ИНФОРМАТИКА

2012 июль-сентябрь № 3

УДК 519.71

О.В. Шут

МЕТОД РЕШЕНИЯ ЗАДАЧ РАСПОЗНАВАНИЯ В БУЛЕВОМ ПРОСТРАНСТВЕ НА ОСНОВЕ ЛОГИЧЕСКОЙ И ПРЕЦЕДЕНТНОЙ МОДЕЛЕЙ

Рассматриваются дедуктивный и индуктивный подходы к решению задач распознавания, в которых используются соответственно логическая и прецедентная модели представления начальной информации. Строится алгебра объектов и показывается ее изоморфность булевой алгебре. Разрабатываются алгоритмы перехода между моделями представления информации. Предлагается модификация метода резолюций для прецедентной модели. Разрабатывается комбинированный алгоритм, объединяющий метод резолюций с семейством алгоритмов распознавания, и показывается, что он работает не хуже любого из алгоритмов, образующих комбинацию.

Введение

Теория распознавания как наука прошла в процессе эволюции два больших этапа. На первом этапе рассматривались прикладные вопросы, которые в постановочном смысле сводились к задачам распознавания [1]. Большинство применений теории распознавания связано с областями науки, плохо поддающимися формализации: медициной, химией, социологией и т. д. В этих областях трудно строить формальные теории, поэтому на первом этапе использовались эвристические алгоритмы. В основу их обоснованности был положен принцип сходимости на множестве объектов, информация о которых априори известна.

Второй этап развития теории распознавания характеризуется переходом от отдельных алгоритмов к рассмотрению моделей решения прикладных задач. На этом этапе эвристикой являлся уже не выбор алгоритма, а выбор принципа, в соответствии с которым алгоритмы могли быть построены стандартным образом. На базе эвристических алгоритмов выстраивались схемы, которые позволяли нивелировать недостатки эвристических алгоритмов. К числу таких схем относятся схемы логической корректировки, алгебраические модели, в основу обоснованности которых положены необходимые условия правильности в виде, например, корректности алгоритмов [2].

В настоящее время можно говорить о третьем этапе развития теории распознавания. Суть его заключается в следующем. Существуют классические алгоритмы, вопрос об обосновании которых давно изучен и не вызывает сомнений. В качестве примера можно назвать метод резолюций, который имеет дедуктивную природу. Задача распознавания может быть сформулирована в двух постановках, которым соответствуют два подхода к решению: дедуктивный (метод резолюций) и индуктивный (алгоритмы распознавания). На третьем этапе разрабатываются алгоритмы, сочетающие преимущества обоих подходов.

Традиционно в каждой из двух постановок задачи распознавания используются свои модели представления начальной информации: логическая для дедуктивного подхода и прецедентная для индуктивного. В данной работе рассматриваются обе эти модели для случая, когда все признаки объектов могут принимать два значения. Показана эквивалентность этих моделей при определенных ограничениях, что дает возможность использовать метод резолюций для решения задач распознавания. На основе метода резолюций и параметрического семейства алгоритмов распознавания построен комбинированный алгоритм, работающий не хуже любого из двух исходных алгоритмов.

1. Постановка задачи

Рассмотрим следующую задачу Z в общей постановке, сформулированной в [3]:

Пусть на конечном множестве объектов X произвольной природы задано конечное число подмножеств (классов) $X_1,...,X_l$. Имеется начальная информация I_0 о принадлежности к классам множества объектов $X^0 \subset X$. Требуется указать алгоритм, определенный на всем *множестве X, который на основании информации I_0 для произвольного объекта $x \in X$ вычисляет принадлежность x к классам $X_1,...,X_l$.*

Конкретные варианты постановки данной задачи определяются способами представления множеств X и I_0, числом классов l и т. д. В данной работе рассматриваются два варианта:

1. Задача Z_1. *Информация I_0 задана логическим способом: через предикаты (правила, логические формулы), которые одновременно используются как для описания объектов, так и для описания функции принадлежности классам. Информация о принадлежности объектов, удовлетворяющих правилам, считается заданной. Необходимо для указанного объекта x определить, выводим ли он из правил, описывающих класс X_i, $i = \overline{1,l}$.*

В качестве примера задачи Z_1 можно привести задачу классификации формул в исчислении высказываний. Такие задачи решаются дедуктивными методами. Общепринятым методом решения задач, сформулированных в постановке задачи Z_1, является метод резолюций [4].

2. Задача Z_2. *Информация I_0 задана прецедентным способом: для каждого из классов $X_1,...,X_l$ явно указаны объекты, принадлежащие этому классу (т. е. для каждого объекта задан его информационный вектор [1]). Требуется указать алгоритм, который на основании информации I_0 для произвольного объекта $x \in X$ вычисляет принадлежность x к классам $X_1,...,X_l$.*

Такая постановка типична для задач распознавания образов с обучением. Для решения задачи Z_2 разработано огромное количество алгоритмов.

Сравнивая задачи Z_1 и Z_2, можно заметить, что задача Z_1 носит чисто теоретический характер, а обоснованность метода ее решения не вызывает сомнений; задача Z_2, в свою очередь, имеет прикладной характер. Однако поскольку задачи Z_1 и Z_2 являются вариантами постановки одной и той же задачи Z, их решения должны быть сравнимы. Поэтому поставим задачу Z_0 исследования связи между решениями задач Z_1 и Z_2. Именно задача Z_0 и рассматривается в данной работе.

На рис. 1 показана связь всех вышеупомянутых задач. Здесь через I_0^1 и I_0^2 обозначены исходные данные (начальная информация I_0), через Out_1 и Out_2 – результаты решения задач Z_1 и Z_2 соответственно, через A_1 и A_2 – алгоритмы решения задач: для задачи Z_1 алгоритм A_1 – это метод резолюций; для задачи Z_2 под алгоритмом A_2 может подразумеваться любой алгоритм, применяемый для решения задач распознавания. Таким образом, на рис. 1 видно, что задача Z_0 состоит в исследовании связи между Out_1 и Out_2. Разобьем эту задачу на следующие подзадачи:

Z_{01} – установление связи между моделями представления начальной информации I_0^1 и I_0^2 и построение алгоритмов перехода между ними;

Z_{02} – сравнение алгоритмов A_1 и A_2 на множестве X;

Z_{03} – построение комбинированного алгоритма, объединяющего алгоритмы A_1 и A_2 и работающего не хуже любого из этих двух алгоритмов на множестве X.

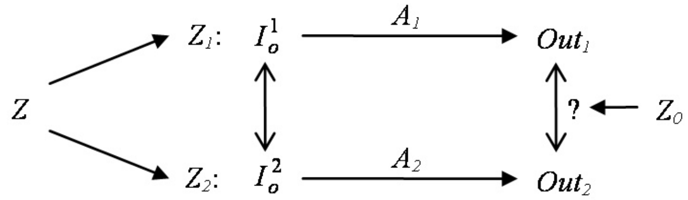

Рис. 1. Схема связи задач Z, Z_1, Z_2, Z_0

Рассмотрим подзадачи по порядку. Для решения подзадачи Z_{01} введем алгебру объектов, описанную далее, и покажем, что между прецедентной и логической моделями представления информации можно установить взаимно-однозначное соответствие.

2. Алгебра объектов

Покажем, что прецедентная и логическая модели представления информации являются эквивалентными в следующем смысле: при определенных ограничениях на число объектов и значения их признаков любая начальная информация может быть представлена с применением любой из этих двух моделей. Для этого рассмотрим алгебру объектов, которая используется в случае прецедентного представления информации, и покажем ее изоморфность булевой алгебре, используемой в логическом подходе.

2.1. Основные определения

Воспользуемся моделью представления объектов, предложенной в [5]. Пусть объект имеет конечное число признаков. Обозначим через $S = \{s_1, s_2, ..., s_n\}$ множество всех признаков, используемых в предметной области, в которой рассматривается задача Z_1 или Z_2, и зафиксируем порядок признаков. Для всех признаков будем предполагать, что если значение признака известно, то оно принадлежит множеству $D = \{0, 1\}$. Возможна также ситуация, когда значение признака неизвестно. Введем специальный символ для неизвестного значения, например «?», и обозначим расширенное таким образом множество значений признаков через $\overline{D} = \{0, 1, ?\}$.

Определение 1. *Объектом* называется отображение вида $p : s_1 \times s_2 \times ... \times s_n \to \overline{D}^n$, где $\overline{D}^n = \underbrace{\overline{D} \times ... \times \overline{D}}_{n}$.

Объект p, обладающий признаками $s_j \in S$, $j = \overline{1,n}$, которые соответственно имеют значения $d_j^p \in \overline{D}$, запишем как $p(s_1, s_2, ..., s_n) = (d_1^p, d_2^p, ..., d_n^p)$.

Определение 2. Объекты p и q называются *равными*, если $\forall j \; d_j^p = d_j^q$.

Иначе говоря, объекты равны, если значения соответствующих признаков этих объектов совпадают.

Определение 3. *Набором объектов* называется любое множество объектов из предметной области, рассматриваемой в задаче Z_2.

Набор P, состоящий из объектов $p_1, p_2, ..., p_r$, записывается как $P = \{p_1, p_2, ..., p_r\}$.

Определение 4. *Объектом-признаком* называется объект, для которого известно значение только одного признака из n.

Определение 5. Объект называется *нормализованным*, если известны значения всех его признаков.

Определение 6. Набор называется *нормализованным*, если все содержащиеся в нем объекты нормализованы.

Рассмотрим произвольный объект p, у которого значение признака s_j неизвестно. Если нет дополнительных ограничений на неизвестное значение признака, этот признак может принимать любое значение из D. Поэтому в данной работе используется следующая интерпретация: объект p, значение признака s_j которого неизвестно, рассматривается как набор из двух объектов, у одного из которых значение этого признака равно 0, а у другого 1; значения же всех остальных признаков совпадают со значениями соответствующих признаков p:

$$p = (..., \overset{j}{?}, ...) = \{(..., \overset{j}{0}, ...), (..., \overset{j}{1}, ...)\}.$$

Опишем процедуру нормализации произвольного набора P.

Алгоритм нормализации $Norm$:

Шаг 1. Выберем объект $p \in P$ и признак s_j такие, что $d_j^p = ?$. Если такие объект и признак выбрать невозможно, то переходим к шагу 4.

Шаг 2. Построим объекты p' и p'' по следующим правилам:

$$d_i^{p'} = \begin{cases} 0, i = j; \\ d_i^p, i \neq j; \end{cases} \quad d_i^{p''} = \begin{cases} 1, i = j; \\ d_i^p, i \neq j. \end{cases}$$

Шаг 3. Заменим в наборе P объект p на два объекта: $P := (P \setminus \{p\}) \cup \{p', p''\}$. Переходим к шагу 1.

Шаг 4. Алгоритм завершает работу.

Если необходимо нормализовать не набор, а отдельный объект, то на вход алгоритма $Norm$ подается набор, состоящий из одного этого объекта.

Исследуем свойства алгоритма нормализации.

Свойство 1. Если набор P нормализован, то $Norm(P) = P$.

Доказательство. Если набор P нормализован, то после шага 1 алгоритма $Norm$ произойдет переход на шаг 4, т. е. в результате нормализации набор P не будет изменен. ■

Свойство 2. $Norm(Norm(P)) = Norm(P)$.

Доказательство свойства 2 следует из свойства 1 и того факта, что набор $Norm(P)$ всегда является нормализованным. ■

Свойство 3. $Norm(P) \setminus Norm(Q) = Norm(P \setminus Q)$.

Свойство 4. $Norm(P) \bigcap Norm(Q) = Norm(P \bigcap Q)$.

Свойство 5. $Norm(P) \bigcup Norm(Q) = Norm(P \bigcup Q)$.

Доказательство свойств 3–5 следует из того, что процедура нормализации, применяемая к произвольному объекту, не затрагивает другие объекты. ■

Заметим, что в процессе нормализации может возникнуть ситуация, когда новый объект совпадает с существующим. Для однозначности изложения будем считать, что эти новые объекты не включаются в набор, т. е. набор не может содержать одинаковые объекты. При необходимости будем ограничиваться рассмотрением только нормализованных наборов и объектов.

2.2. Основные операции алгебры объектов

Рассмотрим основные операции над объектами и их наборами. Пусть заданы наборы P и Q . Обозначим через 0_n пустой набор, не содержащий ни одного объекта, а через 1_n – набор всех возможных объектов, обладающих n признаками из множества S .

Введем следующие операции:

1. *Отрицание.*

Отрицание $\overline{P}$ набора P определяется следующим образом:

$$\overline{P} = 1_n \setminus P .$$

2. *Умножение.*

Объекты p и q назовем *совместимыми*, если для любого признака s_j выполняется хотя бы одно из следующих условий:

а) $d_j^p = d_j^q$;

б) $d_j^p = ? \vee d_j^q = ?$.

Произведение $p \wedge q$ (или pq) совместимых объектов p и q есть объект, значения признаков которого определяются следующим образом:

$$d_j^{pq} = \begin{cases} d_j^q, & d_j^q \neq ?; \\ d_j^p, & d_j^p \neq ?; \\ ?, & d_j^p = d_j^q = ?. \end{cases}$$

Операция умножения определена только для совместимых объектов.

Произведение $P \wedge Q$ (или PQ) наборов P и Q есть набор, состоящий из всевозможных произведений совместимых объектов, один из которых входит в P, а другой в Q:

$$P \wedge Q = \bigcup_{p \in P, q \in Q} \{pq\}.$$

3. *Сложение*.

Сумма $P \vee Q$ наборов P и Q определяется по следующему правилу:

$$P \vee Q = (PQ) \cup (\overline{P}Q) \cup (P\overline{Q}).$$

В работе [5] показано, что любой набор может быть представлен в виде суммы объектов, входящих в этот набор, а любой объект, в свою очередь, может быть представлен в виде произведения его объектов-признаков:

$$P = \{p_1, ..., p_r\} = \overset{r}{\underset{i=1}{\vee}} p_i = \overset{r}{\underset{i=1}{\vee}} \overset{n}{\underset{j=1}{\wedge}} (?, ..., d_j^{p_i}, ..., ?).$$

Для краткости обозначим объекты-признаки через $d'^{p}_j = (?, ..., d_j^p, ..., ?)$. Тогда $P = \{p_1, ..., p_r\} = \overset{r}{\underset{i=1}{\vee}} p_i = \overset{r}{\underset{i=1}{\vee}} \overset{n}{\underset{j=1}{\wedge}} d'^{p_i}_j$.

Таким образом, построена алгебра объектов $G_n = < P_n, \{\neg, \wedge, \vee\} >$, основным множеством которой является множество P_n наборов объектов, обладающих n признаками, а основными операциями – операции $\neg, \wedge, \vee$ на этом множестве [6].

Введем также следующие обозначения. Пусть N – количество всех возможных объектов, обладающих n признаками. Видно, что $N = \prod_{j=1}^{n} |D| = 2^n$. Обозначим множество N-мерных двоичных векторов через C_N, а булеву алгебру N-мерных двоичных векторов – через $B_N = < C_N, \{\neg, \wedge, \vee\} >$. Покажем, что алгебры G_n и B_N изоморфны.

2.3. Изоморфизм алгебры объектов и булевой алгебры

Поставим в соответствие каждому объекту и набору объектов его код, представляющий собой последовательность нулей и единиц. Занумеруем все объекты от 1 до N. Произвольному объекту p поставим в соответствие код

$$c(p) = (\underbrace{0...0 \quad 1 \quad 0...0}_{N}),$$

где единица стоит в позиции, порядковый номер которой равен номеру объекта p.

Каждому набору объектов сопоставим код

$$c(P) = \underset{p \in P}{\vee} c(p).$$

Например, набор 0_n имеет код $C(0_n) = \underbrace{0...0}_{N}$, а набор 1_n – код $c(1_n) = \underbrace{1...1}_{N}$.

Так как во всех кодах наборов единицы стоят в различных позициях, то c устанавливает взаимно-однозначное соответствие между множеством кодов C_N и множеством наборов P_n.

Покажем, что c сохраняет операции над объектами и их наборами: код результата выполнения любой операции над набором или парой наборов равен результату соответствующей булевой операции над кодом данного набора или пары наборов, выполненной покомпонентно.

Теорема 1. *Для любых наборов P и Q справедливы равенства:*

1) $\overline{c(P)} = c(\overline{P})$;

2) $c(P) \wedge c(Q) = c(P \wedge Q)$;

3) $c(P) \vee c(Q) = c(P \vee Q)$.

Доказательство. Выберем произвольный объект $p \in P$, имеющий номер t. Проверим выполнение равенств 1–3 для соответствующих операций:

1. Отрицание:

$$c(P) = (\dots \overset{t}{1} \dots) \Leftrightarrow c(\overline{P}) = (\dots \overset{t}{0} \dots);$$

$$p \in P \Leftrightarrow p \notin \overline{P} \Leftrightarrow c(\overline{P}) = (\dots \overset{t}{0} \dots) = \overline{c(P)}.$$

2. Умножение: пусть одновременно выполняется $p \in P$ и $p \in Q$;

$$\left.\begin{array}{l} p \in P \Leftrightarrow c(P) = (\dots \overset{t}{1} \dots) \\ p \in Q \Leftrightarrow c(Q) = (\dots \overset{t}{1} \dots) \end{array}\right\} \Leftrightarrow c(P \wedge Q) = (\dots \overset{t}{1} \dots) = c(P) \wedge c(Q).$$

3. Сложение: пусть p не входит ни в P, ни в Q;

$$\left.\begin{array}{l} p \notin P \Leftrightarrow c(P) = (\dots \overset{t}{0} \dots) \\ p \notin Q \Leftrightarrow c(Q) = (\dots \overset{t}{0} \dots) \end{array}\right\} \Leftrightarrow c(P \vee Q) = (\dots \overset{t}{0} \dots) = c(P) \vee c(Q).$$

Теорема доказана. ■

Теорема 2. *Алгебра объектов G_n изоморфна булевой алгебре B_N.*

Доказательство. Чтобы показать изоморфизм алгебр, нужно построить отображение h множества P_n в множество C_N, причем h должно обладать следующими свойствами [6]:

1. Отображение h должно являться гомоморфизмом, т. е. сохранять операции отрицания, пересечения и объединения: $\overline{h(P)} = h(\overline{P})$, $h(P) \wedge h(Q) = h(P \wedge Q)$, $h(P) \vee h(Q) = h(P \vee Q)$.

2. Отображение h должно быть взаимно-однозначным.

В качестве искомого отображения можно взять отображение c, ставящее в соответствие произвольному набору его код в виде последовательности нулей и единиц длины N. По теореме 1 c – гомоморфизм; кроме того, как было показано выше, c является взаимно-однозначным, что и доказывает изоморфизм алгебр. ■

Из теоремы 2 следует эквивалентность прецедентного и логического способов представления информации. Для завершения решения подзадачи Z_{01} предложим конкретные алгоритмы перехода от логической модели представления информации к прецедентной и наоборот.

2.4. Алгоритмы преобразования моделей представления информации

Обозначим через In_{12} алгоритм перехода от логической модели к прецедентной (т. е. от I_0^1 к I_0^2), а через In_{21} – алгоритм обратного перехода. На рис. 2 показана связь между I_0^1 и I_0^2 через алгоритмы In_{12} и In_{21}.

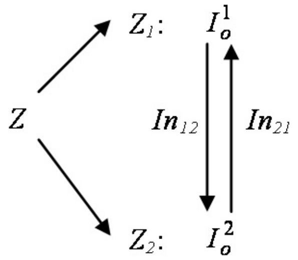

Рис. 2. Алгоритмы In_{12} и In_{21}

В данном разделе рассматриваются только нормализованные объекты. Поэтому вначале покажем, что нормализация наборов сохраняет операции над ними.

Теорема 3. *Для любых наборов P и Q справедливы равенства:*

1. $\overline{Norm(P)} = Norm(\overline{P})$.

2. $Norm(P) \wedge Norm(Q) = Norm(P \wedge Q)$.

3. $Norm(P) \vee Norm(Q) = Norm(P \vee Q)$.

Доказательство: Проверим выполнение равенств 1–3 для соответствующих операций:

1. Отрицание:

$$Norm(\overline{P}) = Norm(1_n \setminus P) = Norm(1_n) \setminus Norm(P) = 1_n \setminus Norm(P) = \overline{Norm(P)}.$$

2. Умножение.

Рассмотрим совместимые объекты p и q. Так как $d_j^{Norm(\{p\})} \neq ?$, $d_j^{Norm(\{q\})} \neq ?$, $j = \overline{1,n}$, то по определению произведения совместимых объектов $d_j^{Norm(\{pq\})} = d_j^{Norm(\{p\})} = d_j^{Norm(\{q\})}$. Из произвольности выбора j следует, что $Norm(\{pq\}) = Norm(\{p\}) \wedge Norm(\{q\})$.

$$Norm(P \wedge Q) = Norm(\bigcup_{p\in P, q\in Q}\{pq\}) = \bigcup_{p\in P, q\in Q} Norm(\{pq\}) = \bigcup_{p\in P, q\in Q}(Norm(\{p\}) \wedge Norm(\{q\})) =$$

$$= (\bigcup_{p\in P} Norm(\{p\})) \wedge (\bigcup_{q\in Q} Norm(\{q\}) = Norm(\bigcup_{p\in P}\{p\}) \wedge Norm(\bigcup_{q\in Q}\{q\}) = Norm(P) \wedge Norm(Q).$$

3. Сложение:

$$Norm(P) \vee Norm(Q) = Norm(P)Norm(Q) \cup \overline{Norm(P)}Norm(Q) \cup Norm(P)\overline{Norm(Q)} =$$

$$= Norm(PQ) \cup Norm(\overline{P}Q) \cup Norm(P\overline{Q}) = Norm(P \vee Q).$$

Теорема доказана. ■

Опишем подробнее логическую модель представления информации. В задаче Z_1 начальная информация представлена в виде логических формул, используемых для описания принадлежности объектов классам. Введем переменные $x_1, x_2, ..., x_n$, соответствующие значениям признаков $s_1, s_2, ..., s_n$, где $x_i \in D$. Множество объектов, принадлежащих классу X_i, описывается логической формулой φ, состоящей из переменных $x_1, x_2, ..., x_n$ и операций классической алгебры логики. Как принято в логике, не будем проводить различие между формулой и ее значением. Так как $x_i \in D$, то $\varphi : D^n \to D$, т. е. φ можно рассматривать как булеву функцию [7].

Рассмотрим объект $p(s_1, s_2, ..., s_n) = (d_1^p, d_2^p, ..., d_n^p)$. Пусть значение 0 на выходе φ означает, что $p \notin X_i$, а значение 1 означает, что $p \in X_i$:

$$\phi(p) = \phi(d_1^p, ..., d_n^p) = \begin{cases} 1, p \in X_i; \\ 0, p \notin X_i. \end{cases}$$

Поскольку φ – булева функция, ее можно представить в виде СДНФ [7]:

$$\varphi(x_1, ..., x_n) = \bigvee_{(d_1, ..., d_n)} x_1^{d_1} ... x_n^{d_n} \varphi(d_1, ..., d_n), \tag{1}$$

где $x_i^{d_i} = \begin{cases} x_i, d_i = 1; \\ \overline{x_i}, d_i = 0. \end{cases}$

Если $X_i \neq \varnothing$, то (1) можно сократить:

$$\varphi(x_1, ..., x_n) = \bigvee_{\substack{(d_1, ..., d_n) \\ \varphi(d_1, ..., d_n) = 1}} x_1^{d_1} ... x_n^{d_n}. \tag{2}$$

Пусть r – число элементарных конъюнкций (ЭК) в (2). Обозначим j-ю ЭК через $\varphi_j(x_1, ..., x_n) = x_1^{d_{j1}} ... x_n^{d_{jn}}$. Тогда φ можно представить следующим образом:

$$\varphi(x_1, ..., x_n) = \bigvee_{j=1}^{r} \varphi_j(x_1, ..., x_n).$$

Исходя из результатов предыдущего раздела, можно утверждать, что каждой формуле φ взаимно-однозначно соответствует набор P, представляющий собой множество объектов, описываемое формулой φ. Поэтому алгоритм In_{12} по формуле φ должен построить такой набор P, а алгоритм In_{21} по набору P – такую формулу φ, что выполняется равенство

$$\varphi(p) = 1 \Leftrightarrow p \in P. \tag{3}$$

2.4.1. Алгоритмы In_{12} и In_{21}. Опишем алгоритм In_{12}. Пусть задано правило φ в виде (2). Необходимо построить набор P, для которого выполняется (3).

Алгоритм In_{12}:

Шаг 1. Каждой ЭК $\varphi_j(x_1, ..., x_n) = x_1^{d_{j1}} ... x_n^{d_{jn}}$ в (2) поставим в соответствие объект $p_j(s_1, s_2, ..., s_n) = (d_{j1}, ..., d_{jn})$.

Шаг 2. Построим набор $P = p_1 \vee p_2 \vee ... \vee p_r$.

Шаг 3. Алгоритм завершает работу.

Опишем теперь алгоритм In_{21}. Пусть задан набор P. Необходимо указать правило φ в виде (2), для которого выполняется (3).

Алгоритм In_{21}:

Шаг 1. Каждому объекту $p_j(s_1, s_2, ..., s_n) = (d_{j1}^p, ..., d_{jn}^p)$, $p_j \in P$, поставим в соответствие ЭК $\varphi_j(x_1, ..., x_n) = x_1^{d_{j1}^p} ... x_n^{d_{jn}^p}$.

Шаг 2. Набору P поставим в соответствие ДНФ $\varphi(x_1,...,x_n) = \underset{j=1}{\overset{r}{\vee}} \varphi_j(x_1,...,x_n)$, где r – количество объектов в наборе P.

Шаг 3. Алгоритм завершает работу.

2.4.2. Свойства алгоритмов In_{12} и In_{21}. Покажем, что для алгоритмов In_{12} и In_{21} выполняется (3). Пусть $P = In_{12}(\varphi)$ обозначает, что набор P является результатом применения алгоритма In_{12} к формуле φ, а $\varphi = In_{21}(P)$ обозначает, что φ является результатом применения алгоритма In_{21} к набору P.

Теорема 4. *Для произвольных правила φ и объекта p выполняется (3):*

$$\varphi(p) = 1 \Leftrightarrow p \in P \text{ , где } P = In_{12}(\varphi),\ p(s_1, s_2,..., s_n) = (d_1^p, d_2^p,..., d_n^p).$$

Доказательство:

1. Пусть $\varphi(p) = 1$. Тогда в φ существует хотя бы одна ЭК, для которой выполняется равенство $\varphi_j(d_1^p,..., d_n^p) = 1$. Для ЭК φ_j на шаге 1 построен объект $p(s_1, s_2,..., s_n) = (d_1^p, d_2^p,..., d_n^p)$, который на шаге 2 включен в набор P, что и требовалось доказать.

2. Пусть $p \in P$. На шаге 1 объект p поставлен в соответствие ЭК $\varphi_j(d_1^p,..., d_n^p) = = x_1^{d_1^p} ... x_n^{d_n^p}$. Так как ЭК φ_j входит в φ, то $\varphi(p) = \varphi(d_1^p,..., d_n^p) = 1$. ■

Теорема 5. *Для произвольных набора P и объекта p выполняется (3):*

$$p \in P \Leftrightarrow \varphi(p) = 1 \text{, где } \varphi = In_{21}(P),\ p(s_1, s_2,..., s_n) = (d_1^p, d_2^p,..., d_n^p).$$

Доказательство. Видно, что формула $\varphi = In_{21}(P)$ имеет вид (2).

1. Пусть $p \in P$. На шаге 1 объекту p поставлена в соответствие ЭК $\varphi_j(d_1^p,..., d_n^p) = 1$. На шаге 2 построена ДНФ $\varphi(d_1^p,..., d_n^p) = \underset{j=1}{\overset{r}{\vee}} \varphi_j(d_{j1}^p,..., d_{jn}^p) = 1$, что и требовалось.

2. Пусть $\varphi(p) = 1$. Тогда в φ существует хотя бы одна ЭК, для которой выполняется равенство $\varphi_j(d_1^p,..., d_n^p) = 1$. На шаге 1 ЭК φ_j поставлена в соответствие объекту $p(s_1, s_2,..., s_n) = (d_1^p, d_2^p,..., d_n^p)$. Следовательно, $p \in P$. ■

Следствие. *Преобразования, производимые алгоритмами In_{12} и In_{21}, взаимно обратны: для произвольных правила φ, набора P и объекта p выполняются следующие свойства:*

1) $\varphi(p) = 1 \quad \Leftrightarrow \quad (In_{21} \circ In_{12}(\varphi))(p) = 1$;

2) $p \in P \quad \Leftrightarrow \quad p \in (In_{12} \circ In_{21}(P))$.

Доказательство.

1. $\varphi(p) = 1 \Leftrightarrow$ [по теореме 4] $\Leftrightarrow p \in In_{12}(\varphi) \Leftrightarrow$ [по теореме 5] $\Leftrightarrow (In_{21} \circ In_{12}(\varphi))(p) = 1$.

2. $p \in P \Leftrightarrow$ [по теореме 5] $\Leftrightarrow \varphi(d_1^p,..., d_n^p) = 1 \Leftrightarrow$ [по теореме 4] $\Leftrightarrow p \in (In_{12} \circ In_{21}(P))$. ■

2.5. Задача Z_3

Итак, решена подзадача Z_{01}: установлена эквивалентность прецедентной и логической моделей представления информации и разработаны алгоритмы для перехода между этими моделями. Преобразуем задачу Z_1: представим начальную информацию I_0^1, заданную логическим способом, в прецедентном виде (т. е. в том же виде, в котором представлена начальная информация I_0^2). Эту новую задачу обозначим через Z_3, а ее исходные данные и алгоритм решения – через I_0^3 и A_3 соответственно. Связь между задачами Z_1, Z_2 и Z_3 показана на рис. 3.

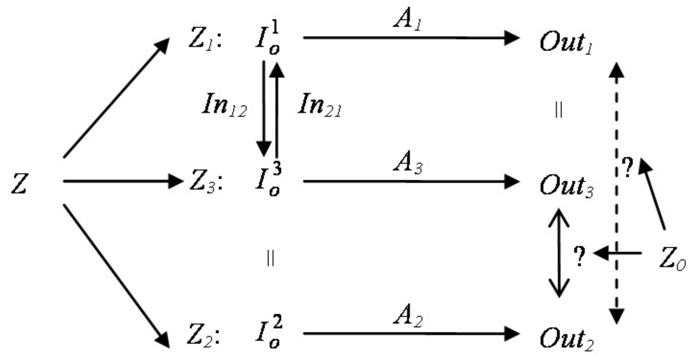

Рис. 3. Схема связи задач Z, Z_1, Z_2, Z_3, Z_0

С одной стороны, задача Z_3 представляет собой задачу Z_1 с преобразованными исходными данными, поэтому задача Z_0 сводится к сравнению решений задач Z_2 и Z_3. С другой стороны, так как информация I_0^3 представлена тем же способом, что и I_0^2, для решения задачи Z_3 может быть применен алгоритм A_2, а для решения задачи Z_2 – алгоритм A_3. Поэтому в дальнейшем будем отождествлять задачи Z_2 и Z_3 и говорить только о задаче Z_2, подразумевая задачу, начальная информация в которой задана прецедентным способом.

3. Сравнение алгоритмов решения задачи Z_2

В данном разделе рассмотрим алгоритм A_3, представляющий собой модификацию метода резолюций для решения задачи Z_2. Этот модифицированный метод назовем *методом объектных резолюций.* Для решения подзадачи Z_{02} опишем параметрическое семейство алгоритмов, предложенное в [3], и сравним его с алгоритмом A_3.

3.1. Метод объектных резолюций

Рассмотрим объекты p и q и признак s_k, такие, что $d_k^p = 1$, $d_k^q = 0$. Будем считать, что для объектов p и q признак s_k единственный (т. е. значения остальных признаков либо совпадают, либо не заданы хотя бы в одном объекте). *Объектной резольвентой* назовем объект r, значения признаков которого определяются по следующему правилу:

$$d_j^r = \begin{cases} ?, j = k; \\ d_j^p \wedge d_j^q, j \neq k. \end{cases}$$

Через Or обозначим операцию построения объектной резольвенты: $r = Or(p,q)$.

Покажем, что операция построения объектной резольвенты эквивалентна операции построения резольвенты в классическом методе резолюций в следующем смысле: результаты применения метода объектных резолюций к объектам и классического метода резолюций к формулам, описывающим эти объекты, совпадают.

Теорема 6. *Пусть для объектов p и q $\exists! s_k$, $d_k^p = 1$, $d_k^q = 0$. Пусть t – объект, соответствующий обычной резольвенте для формул, описывающих p и q; $r = Or(p,q)$. Тогда $t = r$.*

Доказательство. Представим p и q в виде конъюнктов значений их объектов-признаков, как показано в [5], обозначив эти представления через P' и Q' соответственно:

$$P' = d'^p_1 \wedge \ldots \wedge d'^p_k \wedge \ldots \wedge d'^p_n, \; Q' = d'^q_1 \wedge \ldots \wedge d'^q_k \wedge \ldots \wedge d'^q_n .$$

Чтобы применить к P' и Q' классический метод резолюций, преобразуем их в КНФ:

$$P' = \overline{\overline{d'^p_1} \vee \ldots \vee \overline{d'^p_k} \vee \ldots \vee \overline{d'^p_n}}, \; Q' = \overline{\overline{d'^q_1} \vee \ldots \vee \overline{d'^q_k} \vee \ldots \vee \overline{d'^q_n}} \Rightarrow$$
$$\Rightarrow \overline{P'} = \overline{d'^p_1} \vee \ldots \vee \overline{d'^p_k} \vee \ldots \vee \overline{d'^p_n}, \; \overline{Q'} = \overline{d'^q_1} \vee \ldots \vee \overline{d'^q_k} \vee \ldots \vee \overline{d'^q_n} .$$

Так как $\overline{d^p_k} = 0$, $\overline{d^q_k} = 1$, для $\overline{P'}$ и $\overline{Q'}$ можно построить резольвенту

$$R = \overline{d'^p_1} \vee \ldots \vee \overline{d'^p_k} \vee \ldots \vee \overline{d'^p_n} \vee \overline{d'^q_1} \vee \ldots \vee \overline{d'^q_k} \vee \ldots \vee \overline{d'^q_n} .$$

Построим объект t, которому соответствует R:

$$R = \overline{d'^t_1} \vee \ldots \vee \overline{d'^t_{k-1}} \vee \overline{d'^t_{k+1}} \vee \ldots \vee \overline{d'^t_n} \Rightarrow \overline{R} = \overline{\overline{d'^t_1} \vee \ldots \vee \overline{d'^t_{k-1}} \vee \overline{d'^t_{k+1}} \vee \ldots \vee \overline{d'^t_n}} =$$
$$= d'^t_1 \wedge \ldots \wedge d'^t_{k-1} \wedge d'^t_{k+1} \wedge \ldots \wedge d'^t_n \Rightarrow t(s_1, \ldots, s_n) = (d^t_1, \ldots, d^t_{k-1}, ?, d^t_{k+1}, \ldots, d^t_n) .$$

Сравним значения признаков t со значениями признаков r:

$$d^t_j = \begin{cases} ?, \; j = k \\ \overline{\overline{d'^p_j} \vee \overline{d'^q_j}} = d^p_j \wedge d^q_j, \; j \neq k \end{cases} = d^r_j, \; j = \overline{1,n} .$$

Следовательно, $d^t_j = d^r_j$ для всех $s_j \in S$. Поэтому $t = r$. ■

3.2. Алгоритмическая реализация метода объектных резолюций

Зафиксируем номер i класса X_i и определим, принадлежит ли объект x этому классу.

Алгоритм объектных резолюций A_3:

Шаг 1. Введем множество $Y = X^0_i$.

Шаг 2. Если $x \in Y$, то переходим к шагу 6, иначе – к шагу 3.

Шаг 3. Если все пары объектов уже рассматривались, то переходим к шагу 6. Иначе выбираем из Y такую нерассмотренную пару объектов p и q, что $\exists! s_k, d^p_k = 1$, $d^q_k = 0$.

Шаг 4. Вычисляем $r = Or(p, q)$.

Шаг 5. Добавляем к Y объект r: $Y := Y \bigcup \{r\}$. Возвращаемся на шаг 2.

Шаг 6. Алгоритм завершает работу.

Алгоритм A_3 можно применять как для прямого, так и для обратного вывода. В последнем случае на шаге 1 вводится множество $Y = X^0_i \bigcup \{x\}$, а в качестве объекта x рассматривается объект, значения всех признаков которого неизвестны: $o(s_1, \ldots, s_n) = (? \ldots ?)$. В зависимости от типа вывода результат работы алгоритма A_3 можно интерпретировать следующим образом:

1. Прямой вывод: если алгоритм закончил работу из-за получения объекта x, это значит, что, заменяя в описании множества X^0_i знаки «?» на конкретные значения признаков, можно получить x. Поэтому объект x принадлежит классу X_i.

2. Обратный вывод: если алгоритм закончил работу после шага 2 из-за получения объекта o, это значит, что, заменяя знаки «?» на конкретные значения признаков, из множества Y можно получить любой объект, т. е. Y потенциально содержит все возможные объекты из X. Поэтому объект x не может принадлежать классу X_i.

Если ни один из результатов 1 и 2 не получен, с помощью данного алгоритма нельзя сделать никаких выводов о принадлежности объекта x классу X_i.

Итак, алгоритм A_3 представляет собой алгоритм A_1, преобразованный для прецедентной модели представления начальной информации. В процессе решения подзадачи Z_{01} была установлена эквивалентность прецедентной и логической моделей. Поэтому не будем проводить различие между алгоритмами A_1 и A_3 и при решении оставшихся подзадач задачи Z_0 в качестве алгоритма A_1 используем алгоритм A_3.

3.3. Семейство алгоритмов распознавания

Опишем вкратце параметрическое семейство алгоритмов, предложенное в [3].

Введем предварительно следующую функцию $\mu : X^2 \to [0,1]$, характеризующую близость объекта x к объекту обучающей выборки $y \in X_i^0$:

$$\mu(x,y) = \max\{0, (\sum_{j\in S}(-1)^t a_{ij}) \cdot (\sum_{j\in S} a_{ij})^{-1}\},$$

где $\|a_{ij}\|$ – матрица, поставленная в соответствие множеству $\{1,...,l\}\times|S|$ и удовлетворяющая следующим ограничениям: $a_{ij}\in R$, $\forall i,j \ (a_{ij}\geq 0)$, $\forall i \ (\sum_j a_{ij} > 0)$; l – число классов; S – множество признаков всех объектов; $t = \begin{cases} 1, & d_j^x \neq d_j^y; \\ 2, & d_j^x = d_j^y. \end{cases}$

В определении функции предполагается, что все объекты нормализованы. В [3] предложена также одна из возможных схем вычисления параметров $\|a_{ij}\|$.

Опишем теперь алгоритм распознавания A_2, применяемый для решения задачи Z_2:

Шаг 1. Для всех объектов $x\in X$ и $x^0 \in X^0$ вычислим $\mu(x,x^0)$.

Шаг 2. Для всех $i=\overline{1,l}$ вычислим $P_i^{A_2}(x) = \max\limits_{x_j^0\in X_i^0}\{\mu(x,x_j^0)\}$.

Шаг 3. Алгоритм завершает работу.

Видно, что $P_i^{A_2}(x)\in[0,1]$, $i=\overline{1,l}$. В данном случае предполагается, что $P_i^{A_2}(x)$ отражает близость объекта x к классу X_i; в частности, если $P_i^{A_2}(x)=1$, то $x\in X_i$.

3.4. Сравнение алгоритмов объектных резолюций и распознавания

Введем систему предикатов $P_1,...,P_l$, характеризующих принадлежность объектов к классам $X_1,...,X_l$, как это сделано в [8]:

$$\forall x\in X \quad (P_i(x)\in\{0,1\} \wedge (P_i(x)=1 \Leftrightarrow x\in X_i)).$$

Как правило, точные значения этих предикатов известны только для объектов из обучающей выборки X_0; задача Z_2 состоит именно в том, чтобы определить значения предикатов $P_1,...,P_l$ для остальных объектов.

По результатам работы алгоритмов A_2 и A_3 сформируем классификационные векторы $A_2(x) = (P_1^{A_2}(x),..., P_l^{A_2}(x))$, $A_3(x) = (P_1^{A_3}(x),..., P_l^{A_3}(x))$. Предварительно приведем результаты алгоритма A_3 к численному виду. Определим $P_i^{A_3}(x)$ следующим образом:

$$P_i^{A_3}(x) = \begin{cases} 1, x \in X_i; \\ -1, x \notin X_i. \end{cases}$$

В определении $P_i^{A_3}$ решение о принадлежности объекта x классу X_i принимается алгоритмом A_3.

Традиционно в теории распознавания в качестве показателя эффективности алгоритмов используется функционал качества, основанный на доле правильно классифицированных объектов [1]. Введем такой функционал качества, значения которого легко интерпретировались бы в терминах совпадения или близости $P_i(x)$ и $P_i^A(x)$. Рассмотрим следующий функционал для произвольного алгоритма A, решающего задачу Z_2 на множестве $X' \subseteq X$:

$$\Phi_A(X') = 1 - \frac{1}{l}\frac{1}{|X'|}\sum_{i=1}^{l}\sum_{x \in X'}(\left|P_i(x) - P_i^{A_j}(x)\right|).$$

Функционал Φ_A является частным случаем функционала, рассматриваемого в [8].

Из определения Φ_A следует, что чем ближе значение $\Phi_A(X')$ к 1, тем точнее алгоритм A работает на множестве X'; в частности, если $\Phi_A(X') = 1$, то $\forall x \in X'\ P_i^A(x) = P_i(x)$. Поэтому из нескольких алгоритмов предпочтительным будем считать тот, который имеет наибольшее значение функционала Φ_A.

Теорема 7. *Пусть* $Y_i = \{x \mid P_i^{A_3}(x) = 1\}$, $Y = \bigcup_{i=1}^{l} Y_i$. *Тогда*

1) $\Phi_{A_2}(Y) \le \Phi_{A_3}(Y)$*;*

2) $\Phi_{A_3}(X \setminus Y) \le \Phi_{A_2}(X \setminus Y)$.

Доказательство:

1. Если $x \in Y_i$, то $P_i^{A_3}(x) = P_i(x) = 1 \Rightarrow \left|P_i(x) - P_i^{A_3}(x)\right| = 0 \Rightarrow \sum_{x \in Y}(\left|P_i(x) - P_i^{A'}(x)\right|) = 0 \Rightarrow$

$\Rightarrow \Phi_{A_3}(Y) = 1 \Rightarrow \Phi_{A_2}(Y) \le 1 = \Phi_{A_3}(Y)$.

2. Если $x \in X \setminus Y$, то $\forall i = \overline{1,l}\ \ P_i^{A_3}(x) = -1$, $P_i(x) \in \{0,1\} \Rightarrow \forall i = \overline{1,l}\ \left|P_i(x) - P_i^{A_3}(x)\right| \ge 1 \Rightarrow$

$\Rightarrow \frac{1}{|X \setminus Y|}\sum_{x \in X \setminus Y}(\left|P_i(x) - P_i^{A'}(x)\right|) \ge 1 \Rightarrow \Phi_{A_3}(X \setminus Y) \le 0 \Rightarrow \Phi_{A_3}(X \setminus Y) \le 0 \le \Phi_{A_2}(X \setminus Y)$. ■

Из теоремы 7 можно сделать вывод, что на множестве Y предпочтительно использовать алгоритм A_3, а на множестве $X \setminus Y$ – алгоритм A_2. Таким образом, подзадача Z_{02} решена.

4. Комбинированный алгоритм

Перейдем к подзадаче Z_{03}. При сравнении алгоритмов было показано, что на некотором множестве наиболее точные результаты дает алгоритм A_3, а на дополнении этого множества – алгоритм A_2. Чтобы достичь наилучших результатов, построим комбинированный алгоритм A_C, объединяющий A_2 и A_3, и покажем, что $\Phi_{A_C}(X)$ не меньше, чем $\Phi_{A_2}(X)$ и $\Phi_{A_3}(X)$.

4.1. Описание комбинированного алгоритма

Введем алгоритм A_C, являющийся объединением алгоритмов A_2 и A_3. Алгоритмы объектных резолюций и распознавания были описаны выше, поэтому не будем приводить их в подробностях, а лишь включим их в описание алгоритма A_C как отдельные шаги.

Комбинированный алгоритм A_C:

Шаг 1. Выберем объект $x \in X$.

Шаг 2. Выберем номер класса $i = \overline{1,l}$.

Шаг 3. Применим алгоритм объектных резолюций A_3 к объекту x и множеству X_i^0.

Шаг 4. Если $P_i^{A_3}(x) = 1$, то переходим к шагу 7. Иначе переходим к шагу 5.

Шаг 5. Если все классы исчерпаны, то переходим к шагу 6. Иначе переходим к шагу 2.

Шаг 6. Применим алгоритм распознавания A_2 к объекту x.

Шаг 7. Если все объекты исчерпаны, то переходим к шагу 8. Иначе переходим к шагу 1.

Шаг 8. Алгоритм завершает работу.

Результаты алгоритма A_C для объекта x также представим в виде классификационного вектора $A_C(x) = (P_1^{A_C}(x),\ldots,P_l^{A_C}(x))$, где $P_i^{A_C}(x) \in [0,1]$.

Замечание. В алгоритме A_C алгоритм объектных резолюций используется для прямого вывода. Иногда целесообразно использовать обратный вывод; например, это ситуация, когда $l = 2$, т. е. требуется определить, принадлежит ли объект классу или его дополнению. В этом случае нужно на шаге 3 применить алгоритм A_3 дважды: для прямого и обратного вывода.

4.2. Сравнение алгоритмов

Докажем предварительно одно свойство функционала Φ_A, которое используем при исследовании комбинированного алгоритма.

Свойство функционала Φ_A. Если $X = Y \bigcup Z$, где $Y \bigcap Z = \varnothing$, то

$$|X|\Phi_A(X) = |Y|\Phi_A(Y) + |Z|\Phi_A(Z). \tag{4}$$

Доказательство. Так как $Y \bigcap Z = \varnothing$, то $|X| = |Y| + |Z|$. Тогда по определению Φ_A

$$(|Y| + |Z|)\Phi_A(Y \cup Z) = (|Y| + |Z|) - \frac{1}{l}\sum_{i=1}^{l}\sum_{x\in Y\bigcup Z}(|P_i(x) - P_i^A(x)|) =$$

$$= |Y| - \frac{1}{l}\sum_{i=1}^{l}\sum_{x\in Y}(|P_i(x) - P_i^A(x)|) + |Z| - \frac{1}{l}\sum_{x\in Z}(|P_i(x) - P_i^A(x)|) = |Y|\Phi_A(Y) + |Z|\Phi_A(Z),$$

откуда следует (4). ■

Покажем теперь, что функционалы качества алгоритмов A_2 и A_3 не превосходят функционала качества алгоритма A_C на множестве X.

Теорема 8. $\Phi_{A_C}(X) \ge \max\{\Phi_{A_2}(X), \Phi_{A_3}(X)\}$.

Доказательство. Пусть $Y_i = \{x \mid P_i^{A_3}(x) = 1\}$, $Y = \bigcup_{i=1}^{l} Y_i$. Если $x \in Y$, то на шаге 3 алгоритма A_C получен результат $P_i^{A_C}(x) = P_i^{A_3}(x)$, т. е. $\Phi_{A_C}(Y) = \Phi_{A_3}(Y)$. Если же $x \in X \setminus Y$, то на шаге 6 алгоритма A_C получен результат $P_i^{A_C}(x) = P_i^{A_2}(x)$, т. е. $\Phi_{A_C}(X \setminus Y) = \Phi_{A_2}(X \setminus Y)$.

Используя свойство (4), получаем

$$|X|\Phi_{A_C}(X)=|Y|\Phi_{A_C}(Y)+|X\setminus Y|\Phi_{A_C}(X\setminus Y)=|Y|\Phi_{A_3}(Y)+|X\setminus Y|\Phi_{A_2}(X\setminus Y).$$

Покажем, что $\Phi_{A_C}(X)\geq\Phi_{A_2}(X)$ и $\Phi_{A_C}(X)\geq\Phi_{A_3}(X)$:

$$|X|(\Phi_{A_C}(X)-\Phi_{A_2}(X))=|Y|(\Phi_{A_3}(Y)-\Phi_{A_2}(Y))\geq 0 \text{ (по теореме 7);}$$

$$|X|(\Phi_{A_C}(X)-\Phi_{A_3}(X))=|X\setminus Y|(\Phi_{A_2}(X\setminus Y)-\Phi_{A_3}(X\setminus Y))\geq 0 \text{ (по теореме 7).}$$

Отсюда получаем, что $\Phi_{A_C}(X)\geq\max\{\Phi_{A_2}(X),\Phi_{A_3}(X)\}$. ■

Из доказательства теоремы 8 следует, что $\Phi_{A_C}(X)>\max\{\Phi_{A_2}(X),\Phi_{A_3}(X)\}$ тогда и только тогда, когда одновременно $\Phi_{A_3}(Y)>\Phi_{A_2}(Y)$ и $\Phi_{A_2}(X\setminus Y)>\Phi_{A_3}(X\setminus Y)$.

Главное преимущество алгоритма A_C по сравнению с A_3 заключается в том, что A_3 позволяет получить результат лишь для тех объектов, описание которых можно вывести из описаний множеств X_i^0, а алгоритм A_C дает оценки для всех объектов из X. Однако для объектов, принадлежность которых определена на этапе применения A_3, результат является обоснованным, в чем заключается основное преимущество A_C по сравнению с A_2.

Таким образом, подзадача Z_{03} решена: построен алгоритм, объединяющий алгоритмы объектных резолюций и распознавания и работающий на X не хуже обоих этих алгоритмов.

Заключение

В работе рассмотрены два подхода к решению задач распознавания, имеющие дедуктивную и индуктивную природу. В задачах, решаемых при помощи этих подходов, используются соответственно логическая и прецедентная модели представления информации. Показано, что в булевом пространстве при определенных ограничениях эти модели эквивалентны. Предложен метод объектных резолюций как модификация метода резолюций для задач распознавания. Разработан комбинированный алгоритм, объединяющий метод объектных резолюций с семейством алгоритмов распознавания, который работает не хуже любого из алгоритмов, образующих комбинацию.

На текущем этапе работа носит теоретический характер. Однако предшествующие исследования по теме статьи уже нашли практическое применение. Например, результаты, полученные в [3], использовались при разработке интеллектуальной системы поддержки решений в спортивной травматологии, описанной в [9].

В качестве следующего этапа исследования предполагается обобщение рассмотренной в статье задачи для случаев, когда признаки объектов принимают произвольное число значений. Планируется также рассмотрение комбинаторных характеристик предлагаемого алгоритма, его численных оценок, особенностей практического использования.

Список литературы

1. Журавлев, Ю.И. Об алгебраическом подходе к решению задач распознавания или классификации / Ю.И. Журавлев // Проблемы кибернетики. – 1978. – № 33. – С. 5–68.

2. Журавлев, Ю.И. Корректные алгебры над множествами некорректных (эвристических) алгоритмов / Ю.И. Журавлев // Кибернетика. – 1977. – № 4. – С. 14–21.

3. Краснопрошин, В.В. Распознавание с обучением как задача выбора / В.В. Краснопрошин, В.А. Образцов // Цифровая обработка изображений. – Минск : Ин-т техн. кибернетики НАН Беларуси, 1998. – С. 80–94.

4. Чень, Ч. Математическая логика и автоматическое доказательство теорем / Ч. Чень, Р. Ли. – М. : Наука, 1983. – 360 с.

5. Рябцев, А.В. Алгебры для представления обучающей информации в задачах распознавания образов / А.В. Рябцев // Цифровая обработка. – 2002. – № 6. – С. 80–94.

6. Мальцев, А.И. Алгебраические системы / А.И. Мальцев. – М. : Наука, 1970.

7. Яблонский, С.В. Введение в дискретную математику / С.В. Яблонский – М. : Наука, 1986. – 384 с.

8. Краснопрошин, В.В. Проблема принятия решений по прецедентности: разрешимость и выбор алгоритмов / В.В. Краснопрошин, В.А. Образцов // Выбраныя навуковыя працы Беларускага дзяржаўнага універсітэта. Т. 6. Матэматыка. – Минск : БГУ, 2001. – С. 285–312.

9. Интеллектуальная система поддержки решений в спортивной травматологии / В.В. Краснопрошин [и др.] // Вестник Национального технического университета «ХПИ». Тематический выпуск: информатика и моделирование. – 2010. – № 31. – С. 106–111.

Поступила 17.04.12

Белорусский государственный университет,
Минск, пр. Независимости, 4
e-mail: olgashut@tut.by

O.V. Shut

A SOLUTION METHOD OF RECOGNITION IN BOOLEAN SPACE BASED ON LOGICAL AND PRECEDENT MODELS

The paper considers deductive and inductive approaches to recognition problems. These approaches use logical and precedent related information encoding models, respectively. Algebra of objects, which is isomorphic to Boolean algebra, is proposed. Algorithms for conversion of information encoding models are developed. A modification of the resolution method for the precedent model is suggested. A new algorithm, which combines the resolution method and recognition algorithms, is developed.