮНКЦІЯ, БІНАРИЗАЦІЯ, ВІДНОШЕННЯ, ЛОГІЧНА МЕРЕЖА

Вступ

З усіх живих істот лише людина володіє мовою. Мова – основа мислення, інтелектуальна діяльність без мови неможлива. Прояви мови вкрай складні і різноманітні, їх вивчають багато наукових дисциплін – морфологія, синтаксис, лексикологія, семіотика, стилістика та ін. Обираємо для математичного опису морфологію мови, тобто галузь науки про граматику окремо взятого слова.

До числа важливих проблем теорії інтелекту відноситься моделювання природної мови людини. Повна формалізація мовних процесів забезпечить нічим не ускладнений зв'язок між людським і штучним інтелектами, створить необхідні передумови для повної автоматизації програмування, істотно підвищить рівень удосконалення систем штучного інтелекту.

Потреби практики зумовлюють напрям і темп еволюції комп'ютерних інформаційних систем. Бурхливий, некерований і часто непередбачуваний прогрес веб-простору надав зазначеному розвитку певних рис, які вже дозволяють кваліфікувати четверте покоління інформаційних систем. Четверте покоління інформаційних систем – мовно-інформаційні системи ми пов'язуємо із застосуванням механізмів природної мови. Формальною основою цього підходу слугують лексикографічні системи та їх узагальнення (лексикографічні середовища, лексикографічні числення та лінгвістичні системи). Мовно-інформаційні системи є інтелектуально орієнтованими ab initio. Адже, як відзначалося в [1], «штучний інтелект є формою індивідуалізації технічних систем, якій притаманний мовний статус». Вони, очевидно, є інструментом комунікації.

Лінгвістичні праці містять неформалізовані описи морфологічних явищ, які непросто перевести на математичну мову. Ці описи спрямовані на розуміння людиною. Людина здатна розуміти їх «з півслова». Крім того, людина, що не приступала до вивчення морфології, фактично уже володіє мовою, морфологічні явища їй знайомі з власної мовної практики. Математичні описи морфології мови адресовані обчислювальній машині, вони повинні

бути зрозумілі їй. Однак рівень «інтелекту» обчислювальної машини в порівнянні з людським надзвичайно малий. Обчислювальна машина в сутності мовних здібностей – це «чиста дошка», ніякими мовних навичками вона не володіє. Це веде до того, що неформалізовані описи, які містяться в посібниках з морфології мови, обчислювальній машині зовсім незрозумілі, недоступні. До цих описів необхідно додати ще багато формалізмів, перш ніж машина зможе їх сприйняти й освоїти [2].

Задача полягає в тому, щоб розробити такі методи побудови логічних мереж, щоб ці мережі автоматично вирішували будь-які системи рівнянь алгебри предикатів так само безвідмовно, як це може робити людина, оперуючи формулами алгебри предикатів. Вирішуючи завдання, логічна мережа копіює дії людини з тією лише різницею, що людина при цьому діє послідовно, а мережа – паралельно [3].

Побудова моделі флективної обробки та логічної мережі для дієслів української мови

Вихідним матеріалом для моделювання режимів опрацювання слів на фонетичному та морфологічному рівнях природної мови є лінгвістичні закономірності. Математичним апаратом моделювання є алгебра скінченних предикатів [4].

Під час побудови математичної моделі флективної обробки мови описується фрагментне морфологічне відношення $L(X, Y)$, що відображає зв'язок між змістом X та змінним фрагментом тексту Y . У якості фрагменту, із усіх морфем слова (префікс, корінь, суфікс, закінчення, постфікс), вибрано закінчення. Основним типом граматичних афіксів в українській мові є флексія (закінчення), що відображає відношення слова до інших слів у словосполученні та реченні. $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ – набір граматичних ознак, які характеризують вплив дальнього тексту на закінчення; $Z = \{z_1, z_2, z_3\}$ – дієслівне закінчення, яке складається з першої, другої, третьої букв – z_1, z_2 и z_3 .

Позначення граматичних ознак та їх значення наступні: x_1 – число зі значеннями одина (о),

множина (м); x_2 – особа зі значеннями перша (1), друга (2), третя (3); x_3 – час зі значеннями минулий (м), неминулий (н); x_4 – рід зі значеннями чоловічий (ч), жіночий (ж), середній (с); x_5 – репрезентація зі значеннями особова форма (о), інфінітив (і); x_6 – спосіб зі значеннями дійсний (д), наказовий (н); x_7 – дієвідміна зі значеннями перша (1), друга (2).

В табл. 1, яка називається парадигматичною, представлені закінчення дієслів української мови.

Таблиця 1

Парадигматична таблиця

			x_5^o				x_5^i	
			x_6^d			$x_3^{T,MB} x_6^H$		
			$x_3^{T,MB}$	x_3^M				
				x_4^ch	x_4^zh	x_4^c		
x_1^o	x_2^1	x_7^1	- у	- в	- ла	- ло	X	- ти
		x_7^2	- ю					- ть
	x_2^2	x_7^1	- еш	- в	- ла	- ло	- и	- ти
		x_7^2	- еш				- Ø	- ть
	x_2^3	x_7^1	- е	- в	- ла	- ло	X	- ти
		x_7^2	- є					- ть
x_1^M	x_2^1	x_7^1	- емо	- ли	- ли	- ли	- ім	- ти
		x_7^2	- ємо				- імо	- ть
	x_2^2	x_7^1	- ете	- ли	- ли	- ли	- іть	- ти
		x_7^2	- єте				- те	- ть
	x_2^3	x_7^1	- уть	- ли	- ли	- ли	X	- ти
		x_7^2	- ють					- ть
	x_2^4	x_7^1	- ять				X	
		x_7^2	- ать					

Контекст, в який включено дієслово, впливає на форму слова граматичними ознаками $x_1 - x_7$. Вплив контексту на закінчення слова визначається лише значеннями граматичних ознак. Вплив контексту не може бути більше, ніж елементів парадигматичної таблиці. Нумеруємо осередки парадигматичної таблиці. Отримані номери осередків надалі будуть використані для позначення впливів контексту. Для кожної осередку (елементу) парадигматичної таблиці вводимо свій номер q (табл. 2).

Таблиця 2

			x_5^o				x_5^i	
			x_6^d			$x_3^{T,MB} x_6^H$		
			$x_3^{T,MB}$	x_3^M				
				x_4^ch	x_4^zh	x_4^c		
x_1^o	x_2^1	x_7^1	1	12	18	24	X	33
		x_7^2						
	x_2^2	x_7^1	2	13	19	25	30	34
x_1^M	x_2^1	x_7^1	4	14	20	26	X	35
		x_7^2	5					
	x_2^2	x_7^1	6	15	21	27	31	36
x_1^o	x_2^1	x_7^1	8	16	22	28	32	37
		x_7^2	9					
	x_2^2	x_7^1	10	17	23	29	X	38
x_7^2	11							

Виражаємо номери q осередків парадигматичної таблиці через ознаки $x_1 - x_7$:

$$\begin{aligned}
 x_1^o x_2^1 x_3^H x_5^o x_6^d &= q^1; & x_1^o x_2^2 x_3^H x_5^o x_6^d x_7^1 &= q^2; \\
 x_1^o x_2^2 x_3^H x_5^o x_6^d x_7^2 &= q^3; & x_1^o x_2^3 x_3^H x_5^o x_6^d x_7^1 &= q^4; \\
 x_1^o x_2^3 x_3^H x_5^o x_6^d x_7^2 &= q^5; & x_1^M x_2^1 x_3^H x_5^o x_6^d x_7^1 &= q^6; \\
 x_1^M x_2^1 x_3^H x_5^o x_6^d x_7^2 &= q^7; & x_1^o x_2^2 x_3^H x_5^o x_6^d x_7^1 &= q^8; \\
 x_1^M x_2^2 x_3^H x_5^o x_6^d x_7^1 &= q^9; & x_1^M x_2^3 x_3^H x_5^o x_6^d x_7^1 &= q^{10}; \\
 x_1^M x_2^3 x_3^H x_5^o x_6^d x_7^2 &= q^{11}; & x_1^o x_2^1 x_3^M x_4^ch x_5^o x_6^d &= q^{12}; \\
 x_1^o x_2^2 x_3^M x_4^ch x_5^o x_6^d &= q^{13}; & x_1^o x_2^2 x_3^M x_4^ch x_5^o x_6^d &= q^{14}; \\
 x_1^M x_2^1 x_3^M x_4^ch x_5^o x_6^d &= q^{15}; & x_1^M x_2^2 x_3^M x_4^ch x_5^o x_6^d &= q^{16}; \\
 x_1^M x_2^3 x_3^M x_4^ch x_5^o x_6^d &= q^{17}; & x_1^o x_2^1 x_3^M x_4^zh x_5^o x_6^d &= q^{18}; \\
 x_1^o x_2^2 x_3^M x_4^zh x_5^o x_6^d &= q^{19}; & x_1^o x_2^3 x_3^M x_4^zh x_5^o x_6^d &= q^{20}; \\
 x_1^M x_2^1 x_3^M x_4^zh x_5^o x_6^d &= q^{21}; & x_1^M x_2^2 x_3^M x_4^zh x_5^o x_6^d &= q^{22}; \\
 x_1^M x_2^3 x_3^M x_4^zh x_5^o x_6^d &= q^{23}; & x_1^o x_2^1 x_3^M x_4^c x_5^o x_6^d &= q^{24}; \\
 x_1^o x_2^2 x_3^M x_4^c x_5^o x_6^d &= q^{25}; & x_1^o x_2^2 x_3^M x_4^c x_5^o x_6^d &= q^{26}; \\
 x_1^M x_2^1 x_3^M x_4^c x_5^o x_6^d &= q^{27}; & x_1^M x_2^2 x_3^M x_4^c x_5^o x_6^d &= q^{28}; \\
 x_1^M x_2^3 x_3^M x_4^c x_5^o x_6^d &= q^{29}; & x_1^o x_2^2 x_3^H x_5^o x_6^H &= q^{30}; \\
 x_1^M x_2^1 x_3^H x_5^o x_6^H &= q^{31}; & x_1^M x_2^2 x_3^H x_5^o x_6^H &= q^{32}; \\
 x_1^o x_2^1 x_5^i &= q^{33}; & x_1^o x_2^2 x_5^i &= q^{34}; & x_1^o x_2^3 x_5^i &= q^{35}; \\
 x_1^M x_2^1 x_5^i &= q^{36}; & x_1^M x_2^2 x_5^i &= q^{37}; & x_1^M x_2^3 x_5^i &= q^{38}.
 \end{aligned}$$

Таблиця 3

Виконуємо операцію по членам диз'юнкції можливо більшого числа родинних рівностей:

$$\begin{aligned}
 &x_1^0 x_2^1 x_3^H x_5^0 x_6^L = q^1; & x_1^0 x_2^2 x_3^H x_5^0 x_6^L x_7^1 = q^2; \\
 &x_1^0 x_2^2 x_3^H x_5^0 x_6^L x_7^2 = q^3; & x_1^0 x_2^3 x_3^H x_5^0 x_6^L x_7^1 = q^4; \\
 &x_1^0 x_2^3 x_3^H x_5^0 x_6^L x_7^2 = q^5; & x_1^M x_2^1 x_3^H x_5^0 x_6^L x_7^1 = q^6; \\
 &x_1^M x_2^1 x_3^H x_5^0 x_6^L x_7^2 = q^7; & x_1^M x_2^2 x_3^H x_5^0 x_6^L x_7^1 = q^8; \\
 &x_1^M x_2^2 x_3^H x_5^0 x_6^L x_7^2 = q^9; & x_1^M x_2^3 x_3^H x_5^0 x_6^L x_7^1 = q^{10}; \\
 &x_1^M x_2^3 x_3^H x_5^0 x_6^L x_7^2 = q^{11}; \\
 &x_1^0 x_3^M x_4^H x_5^0 x_6^L (x_2^1 \vee x_2^2 \vee x_2^3) = q^{12} \vee q^{13} \vee q^{14}; \\
 &x_1^M x_3^M x_4^H x_5^0 x_6^L (x_2^1 x_4^H \vee x_2^2 x_4^H \vee x_2^3 x_4^H \vee x_2^1 x_4^J \vee x_2^2 x_4^J \vee \\
 &\vee x_2^3 x_4^J \vee x_2^1 x_4^C \vee x_2^2 x_4^C \vee x_2^3 x_4^C) = q^{15} \vee q^{16} \vee q^{17} \vee \\
 &\vee q^{21} \vee q^{22} \vee q^{23} \vee q^{27} \vee q^{28} \vee q^{29}; \\
 &x_1^0 x_3^M x_4^J x_5^0 x_6^L (x_2^1 \vee x_2^2 \vee x_2^3) = q^{18} \vee q^{19} \vee q^{20}; \\
 &x_1^0 x_3^M x_4^C x_5^0 x_6^L (x_2^1 \vee x_2^2 \vee x_2^3) = q^{24} \vee q^{25} \vee q^{26}; \\
 &x_1^0 x_2^2 x_3^H x_5^0 x_6^H = q^{30}; & x_1^M x_2^1 x_3^H x_5^0 x_6^H = q^{31}; \\
 &x_1^M x_2^2 x_3^H x_5^0 x_6^H = q^{32}; \\
 &x_5^i (x_1^0 x_2^1 \vee x_1^0 x_2^2 \vee x_1^0 x_2^3 \vee x_1^M x_2^1 \vee x_1^M x_2^2 \vee x_1^M x_2^3) = \\
 &= q^{33} \vee q^{34} \vee q^{35} \vee q^{36} \vee q^{37} \vee q^{38}.
 \end{aligned}$$

Родинними називаються такі рівності, які і після виконання над ними операції почленної диз'юнкції призводять до рівностей з лівою частиною у вигляді логічного добутку, кожен співмножник якого залежить лише від однієї предметної змінної. Мотивом, що спонукав виконати операцію почленної диз'юнкції, є прагнення отримати економну множину імен впливів контекстів, в якій кожному впливу контексту відповідало б в точності одне ім'я.

Формуємо функцію переходу від номерів осередків парадигматичної таблиці q до номерів впливу контексту r :

$$\begin{aligned}
 &q^1 = r^1; & q^2 = r^2; & q^3 = r^3; & q^4 = r^4; & q^5 = r^5; \\
 &q^6 = r^6; & q^7 = r^7; & q^8 = r^8; & q^9 = r^9; \\
 &q^{10} = r^{10}; & q^{11} = r^{11}; & q^{12} \vee q^{13} \vee q^{14} = r^{12}; \\
 &q^{15} \vee q^{16} \vee q^{17} \vee q^{21} \vee q^{22} \vee \\
 &\vee q^{23} \vee q^{27} \vee q^{28} \vee q^{29} = r^{13}; \\
 &q^{18} \vee q^{19} \vee q^{20} = r^{14}; & q^{24} \vee q^{25} \vee q^{26} = r^{15}; \\
 &q^{30} = r^{16}; & q^{31} = r^{17}; & q^{32} = r^{18}; \\
 &q^{33} \vee q^{34} \vee q^{35} \vee q^{36} \vee q^{37} \vee q^{38} = r^{19}.
 \end{aligned}$$

Вводимо номери впливів контексту, перенумеровуючи осередки парадигматичної таблиці 2 (табл. 3):

			x_5^0				$x_3^{T,MB} x_6^H$	x_5^i
			x_6^L			x_3^M		
			$x_3^{T,MB}$	x_3^M				
				x_4^H	x_4^J			
x_1^0	x_2^1	x_7^1	1	12	14	15	16	19
		x_7^2						
	x_2^2	x_7^1	2	12	14	15	16	19
		x_7^2		3				
	x_2^3	x_7^1	4	12	14	15	16	19
		x_7^2		5				
x_1^M	x_2^1	x_7^1	6	13	13	13	17	19
		x_7^2		7				
	x_2^2	x_7^1	8	13	13	13	18	19
		x_7^2		9				
	x_2^3	x_7^1	10	13	13	13	16	19
		x_7^2		11				

Формуємо залежність номера впливу контексту r від змінних $x_1, x_2, x_3, x_4, x_5, x_6, x_7$:

$$\begin{aligned}
 &x_1^0 x_2^1 x_3^H x_5^0 x_6^L x_7^1 = r^1; & x_1^0 x_2^2 x_3^H x_5^0 x_6^L x_7^1 = r^2; \\
 &x_1^0 x_2^2 x_3^H x_5^0 x_6^L x_7^2 = r^3; & x_1^0 x_2^3 x_3^H x_5^0 x_6^L x_7^1 = r^4; \\
 &x_1^0 x_2^3 x_3^H x_5^0 x_6^L x_7^2 = r^5; & x_1^M x_2^1 x_3^H x_5^0 x_6^L x_7^1 = r^6; \\
 &x_1^M x_2^1 x_3^H x_5^0 x_6^L x_7^2 = r^7; & x_1^M x_2^2 x_3^H x_5^0 x_6^L x_7^1 = r^8; \\
 &x_1^M x_2^2 x_3^H x_5^0 x_6^L x_7^2 = r^9; & x_1^M x_2^3 x_3^H x_5^0 x_6^L x_7^1 = r^{10}; \\
 &x_1^M x_2^3 x_3^H x_5^0 x_6^L x_7^2 = r^{11}; \\
 &x_1^0 x_3^M x_4^H x_5^0 x_6^L (x_2^1 \vee x_2^2 \vee x_2^3) = r^{12}; \\
 &x_1^M x_3^M x_4^H x_5^0 x_6^L (x_2^1 x_4^H \vee x_2^2 x_4^H \vee x_2^3 x_4^H \vee x_2^1 x_4^J \vee x_2^2 x_4^J \vee \\
 &\vee x_2^3 x_4^J \vee x_2^1 x_4^C \vee x_2^2 x_4^C \vee x_2^3 x_4^C) = r^{13}; \\
 &x_1^0 x_3^M x_4^J x_5^0 x_6^L (x_2^1 \vee x_2^2 \vee x_2^3) = r^{14}; \\
 &x_1^0 x_3^M x_4^C x_5^0 x_6^L (x_2^1 \vee x_2^2 \vee x_2^3) = r^{15}; \\
 &x_1^0 x_2^2 x_3^H x_6^H = r^{16}; & x_1^M x_2^1 x_3^H x_6^H = r^{17}; & x_1^M x_2^2 x_3^H x_6^H = r^{18}; \\
 &x_5^i (x_1^0 x_2^1 \vee x_1^0 x_2^2 \vee x_1^0 x_2^3 \vee x_1^M x_2^1 \vee x_1^M x_2^2 \vee x_1^M x_2^3) = r^{19}.
 \end{aligned}$$

Проводимо бінаризацію шойно записаного відношення, яке пов'язує змінну r із змінними $x_1, x_2, x_3, x_4, x_5, x_6, x_7$.

Для цього знаходимо відношення P_i , що зв'язує змінні x_i і r :

$$P_1(x_1, r) = x_1^0(r^1 \vee r^2 \vee r^3 \vee r^4 \vee r^5 \vee r^{12} \vee r^{14} \vee r^{15} \vee r^{16}) \vee x_1^M(r^6 \vee r^7 \vee r^8 \vee r^9 \vee r^{10} \vee r^{11} \vee r^{13} \vee r^{17} \vee r^{18}) \vee (x_1^> \vee x_1^<)r^{19}.$$

Знаходимо відношення P_2 , що зв'язує змінні x_2 і r :

$$P_2(x_2, r) = x_2^1(r^1 \vee r^6 \vee r^7 \vee r^{17}) \vee x_2^2(r^2 \vee r^3 \vee r^8 \vee r^9 \vee r^{16} \vee r^{18}) \vee x_2^3(r^4 \vee r^5 \vee r^{10} \vee r^{11}) \vee (x_2^1 \vee x_2^2 \vee x_2^3)(r^{12} \vee r^{13} \vee r^{14} \vee r^{15} \vee r^{19}).$$

Знаходимо відношення P_3 , що зв'язує змінні x_3 і r :

$$P_3(x_3, r) = x_3^H(r^1 \vee r^2 \vee r^3 \vee r^4 \vee r^5 \vee r^6 \vee r^7 \vee r^8 \vee r^9 \vee r^{10} \vee r^{11} \vee r^{16} \vee r^{17} \vee r^{18}) \vee x_3^M(r^{12} \vee r^{13} \vee r^{14} \vee r^{15}) \vee (x_3^H \vee x_3^M)r^{19}.$$

Знаходимо відношення P_4 , що зв'язує змінні x_4 і r :

$$P_4(x_4, r) = x_4^U r^{12} \vee x_4^X r^{14} \vee x_4^C r^{15} \vee (x_4^U \vee x_4^X \vee x_4^C)(r^1 \vee r^2 \vee r^3 \vee r^4 \vee r^5 \vee r^6 \vee r^7 \vee r^8 \vee r^9 \vee r^{10} \vee r^{11} \vee r^{13} \vee r^{16} \vee r^{17} \vee r^{18} \vee r^{19}).$$

Знаходимо відношення P_5 , що зв'язує змінні x_5 і r :

$$P_5(x_5, r) = x_5^O(r^1 \vee r^2 \vee r^3 \vee r^4 \vee r^5 \vee r^6 \vee r^7 \vee r^8 \vee r^9 \vee r^{10} \vee r^{11} \vee r^{12} \vee r^{13} \vee r^{14} \vee r^{15} \vee r^{16} \vee r^{17} \vee r^{18}) \vee x_5^I r^{19}.$$

Знаходимо відношення P_6 , що зв'язує змінні x_6 і r :

$$P_6(x_6, r) = x_6^D(r^1 \vee r^2 \vee r^3 \vee r^4 \vee r^5 \vee r^6 \vee r^7 \vee r^8 \vee r^9 \vee r^{10} \vee r^{11} \vee r^{12} \vee r^{13} \vee r^{14} \vee r^{15}) \vee x_6^H(r^{16} \vee r^{17} \vee r^{18}) \vee (x_6^D \vee x_6^H)r^{19}.$$

Знаходимо відношення P_7 , що зв'язує змінні x_7 і r :

$$P_7(x_7, r) = x_7^1(r^2 \vee r^4 \vee r^6 \vee r^8 \vee r^{10}) \vee x_7^2(r^3 \vee r^5 \vee r^7 \vee r^9 \vee r^{11}) \vee (x_7^1 \vee x_7^2)(r^1 \vee r^{12} \vee r^{13} \vee r^{14} \vee r^{15} \vee r^{16} \vee r^{17} \vee r^{18} \vee r^{19}).$$

Вводимо дієслівне закінчення, яке складається з першої, другої, третьої букв – z_1 , z_2 та z_3 відповідно.

Розглянемо відношення P_8 , яке зв'язує номер впливу контексту r з першою буквою закінчення. Запишемо відношення P_8 у вигляді формули:

$$P_8(z_1, z) = (z_1^Y \vee z_1^{10})(r^1 \vee r^{11}) \vee (z_1^E \vee z_1^T)(r^2 \vee r^4 \vee r^6 \vee r^8) \vee (z_1^8 \vee z_1^W)(r^3 \vee r^5 \vee r^7 \vee r^9) \vee (z_1^A \vee z_1^B)r^{11} \vee z_1^B r^{12} \vee z_1^I(r^{13} \vee r^{14} \vee r^{15}) \vee (z_1^8 \vee z_1^-)r^{16} \vee (z_1^I \vee z_1^M)r^{17} \vee (z_1^I \vee z_1^T)r^{18} \vee z_1^T r^{19}.$$

Розглянемо відношення P_9 , яке зв'язує номер впливу контексту r з другою буквою закінчення. Запишемо відношення P_9 у вигляді формули:

$$P_9(z_2, z) = z_2^-(r^1 \vee r^4 \vee r^{12} \vee r^{16}) \vee z_2^{III}(r^2 \vee r^3) \vee z_2^T(r^5 \vee r^8 \vee r^9 \vee r^{10} \vee r^{11}) \vee z_2^M(r^6 \vee r^7) \vee z_2^H r^{13} \vee z_2^A r^{14} \vee z_2^O r^{15} \vee (z_2^M \vee z_2^O)r^{17} \vee (z_2^T \vee z_2^E)r^{18} \vee (z_2^H \vee z_2^B)r^{19}.$$

Розглянемо відношення P_{10} , яке зв'язує номер впливу контексту r з третьою буквою закінчення. Запишемо відношення P_{10} у вигляді формули:

$$P_{10}(z_3, z) = z_3^-(r^1 \vee r^2 \vee r^3 \vee r^4 \vee r^{12} \vee r^{13} \vee r^{14} \vee r^{15} \vee r^{16} \vee r^{19}) \vee z_3^b(r^5 \vee r^{10} \vee r^{11}) \vee z_3^O(r^6 \vee r^7) \vee z_3^E(r^8 \vee r^9) \vee (z_3^- \vee z_3^O)r^{17} \vee (z_3^b \vee z_3^-)r^{18}.$$

Ми побудували математичну модель для дієслів української мови. Вона характеризується

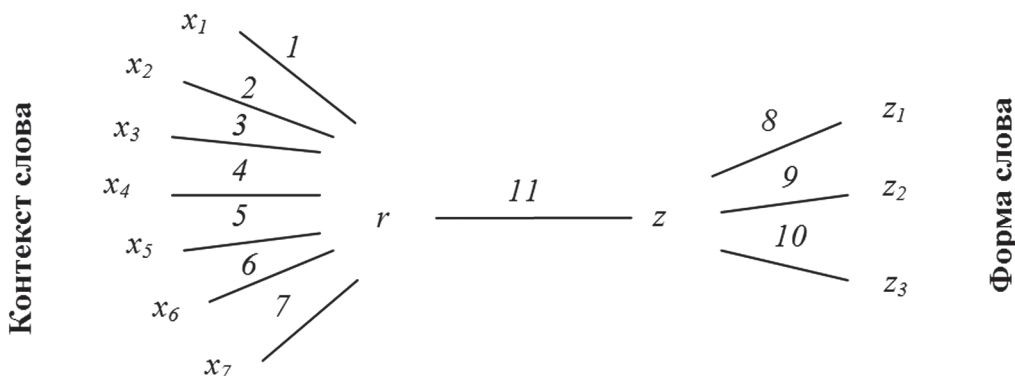


Рис. 1. Логічна мережа

системою бінарних відносин P_1-P_{10} , що задаються формулами відповідних предикатів. Утворюючи кон'юнкцію всіх цих предикатів, отримуємо предикат моделі:

$$P(x_1, x_2, x_3, x_4, x_5, x_6, x_7, r, z, z_1, z_2, z_3) = P_1(x_1, r) \wedge \\ \wedge P_2(x_2, r) \wedge P_3(x_3, r) \wedge P_4(x_4, r) \wedge P_5(x_5, r) \wedge \\ \wedge P_6(x_6, r) \wedge P_7(x_7, r) \wedge P_8(z_1, z) \wedge P_9(z_2, r) \wedge P_{10}(z_3, r).$$

Предикату моделі відповідає відношення моделі P , що зв'язує між собою предметні змінні $x_1, x_2, x_3, x_4, x_5, x_6, x_7, r, z, z_1, z_2, z_3$. Відношення моделі P можна наочно зобразити у вигляді мережі, яка називається логічною. Логічну мережу можна побудувати для будь-якої моделі. У нашому конкретному випадку отримуємо логічну мережу дієслів. Вона зображена на рис. 1.

Кожній гілці логічної мережі ставиться у відповідність своє бінарне відношення моделі, яка називається відношенням цієї гілки. Кожна гілка позначається номером свого відношення. Вона з'єднує два полюси, що відповідають тим предметним змінним, які зв'язуються відношенням, відповідним даній гілці. В побудованій мережі є 11 полюсів і 11 гілок.

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, r, z, z_1, z_2, z_3.$$

Полюси мережі діляться на два класи - зовнішні і внутрішні. Кожен зовнішній полюс з'єднаний лише з однією гілкою, кожен внутрішній - більш, ніж з однією. У побудованій мережі до зовнішніх відносяться полюси, помічені предметними змінними $x_1, x_2, x_3, x_4, x_5, x_6, x_7, z_1, z_2, z_3$, до внутрішніх - r, z . Зовнішні полюси мережі об'єднують у дві групи: 1) $x_1, x_2, x_3, x_4, x_5, x_6, x_7$ 2) z_1, z_2, z_3 . У першу групу входять полюси, помічені тими предметними змінними, значення яких характеризують контекст, який оточує слово. У другу групу входять полюси, які характеризують словоформу, що відповідає даному контексту і даному слову.

Кожну логічну мережу можна перетворити в електронну схему для автоматичного рішення деякого класу задач, що визначається тією моделлю, для якої була побудована дана мережа [3].

Висновки

В статті була побудована модель логічної мережі для дієслів української мови. Вона характеризується системою бінарних відношень та формулами відповідних предикатів. Мережа призначена для розв'язання системи рівнянь, що задають відповідну модель.

Електронну схему, яка зображена на рис. 1, можна використовувати для автоматичного розв'язання класу задач, що відносяться до визначення правильної форми дієслів. Наприклад, задача синтезу словоформи (задані слово і оточуючий контекст, потрібно визначити відповідну їм словоформу), задача аналізу словоформи (задані форма слова і оточуючий її контекст, потрібно визначити граматичні ознаки, які відповідають цій словоформі) та задача нормалізації словоформи (задані форма слова і оточуючий її контекст, потрібно знайти слово, що відповідає цій словоформі).

В подальшому планується до логічної мережі додати третю гілку, до якої будуть входити полюси, що характеризують саме слово. Завдяки цьому розв'язання поставлених задач буде більш точним.

Список літератури: 1. Широков, В.А. Феноменологія інформаційних систем [Текст] / В.А. Широков. - К.: Наукова думка, 2004. - 326 с. 2. Бондаренко, М.Ф. Теорія інтелекта [Текст] : учебник / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнарченко. - Х.: СМІТ, 2006. - 575 с. 3. Шабанов-Кушнарченко Ю.П. Модели языка [Текст] / М.Ф. Бондаренко, Ю.П. Шабанов-Кушнарченко, В.А. Чикина // Бионика интеллекта - 2004. - № 61/1. - С. 27-37. 4. Бондаренко, М.Ф. Основи території синтезу надшвидкодуючих структур мовних систем штучного інтелекту [Текст] / М.Ф. Бондаренко, З. Д. Коноплянко, Г.Г. Четвериков. - К., 1997. - 263 с.

Надійшла до редколегії 24.11.2011

УДК 519.62

Построение логической сети для флективной обработки глаголов украинского языка / Т.Н. Федорова // Бионика интеллекта: науч.-техн. журнал. - 2012. - № 1 (78). - С. 111-115.

Статья посвящена построению модели флективной обработки логической сетью для глаголов украинского языка. Она характеризуется системой бинарных отношений и формулами соответствующих предикатов. Сеть предназначена для решения системы уравнений, которые задаются соответствующей моделью.

Ил. 1. Библиогр.: 4 назв.

UDK 519.62

Building the logical network for the inflected processing of the Ukrainian language's verbs / T.N. Fyodorova // Bionics of Intelligence: Sci. Mag. - 2012. - № 1 (78). - P. 111-115.

The article is devoted to constructing model of inflectional processing of the logical network for verbs of the Ukrainian language. It is characterized by a system of binary relations and the formulas of the predicate. The network is intended to solve systems of equations, which are determined by the appropriate model.

Fig. 1. Ref.: 4 items.

УДК 519.7



И.В. Каменева

ХНУРЭ, г. Харьков, Украина, irina.kamenieva@gmail.com

АНАЛИЗ ФЛЕКСИИ ПИСЬМЕННОЙ И УСТНОЙ МОДЕЛЕЙ ПОЛНЫХ НЕПРИТЯЖАТЕЛЬНЫХ ИМЕН ПРИЛАГАТЕЛЬНЫХ РУССКОГО ЯЗЫКА

В данной статье рассматриваются разработки морфологических структур русского языка предшественников, проводится сравнительный анализ эффективности реляционной сети склонения полных непряжательных имен прилагательных русского языка.

МОЗГОПОДОБНЫЕ СТРУКТУРЫ, РЕЛЯЦИОННАЯ СЕТЬ, МАТЕМАТИЧЕСКАЯ МОДЕЛЬ,
ПОЛНЫЕ НЕПРИТЯЖАТЕЛЬНЫЕ ИМЕНА ПРИЛАГАТЕЛЬНЫЕ

Введение

Использование механизмов естественного языка, с которым непосредственно связан человек, обеспечивает создание усовершенствованных новейших средств и технологий обработки информации для интеллектуализации работы с текстовой информацией. На сегодняшний день проблема формализации естественного языка является актуальной, и интеллектуальные системы автоматизированной обработки естественно-языковой информации активно развиваются.

Механизм естественного языка невозможно охватить одним конкретным исследованием. В нем для исследования приходится выделять некоторую область, с которой предстоит упорная и без сомнения трудная работа. Существует огромное множество естественных языков, но исследования проводятся в недрах русского языка, в частности, раздела морфологии.

Как известно, одна из задач морфологии – изучение правил образования форм слова. «Слово – это «атом» языка, нечто первичное и простейшее в языке. И, тем не менее, даже отдельно взятое слово представляет собой чрезвычайно сложный объект для математического описания» [1]. Большое значение при исследовании взаимодействия термина и текста имеет морфологическая структура слова.

Обработка естественных языковых конструкций позволяет разрабатывать модели естественного языка и свидетельствует о том, что классическая грамматика может с успехом использоваться как основа для математического и технического моделирования механизма естественного языка.

Попытки формализации естественного языка привели к формальному описанию отношений, без которых в принципе работать со структурой языка невозможно. Поскольку в проделанной работе представить отношения формулами не удалось, была реализована алгебра предикатов, которая выражает предикаты заданными отношениям и наоборот. Создание алгебры предикатов помогло формально описать и автоматизировать мыслительную деятельность человека, но в процессе работы возникла преграда в записи и обработке

отношений, которые выходят за рамки классического определения понятия алгебры. В процессе долгой и трудоемкой работы была создана алгебраическая система предикатов. Понятие «алгебраическая система» впервые было представлено академиком А.И. Мальцевым [2]. В зависимости от условий поставленной задачи используется также алгебра предикатных операций [3].

Были и другие попытки формализации естественного языка. В частности Давид Гильберт, в своем курсе лекций по основам теоретической логики описал идею косвенного представления отношений, но в дальнейшем ни он сам, ни другие авторы к идее формализации естественного языка больше не обращались [4].

Цель данного исследования – приблизиться к усовершенствованию существующих письменной и звуковой математических моделей, разработанных для склонений полных непряжательных имен прилагательных русского языка.

1. Мозгоподобные структуры

Основным понятием для мозгоподобных структур является математическая структура. Математическая структура – родовое название, объединяющее понятия, общей чертой которых является то, что они применимы к множествам, природа элементов которых не определена. Чтобы определить структуру, задают отношения, в которых находятся элементы множеств (типовая характеристика структуры), а затем постулируют, что данные отношения удовлетворяют условиям – аксиомам структуры».

Характерной особенностью мозгоподобных структур является слияние памяти с обработкой данных: данные обрабатываются одновременно по всей памяти с максимально возможной степенью распараллеливания всех операций. Мозгоподобные структуры с параллельными процессами, управляемыми многими потоками данных и команд, несомненно, представляют собой высший уровень развития архитектур ЭВМ.

Сегодня уже появилась возможность создания вычислительных структур с производитель-

ностью, сравнимой с производительностью мозга человека [5].

Для того чтобы не возникало сложностей с некоторыми выражениями в следующем разделе, определим, что собой представляет математическая модель. Математическая модель – это формализованное описание каких-либо отношений во внешнем мире, выраженных математическими символами.

На сегодняшний день научной школой Юрия Петровича Шабанова-Кушнаренко построены модели спряжения глаголов [6], склонения имен существительных [7], склонения полных непряжательных имен прилагательных [8], которые реализованы в виде реляционных сетей для создания мозгоподобных ЭВМ.

Благодаря работе с непряжательными именами прилагательными, появилась идея сравнить письменную модель полных непряжательных имен прилагательных и звуковую модель склонения потенциальных слов и их потенциальных форм полных непряжательных имен прилагательных для морфологических структур.

2. Анализ процесса склонения полных непряжательных имен прилагательных

Остановимся более подробно на сравнении двух математических моделей (письменной и звуковой), разработанных для полных непряжательных имен прилагательных [9, 10]. Для этих моделей ниже формально представлены предметные переменные: x_1 – род формы слова со значениями М – мужской, Ж – женский, С – средний; x_2 – число формы слова со значениями Е – единственное, М – множественное; x_3 – падеж формы слова со значениями И – именительный, Р – родительный, Д – дательный, В – винительный, Т – творительный, П – предложный; x_4 – признак одушевленности формы слова со значениями О – одушевленный, Н – неодушевленный; x_5 – признак современности формы слова со значениями С – современный, А – архаичный; r – номер влияния контекста, s – тип склонения слова; y_1 – последняя буква основы слова, y_2 – признак ударности слова, y_3 – признак смягчения слова, z – окончание формы слова, где z_l – левая часть окончания, $z_{п}$ – правая часть окончания. Для письменной математической модели полных непряжательных имен прилагательных при развертке окончания z по буквам используется знак * при переменной z_0 , что означает отсутствие знака ударения над первой буквой окончания, а при развертке окончания z по буквам знак * при переменной z_3 означает отсутствие третьей буквы в окончании. Итак, математическую модель склонения полных письменных непряжательных имен прилагательных представляет собой предикат модели, который состоит из 17 конечных бинарных отношений:

$$P(x_1, x_2, x_3, x_4, x_5, r, s, y_1, y_2, y_3, z_l, z_{п}, z, z_0, z_1, z_2, z_3) = P_1(x_1, r) \wedge P_2(x_2, r) \wedge P_3(x_3, r) \wedge P_4(x_4, r) \wedge P_5(x_5, r) \wedge$$

$$\wedge P_6(z_l, r) \wedge P_7(z_{п}, r) \wedge P_8(z_l, z) \wedge P_9(z_{п}, z) \wedge P_{10}(z_l, z_0) \wedge \wedge P_{11}(z_l, z_1) \wedge P_{12}(z_{п}, z_2) \wedge P_{13}(z_{п}, z_3) \wedge P_{14}(s, z_l) \wedge \wedge P_{15}(y_1, s) \wedge P_{16}(y_2, s) \wedge P_{17}(y_3, s).$$

Предикату данной модели соответствует отношение модели P , связывающее между собой предметные переменные

$$x_1, x_2, x_3, x_4, x_5, r, s, y_1, y_2, y_3, z_l, z_{п}, z, z_0, z_1, z_2, z_3.$$

Отношение модели P можно наглядно изобразить в виде сети, называемой логической. Логическую сеть можно построить для любой модели. В конкретном случае логическая сеть склонения полных непряжательных имен прилагательных имеет вид, представленный на рис. 1.

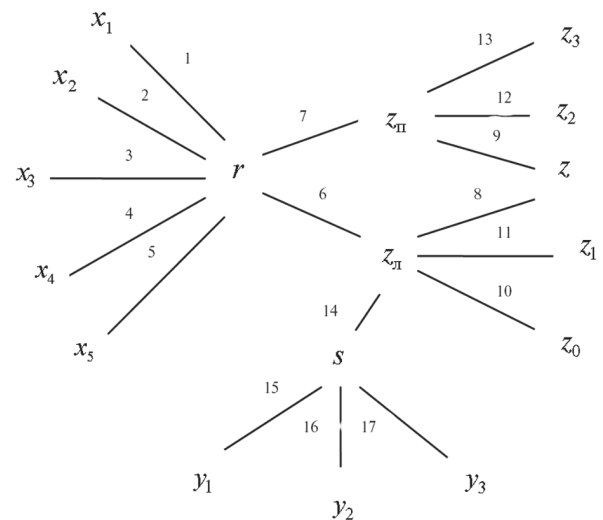


Рис. 1

Математическую модель склонения полных звуковых (устных) непряжательных имен прилагательных представляет собой предикат модели, который состоит из 11 конечных бинарных отношений:

$$P(x_1, x_2, x_3, x_4, x_5, r, s, y_1, y_2, y_3, z_l, z_{п}) = P_1(x_1, r) \wedge P_2(x_2, r) \wedge P_3(x_3, r) \wedge P_4(x_4, r) \wedge P_5(x_5, r) \wedge \wedge P_6(y_1, s) \wedge P_7(y_2, s) \wedge P_8(y_3, s) \wedge P_9(s, z_{п}) \wedge \wedge P_{10}(r, z_{п}) \wedge P_{11}(s, z_l),$$

где аргументы типов склонения: y_1 – последний звук основы слова, y_2 – признак ударности со значениями: У – ударный, Б – безударный, y_3 – признак смягчения со значениями: Т – твердый, М – мягкий. Модель характеризуется системой бинарных отношений $P_1 - P_{11}$, задаваемых формулами соответствующих предикатов. Образую конъюнкцию всех этих предикатов, получаем предикат модели. А также данная модель представляется в виде логической сети (рис. 2).

Каждая из представленных моделей имеет свои достоинства и недостатки. Речь идет о том, что в письменной модели не была рассмотрена потенциальная форма слова, а именно – способность человека склонять не только употребляемые в языке слова, но также и многие псевдослова т.е. словообразные

буквосочетания. Соответственно возникла сложность при склонении несуществующих в грамматике слов. А в звуковой модели такая проблема была решена, и возможность склонения потенциальных слов осуществлена в полном объеме. Недостаток звуковой модели в том, что в ней не рассмотрено разбиение правой $z_{\text{п}}$ и левой $z_{\text{л}}$ частей окончания на буквы окончания и признак ударности, тогда как в письменной модели окончания были успешно рассмотрены и внедрены в модель.

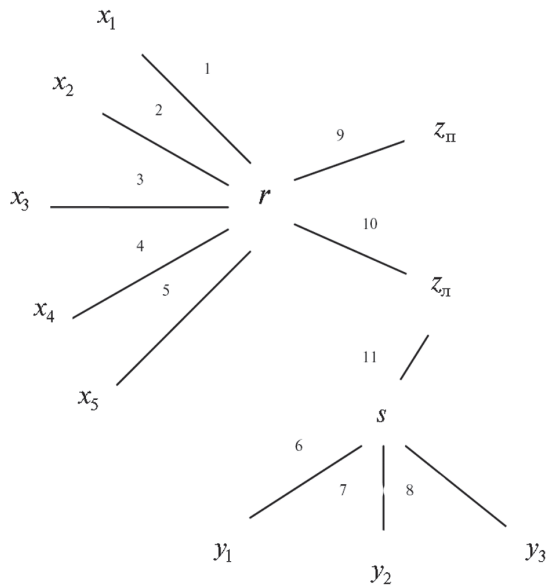


Рис. 2

Результаты моделирования устного словоизменения заставляют внести коррективы в известную модель письменного словоизменения. Например, в словах «длинношей», «-шуй», в качестве последней буквы основы при письменном представлении слов и их форм в этой модели был принят знак пробела, и это не приводило к появлению беглой буквы в основе слова. В устном же варианте при склонении этих слов неизбежна беглость звука основы («-шуй», «-шуйго» и т.д.), и поэтому приходится в роли последнего звука основы принять беглый согласный «й». Если письменную модель скорректировать по устной, то первая примет более логичный вид, поскольку теперь все без исключения последние буквы основы окажутся согласными.

Выводы

Анализируя полученные результаты сравнения, где рассматриваются две конкретные мозгоподобные структуры, а именно устная и письменная модели склонения полных непряжательных имен прилагательных, заключаем, что в дальнейшем планируется устранение представленных недостатков, что позволит решить стоящую перед нами задачу — объединение моделей и проверка их универсальности. Последующие задачи, которые будут исследоваться и разрабатываться, также охватят морфологическую структуру естественного языка,

которая позволит рассмотреть реляционную сеть потенциальных полных непряжательных имен прилагательных, представленных в устной и письменной формах как отдельные части, близкие к совершенной сети.

Список литературы: 1. Бондаренко, М.Ф. Теория интеллекта [Текст]/ М.Ф. Бондаренко, Ю.П. Шабанов-Кушнарченко // Теория интеллекта: Учебник. — Х.: ООО «Компания СМИТ», 2006. — 576 с. 2. Мальцев, А.И. Алгебраические системы [Текст]/ А.И. Мальцев. — М.: Наука, 1970. — 392 с. 3. Бондаренко, М.Ф. Алгебра предикатов и предикатных операций [Текст] / М.Ф. Бондаренко, З.В. Дударь, Н.Т. Процай, В.В. Черкашин, В.А. Чикина, Ю.П. Шабанов-Кушнарченко // Радиоэлектроника и информатика. — 2000. — № 4. — С. 15–23. 4. Гильберт, Д. Основы теоретической логики [Текст] / Д. Гильберт, В. Аккерман. — М.: ИЛ, 1947. — 302 с. 5. Бондаренко, М.Ф. О мозгоподобных структурах академика Виктора Михайловича Глушкова [Текст] / М.Ф. Бондаренко, Н.Е. Русакова, Ю.П. Шабанов-Кушнарченко // Бионика интеллекта науч.-техн. журнал. — 2011. — № 2 (76). — С. 3–9. 6. Дударь, З.В. Математические модели флективной обработки словоформ и их использование в системах автоматической обработки текста русского языка [Текст]: Дис. ... канд. техн. наук. — Х.: ХИРЭ, 1984. — 215 с. 7. Лецинский, В.А. Модели бинарных логических сетей и их применение в искусственном интеллекте [Текст]: Дис. ... канд. техн. наук. — Х.: ХНУРЭ, 2006. — 157 с. 8. Мельникова, Р.В. Алгебраические модели морфологии и их применение в логических сетях. Дис. ... канд. техн. наук. — Х.: ХНУРЭ, 2005 — 152 с. 9. Бондаренко, М.Ф. Модели языка [Текст] / М.Ф. Бондаренко, В.А. Чикина, Ю.П. Шабанов-Кушнарченко // Бионика интеллекта: науч.-техн. журнал. — 2004. — № 1(61). — С. 27–36. 10. Русакова, Н.Е. Модели устной речи [Текст] / Н.Е. Русакова // Бионика интеллекта: науч.-техн. журнал. — 2010. — №1(72). — С. 94–97.

Поступила в редколлегию 22.12.2011

УДК 519.7

Аналіз флексії письмової та усної моделей повних непряжательних прикметників російської мови / І.В. Каменева // Біоніка інтелекту : наук.-техн. журнал. — 2012. — № 1 (78). — С. 116–118.

У статті розглянуто розробки попередників, які займалися морфологічними структурами російської мови, проведено порівняльний аналіз ефективності реляційної мережі відмінювання повних непряжательних прикметників письмової та звукової моделей російської мови, що дало змогу виявити деякі недоліки, щоб врахувати їх для удосконалення майбутніх розробок.

Л. 2. Бібліогр.: 10 найм.

UDC 519.7

The adjectives inflection of Russian language/I. Kamenieva // Bionics of Intelligense: Sci. Mag. — 2012. — № 1 (78). — P. 116–118.

The article discussed the precursors's development who were involved in morphological structures of the Russian language. The comparative analysis of the effectiveness of the relational network adjectives of the Russian language allowed to identify some shortcomings for improve future developments.

Fig. 2. Ref.: 10 items.

ОБ АВТОРАХ

Авраменко Виктор Васильевич	72	канд. техн. наук, доцент кафедры компьютерных наук Сумского государственного университета
Аксак Наталия Георгиевна	63	канд. техн. наук, старший научный сотрудник, доцент кафедры ЭВМ Харьковского национального университета радиоэлектроники
Бельчева Анна Владимировна	97	ассистент кафедры медиасистем и технологий Харьковского национального университета радиоэлектроники
Бых Анатолий Иванович	68	д-р физ.-мат. наук, профессор, заведующий кафедрой биомедицинских электронных приборов и систем Харьковского национального университета радиоэлектроники
Бондаренко Михаил Федорович	3	член-корреспондент НАН Украины, д-р техн. наук, профессор, ректор Харьковского национального университета радиоэлектроники
Ващук Федор Григорьевич	49	д-р техн. наук, профессор, ректор Закарпатского государственного университета, г. Ужгород
Волков Роман Сергеевич	72	аспирант кафедры компьютерных наук Сумского государственного университета
Глушаускайте Ирина Викторовна	102	магистрант Национального технического университета Украины «Киевский политехнический институт»
Гофман Евгений Александрович	29	аспирант кафедры программных средств Запорожского национального технического университета
Гришко Андрей Александрович	78	аспирант кафедры ЭВМ Харьковского национального университета радиоэлектроники
Губаренко Евгений Витальевич	17	аспирант кафедры системотехники Харьковского национального университета радиоэлектроники
Жемчужкина Татьяна Владимировна	68	канд. техн. наук, доцент кафедры биомедицинских электронных приборов и систем Харьковского национального университета радиоэлектроники
Довбыш Анатолий Степанович	53	д-р техн. наук, профессор, заведующий кафедрой компьютерных наук Сумского государственного университета
Дяченко Владислав Александрович	85	ассистент кафедры ЭВМ Харьковского национального университета радиоэлектроники
Заболотняя Татьяна Николаевна	102	канд. техн. наук, старший преподаватель кафедры программного обеспечения компьютерных систем Национального технического университета Украины «Киевский политехнический институт»
Каменева Ирина Витальевна	116	аспирантка кафедры программной инженерии Харьковского национального университета радиоэлектроники
Коваленко Андрей Иванович	3	канд. техн. наук, доцент кафедры информационно-документных систем Харьковской государственной академии культуры
Король Ольга Игоревна	12	аспирантка, ассистент кафедры информатики и интеллектуальной собственности Национального технического университета «Харьковский политехнический институт»

Литвин Олег Николаевич	33, 37	д-р физ.-мат. наук, профессор, заведующий кафедрой высшей и прикладной математики Украинской инженерно-педагогической академии
Мальков Юрий Анатольевич	59	аспирант кафедры ЭВМ Харьковского национального университета радиоэлектроники
Манакова Наталья Олеговна	97	канд. техн. наук, доцент кафедры медиасистем и технологий Харьковского национального университета радиоэлектроники
Матус Светлана Константиновна	23	старший преподаватель кафедры электротехники и автоматики Национального университета водного хозяйства и природопользования, г. Ровно
Мельник Елена Алексеевна	49	старший преподаватель кафедры программного обеспечения систем Закарпатского государственного университета, г. Ужгород
Михаль Олег Филиппович	85, 91	д-р техн. наук, профессор кафедры ЭВМ Харьковского национального университета радиоэлектроники
Носова Татьяна Витальевна	68	канд. техн. наук, доцент кафедры биомедицинских электронных приборов и систем Харьковского национального университета радиоэлектроники
Олейник Андрей Александрович	29	канд. техн. наук, доцент кафедры программных средств Запорожского национального технического университета
Осотов Иван Андреевич	91	аспирант кафедры ЭВМ Харьковского национального университета радиоэлектроники
Пастушенко Владимир Иосифович	23	канд. техн. наук, доцент кафедры электротехники и автоматики Национального университета водного хозяйства и природопользования, г. Ровно
Першина Юлия Игоревна	37	канд. физ.-мат. наук, докторант кафедры высшей и прикладной математики Украинской инженерно-педагогической академии
Петров Эдуард Георгиевич	17	д-р техн. наук, профессор, заведующий кафедрой системотехники Харьковского национального университета радиоэлектроники
Радченко Вячеслав Алексеевич	59	аспирант кафедры ЭВМ Харьковского национального университета радиоэлектроники
Романюк Анатолий Николаевич	53	д-р медицинских наук, профессор, заведующий кафедрой патоморфологии Сумского государственного университета
Руденко Максим Сергеевич	53	аспирант кафедры компьютерных наук Сумского государственного университета
Руденко Олег Григорьевич	45	д-р техн. наук, профессор, заведующий кафедрой ЭВМ Харьковского национального университета радиоэлектроники
Ситников Дмитрий Эдуардович	3	канд. техн. наук, доцент, заведующий кафедрой информационно-документных систем Харьковской государственной академии культуры
Ситникова Полина Эдуардовна	3	канд. техн. наук, доцент, заведующая кафедрой информационных технологий и математики Харьковского гуманитарного университета «Народная украинская академия»
Сныткин Максим Сергеевич	45	научный сотрудник кафедры ЭВМ Харьковского национального университета радиоэлектроники

Стеценко Анастасия Михайловна	23	старший преподаватель кафедры электротехники и автоматики Национального университета водного хозяйства и природопользования, г. Ровно
Субботин Сергей Александрович	29	канд. техн. наук, доцент кафедры программных средств Запорожского национального технического университета
Сухенький Владимир Анатольевич	68	студент кафедры биомедицинских электронных приборов и систем Харьковского национального университета радиоэлектроники
Тянянский Сергей Станиславович	59	д-р техн. наук, доцент кафедры ЭВМ Харьковского национального университета радиоэлектроники
Удовенко Сергей Григорьевич	78	д-р техн. наук, профессор кафедры ЭВМ Харьковского национального университета радиоэлектроники
Федорова Татьяна Николаевна	111	аспирантка кафедры программной инженерии Харьковского национального университета радиоэлектроники
Чалая Лариса Эрнестовна	78	канд. техн. наук, доцент кафедры искусственного интеллекта Харьковского национального университета радиоэлектроники
Шабанов-Кушнаренко Юрий Петрович	3	д-р техн. наук, профессор кафедры программной инженерии Харьковского национального университета радиоэлектроники
Шаронова Наталья Валериевна	12	д-р техн. наук, профессор, заведующая кафедрой интеллектуальных компьютерных систем Национального технического университета «Харьковский политехнический институт»
Шкловец Артем Вадимович	63	аспирант кафедры ЭВМ Харьковского национального университета радиоэлектроники
Ярмош Елена Витальевна	33	старший преподаватель кафедры экономики предпринимательской и образовательной деятельности Украинской инженерно-педагогической академии

ПРАВИЛА оформлення рукописів для авторів науково-технічного журналу «БІОНІКА ІНТЕЛЕКТУ»

Науково-технічний журнал «Біоніка інтелекту» приймає до друку написані спеціально для нього оригінальні рукописи, які раніше ніде не друкувались. Структура рукопису повинна бути такою: індекс УДК, заголовок, відомості про авторів, анотація, ключові слова, вступ, основний текст статті, висновки, список використаної літератури.

Відповідно до Постанови ВАК України від 15.01.2003 №7-05/1 (Бюлетень ВАК, №1, 2003, с. 2), стаття повинна мати такі необхідні елементи: постановка проблеми в загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями; аналіз останніх досліджень і публікацій і виділення не вирішених раніше частин загальної проблеми в даній області; формулювання цілей та завдань дослідження; виклад основного матеріалу досліджень з повним обґрунтуванням отриманих наукових результатів; висновки з даного дослідження та перспективи подальших досліджень у даному напрямку.

Статті мають бути виконані в редакторі Microsoft Word. Формат сторінки – А4 (210x297 мм), поля: верхнє – 25 мм, нижнє – 20 мм, ліве, праве – 17 мм. Кількість колонок – 2, з інтервалом між ними 5 мм, основний шрифт Times New Roman, кегль основного тексту – 10 пунктів, міжрядковий інтервал – множник (1,1), абзацний відступ – 6 мм. Обсяг рукопису – від 4 до 12 сторінок (мови: російська, українська, англійська).

УДК друкується з першого рядка, без відступів, вирівнювання по лівому краю.

Назва статті друкується прописними літерами; шрифт прямий, напівжирний, кегль 12. *Назви розділів* нумерують арабськими цифрами, виділяють жирним шрифтом. Відступи для назви статті, ініціалів та прізвищ авторів, відомостей про авторів, назв розділів, вступу та висновків, списку літератури: зверху – 6 пт, знизу – 3 пт.

Анотацію (мовою статті, абзац 4-10 рядків, кегль 9) розміщують на початку статті, в ній має бути розміщена інформація про результати описаних досліджень.

Ключові слова (4-10 слів з тексту статті, які з точки зору інформаційного пошуку несуть змістовне навантаження) наводять мовою рукопису, через кому в називному відмінку, кегль 9.

Малюнки та таблиці (чорно-білі, контрастні) розміщуються у тексті після першого посилання у вигляді окремих об'єктів і нумерують арабськими цифрами наскрізно нумерацією за наявності більше ніж одного об'єкта. Невеликі схеми, що складаються з 3-4 елементів виконують, використовуючи вставку об'єкта Рисунок Microsoft Word. Більш складні виконують у графічних редакторах у вигляді чорно-білих графічних файлів форматів .tiff, .jpg, .wmf, .cdr із розділенням 300 dpi. Рисунки мають міститися у текстовому файлі й обов'язково

подаватися окремим файлом з відповідною назвою (наприклад, Рис.1.cdr).

Усі елементи малюнка, включаючи написи, повинні бути згруповані. Усі написи в малюнках і таблицях мають бути виконані шрифтом Times New Roman, кегль у малюнках – 10, у таблицях – 9.

Малюнок повинен мати центрований підпис (поза малюнком), шрифт 9, відступи зверху і знизу по 6 пт. Ширина малюнка має відповідати ширині колонки (або ширині сторінки).

Формули, символи, змінні, повинні бути набрані в редакторі формул MathType або Microsoft Equation. Формули розміщують посередині рядка й нумерують за наявності посилань на них у рукописі. Шрифт – Times New Roman. Висота змінної – 10 пунктів, великих і малих індексів – 8 пт, основний математичний символ – 12 (10) пт. Змінні, позначені латинськими літерами, набирають курсивом, грецькі літери, скорочення російських слів і цифри – прямим написанням. Змінні, які є в тексті, також набирають у редакторі формул.

Список літератури вміщує опубліковані джерела, на які є посилання в тексті, укладені у квадратні дужки, друкують без абзацного відступу, кегль 9 пт, відступ зверху – 6 пт.

Після списку літератури з відступом зверху 6 пт зазначають дату подання статті до редколегії. Число та місяць задають двозначними числами через крапку. Розмір шрифта – 9 пт, курсив, вирівнювання по правому краю.

Реферати (Times New Roman, кегль – 9 пунктів, 3-4 речення) подають російською та англійською мовами. Реферат не повинен дублювати текст анотації.

Разом із рукописом (на аркушах білого паперу формату А4 щільністю 80-90 г/м², надрукований на лазерному принтері, у 2-х примірниках) необхідно подати такі документи:

1. Заяву, яку повинні підписати всі автори.
2. Акт експертизи про можливість опублікування матеріалів у відкритому друці.
3. Рецензію, підписану доктором наук.
4. Відомості про авторів.
5. Електронний варіант рукопису, реферату та відомостей про авторів.
6. Оплату за публікацію.

Необхідно також зазначити один з наступних тематичних розділів, якому відповідає рукопис:

1. Теоретичні основи інформатики та кібернетики. Теорія інтелекту
2. Математичне моделювання. Системний аналіз. Прийняття рішень
3. Інтелектуальна обробка інформації. Розпізнавання образів
4. Інформаційні технології та програмно-технічні комплекси
5. Структурна, прикладна та математична лінгвістика
6. Дискусійні повідомлення

АЛГЕБРО-ЛОГІЧНІ ЗАСОБИ МОДЕЛЮВАННЯ ПРИРОДНОЇ МОВИ

Проведено аналіз алгебро-логічної структури природної мови. Розглянуто концептуально-методолгічний підхід до мови людини, що дозволяє сприймати її як деяку алгебру, а її тексти – як формули цієї алгебри.

МОВА ПРИРОДНА, АЛГЕБРА ПРЕДИКАТИВ, ВІДНОШЕННЯ, АЛГЕБРА ПРЕДИКАТНИХ ОПЕРАЦІЙ

Вступ

Формальним моделям семантико-синтаксичних структур мови відводиться вирішальна роль у сучасній проблематиці комп'ютерної лінгвістики та системах штучного інтелекту (ШтІ). Це пов'язано з необхідністю створення програмно-апаратного комплексу генерації та аналізу речень природної мови (ПМ).

1. Дослідження алгебро-логічної структури природної мови

У роботі використовується апарат алгебри предикатів [1]. Множина U може бути як скінченною, так нескінченною. У першому випадку простір U^m називатимемо скінченним, а в іншому – нескінченним.

$$P(x_1, x_2, \dots, x_n) = \begin{cases} 0, & \text{якщо } (x_1, x_2, \dots, x_n) \notin T \\ 1, & \text{якщо } (x_1, x_2, \dots, x_n) \in T. \end{cases} \quad (1)$$

Згідно з (1) можливий перехід від будь-якого відношення T до відповідного йому предикату P . Предикат P , що знаходимо по (1), називатимемо характеристичною функцією відношення T .

2. Шляхи автоматизації обробки мовної інформації

У даний час в системах штучного інтелекту машинний словник та комплекс програм (тезауруси) використовуються, як правило, для виконання будь-якої однієї функції.

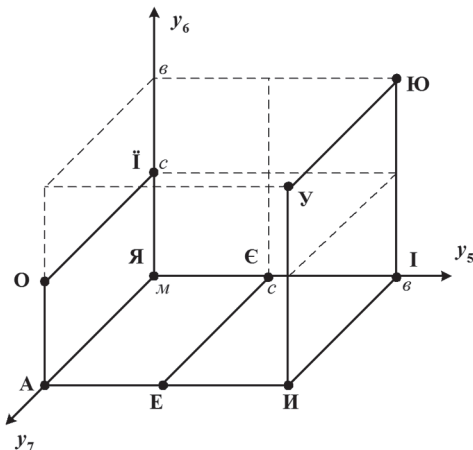


Рис.1. Тривимірний простір ознак для голосних фонем

Висновки

У статті запропонована узагальнену структуру інтелектуальної системи, яка відповідає новій інформаційній технології рішення задач на ЕОМ, що орієнтовані на досягнення високорівневої технології обробки мовної інформації (отримання нової якості). Істотно новим в роботі є розширення алгебри скінченних предикатів (АСП). Тепер вона охоплює не тільки скінченні предикати, а також – нескінченні. Тепер область її рекомендованого застосування розширена та охоплює довільні відношення, які далі будемо описувати за допомогою ДКАП.

Список літератури:

Поступила до редколегії 15.02.20012

УДК 519.62

Алгебро-логические средства моделирования естественного языка / Г.Г. Четвериков, И.Д. Вечирская // Бионика интеллекта: науч.-техн. журнал. 2007. №1 (66). С. 00-00.

В статье рассматриваются перспективные направления развития современных цифровых устройств, сетей и систем. Утверждается, что развитие средств вычислительной техники является основой автоматизации умственной деятельности человека.

Ил. 5. Библиогр.: 7 назв.

UDK 519.7

Algebra-logical tools of modeling natural language / G.G. Chetverikov, I.D. Vechirskaya // Bionics of Intelligense: Sci. Mag. – 2007. №1 (66). С. 00-00.

In article the perspective directions of modern digital devices, networks and systems development are considered. The carried out analysis shows means of computer facilities development is a baseline of automation of the man intellectual activity.

Fig. 5. Ref.: 7 items.

Видавництво здійснює остаточне форматування тексту відповідно вимогам друку.

Адреса редакції:

Україна, 61166, м.Харків, пр. Леніна 14, ХНУРЕ к.127, тел. 702-14-77, факс 702-10-13, e-mail: ira_se@list.ru, bionics@kture.kharkov.ua

Наукове видання

БІОНІКА ІНТЕЛЕКТУ
інформація, мова, інтелект

Науково-технічний журнал

№ 1 (78)

2012

Головний редактор — *М. Ф. БОНДАРЕНКО*

Відповідальний редактор — *Ю. П. Шабанов-Кушнарєнко*

Заступник відповідального редактора — *Г. Г. Четвериков*

Відповідальний секретар — *І. Д. Вечірська*

Коректор — *Л. М. Денісова*

Комп'ютерна верстка — *О. Б. Ісаєва*

Рекомендовано Вченою Радою
Харківського національного університету радіоелектроніки
(протокол № 9 від 24.02.2012)

Адреса редакції:

Україна, 61166, Харків-166, просп. Леніна, 14,
Харківський національний університет радіоелектроніки, к. 127, 285
тел. 702-14-77, факс 702-10-13,
e-mail: bionics@kture.kharkov.ua

Підписано до друку 24.02.2012. Формат 60 × 84 ¹/₈. Друк ризографічний.
Папір офсетний. Гарнітура Newton. Умов. друк. арк. 14,2. Обл.-вид. арк. 14,0.
Тираж 100 прим. Зам. № .

Надруковано в навчально-науковому видавничо-поліграфічному центрі ХНУРЕ
61166, Харків-166, просп. Леніна, 14