УДК 004.8
ББК 32.813

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ НА ОСНОВЕ КОЛОНОК

Чесноков А. М.[1]
(*ФГБУН Институт проблем управления им. В.А.Трапезниникова РАН, Москва*)

В статье рассматриваются интеллектуальные системы на основе колонок. Приводятся основные понятия и определения. Формулируются прямая и обратная задачи. Приводятся решения этих задач и оценки вычислительной эффективности. На этой основе предлагается решение задачи классификации и доказывается возможность реализации произвольных булевых функций. В конце дается оценка классов задач, которые могут решаться с помощью интеллектуальных систем на основе колонок.

Ключевые слова: искусственный интеллект, интеллектуальные системы на основе колонок, колонка.

1. Введение

Начало работ над рассматриваемой моделью интеллектуальных систем связано с публикациями [4] и [5], которые, с одной стороны, помогли составить примерные представления об общем облике систем на основе колонок, с другой, позволили выделить элементарные задачи, без решения которых невозможно функционирование таких систем, а также помогли остановить свой выбор на простом математическом объекте, используемом в качестве колонки. Вообще, простота, высокий уровень общности и невведение новых понятий и ограничений до тех

[1] *Александр Михайлович Чесноков, старший научный сотрудник, кандидат технических наук* (*alex-ches@yandex.ru*).

пор, пока в этом не возникает настоятельная необходимость, – это те положения, которые в значительной мере определяли процесс построения модели. Полученная в результате модель систем на основе колонок чрезвычайно проста.

Имеется пусть и очень большое, но конечное множество имен, которые используются для наименования объектов произвольной природы. В этом множестве выделяются непересекающиеся подмножества, получившие название областей имен. В реальных задачах выделение областей может быть связано, например, с типизацией или с выделением «миров» (domains), как в некоторых реализациях Пролога. Одной из важнейших причин выделения областей является требование обеспечить отсутствие случайных совпадений имен в различных частях большой системы и тем самым исключить возможность возникновения соответствующих ошибок. Для рассматриваемой модели не важно, какие причины могут приводить к выделению областей. Главное, эти причины существуют и это нашло отражение в модели.

Под образом понимается любое конечное множество имен, принадлежащих тем или иным областям имен. Образы, входящие в любое множество образов, могут быть наименованы при помощи имен некоторой области, т.е. каждому образу из множества может быть сопоставлено его имя. Упорядоченная пара (имя образа, образ) получила название колонки.

Система на основе колонок представляет собой одно или несколько множеств колонок и работающий с ними механизм (машина колонок), который, получая информацию о внешнем мире в виде образов, формирует новые колонки, изменяет или удаляет уже существующие и выполняет другие необходимые операции. В основе процесса накопления знаний в подобных системах лежит запоминание новых образов под определенными именами. При этом элементарными базовыми задачами, без которых невозможно функционирование системы, являются прямая задача – по образу получить его имя, и обратная – по имени получить соответствующий образ.

Для данной модели систем на основе колонок сразу возникают три основных вопроса:

1. Что из себя представляют множества колонок, с которыми придется иметь дело системе?
2. Существуют ли методы решения базовых задач?
3. Какие вообще задачи могут решать такие простые системы?

Ответу на эти вопросы и посвящена данная работа.

В разделе 2 приводятся основные понятия и определения, относящиеся к системам на основе колонок: имя, образ, колонка, индекс как множество колонок. Рассматриваются первичные колонки и колонки с классами. Показана связь между индексом как множеством колонок и обычным индексом, заданным в виде таблицы ссылок. В разделе 3 сначала показывается существование решения базовых задач для любого типа образов. Затем для образов в виде конечных неупорядоченных множеств и конечных последовательностей или векторов предлагается более эффективный метод решения. Наконец в разделе 4 показано как базовые задачи используются для решения более сложных задач. Сначала рассматривается задача классификации, затем доказывается возможность реализации в системах на основе колонок произвольных булевых функций. На основе этих результатов дается оценка тех классов задач, для которых могут использоваться интеллектуальные системы на основе колонок.

2. Основные понятия

2.1. ИМЕНА, ОБРАЗЫ, КОЛОНКИ

Рассматривается конечное множество U – *множество имен* некоторых объектов произвольной природы. Не ограничивая общности, будем считать, что оно является подмножеством множества целых чисел Z. Множество U конечное, однако никаких ограничений на его размеры не накладывается. При необходимости его всегда можно увеличить, добавив необходимое число элементов.

В множестве имен U выделяются непересекающиеся подмножества имен, получившие название *областей имен*. При необходимости любая область имен может быть в любой момент расширена, а также может быть введена новая область имен.

Любое конечное множество имен, принадлежащих тем или иным областям имен, называется *образом*.

Рассмотрим некоторое конечное множество образов P. Его можно перенумеровать, используя для этого элементы некоторой области имен U', $|U'| = |P|$:

$P = \{p_i \mid i \in U'\}$,

где $|\cdot|$ – мощность множества. То, что образы множества P перенумерованы, означает, что установлено взаимно однозначное соответствие $\varphi: i \leftrightarrow p_i$ между областью имен U' и множеством образов P.

Колонкой называется упорядоченная пара (i, p_i), где i – имя колонки, $p_i = \varphi(i)$ – образ, содержащийся в колонке. Колонка обозначается как $(i \mid p_i)$. Также будет использоваться обозначение $i \to p_i$. В этом случае будем говорить, что имя колонки i является *ссылкой* или *указателем* на содержащийся в колонке образ p_i. В свою очередь, часто будет говориться, что образ p_i имеет имя i или известен под именем i. Под этим будет подразумеваться то, что образ p_i содержится в колонке $(i \mid p_i)$. Отображение $\varphi: i \to p_i$ будет называться *отображением наименования*.

2.2. ПЕРВИЧНЫЕ КОЛОНКИ

В образы колонок могут входить имена других колонок, т.е. имя в образе одной колонки может быть именем другой колонки и служить указателем на соответствующий образ. В образ также могут входить имена, которые еще не использовались для наименования ни одного образа. Такие не связанные ни с одним образом имена будут называться *чистыми* или *пустыми именами*. Положив, что чистое имя ссылаются на пустое множество, его можно рассматривать как колонку вида $(i \mid \varnothing)$ или $i \to \varnothing$. Таким образом, можно считать, что в образе одной колонки содержатся имена других колонок, каждое из которых служит указателем на соответствующий

образ, возможно, пустой. В результате образуется показанная на рис. 1 сложная структура колонок (для наглядности на рисунке дублируются имена колонок).

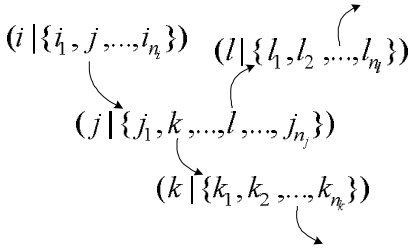

Рис. 1. Структура из колонок

Рассматриваемые простейшие колонки называются *первичными колонками*. Множество первичных колонок будет обозначаться как C^1.

2.3. ФАКТОРИЗАЦИЯ КОЛОНОК, КОЛОНКИ С КЛАССАМИ

Отличительной особенностью первичных колонок C^1 является взаимно однозначное соответствие между единственным именем и единственным образом. Однако реально может возникнуть необходимость в том, чтобы образ имел несколько имен. Следовательно, должны иметься несколько колонок с разными именами и одним и тем же образом. Если рассмотреть отображение наименования $\varphi: i \to p$, которое имени ставит в соответствие образ, то все эти имена образуют класс эквивалентности, состоящий из имен с одним и тем же образом, т.е. имеет место *факторизация колонок по образу*. Взаимно однозначное соответствие между именем колонки и образом сохраняется, но как взаимно однозначное соответствие класса имен, указывающих на один и тот же образ: $[i] \leftrightarrow p$, где $[i]$ – класс эквивалентных имен. Отображение наименования $\varphi: [i] \to p$ определяется как $\varphi([i]) = \varphi(i_k)$ для $\forall i_k \in [i]$. Таким образом, кроме колонок C^1 необходимо также рассматривать и колонки вида $([i] \mid p_i)$:

$$\begin{array}{ccc} p & \dots & p \\ \uparrow & \dots & \uparrow \\ i_1 & \dots & i_m \end{array} \quad \Rightarrow \quad \begin{array}{c} p \\ \uparrow \\ [i] = \{i_1, \dots, i_m\} \end{array}$$

Пусть теперь имеет место противоположный случай, когда несколько образов имеют одно и то же имя. Теперь класс эквивалентности возникает по отображению $\varphi^{-1}: p \to i$, которое образу ставит в соответствие его имя. В класс входят все образы с одним и тем же именем, т.е. имеет место *факторизация колонок по имени*. Так же, как и в предыдущем случае, взаимно однозначное соответствие сохраняется. Однако на этот раз в виде взаимно однозначного соответствия класса образов, имеющих одно и то же имя: $i \leftrightarrow [p]$, где $[p]$ – класс эквивалентных образов. Отображение $\varphi^{-1}: [p] \to i$ определяется как $\varphi^{-1}([p]) = \varphi^{-1}(p_k)$ для $\forall p_k \in [p]$. Таким образом, кроме колонок C^1 также должны рассматриваться и колонки вида $(i \mid [p])$:

$$\begin{array}{ccc} p_1 & \dots & p_m \\ \uparrow & \dots & \uparrow \\ i & \dots & i \end{array} \quad \Rightarrow \quad \begin{array}{c} [p] = \{p_1, \dots, p_m\} \\ \uparrow \\ i \end{array}$$

Объединив два рассмотренных выше случая факторизации, получим колонки вида $([i] \mid [p_i])$. Множество таких колонок будет называться *множеством колонок с классами* и обозначаться через C^2. Колонки множества C^2 – это колонки, имена которых могут представлять классы эквивалентных имен, а состоящие из них образы – классы эквивалентных образов. Очевидно, $C^1 \subset C^2$.

Колонки множества C^2 можно привести к виду, когда образы колонок не являются классами эквивалентности, т.е. когда классы эквивалентности встречаются только среди имен.

Пусть имеется некоторая колонка $(i \mid [p]) \in C^2$, где образ колонки – это класс $[p] = \{p_1, \dots, p_m\}$, $m > 1$. Используем m чистых имен i_k для того, чтобы наименовать все образы p_k, входящие в класс эквивалентности $[p]$. Получим m колонок $(i_k \mid p_k)$. Сформируем образ $\{i_1, \dots, i_m\}$, состоящий из имен этих колонок. Тогда исходную колонку $(i \mid [p])$ можно заменить $m + 1$ колонкой $(i \mid \{i_1, \dots, i_m\})$, $(i_k \mid p_k)$, $k = 1, \dots, m$:

$$i \to [p] = i \to \{p_1, \ldots, p_m\} \quad \Rightarrow \quad \overset{\overset{p_1}{\uparrow} \quad \overset{p_m}{\uparrow}}{i \to \{i_1, \ldots, i_m\}}$$

На месте колонки $(i \mid [p])$ появилась первичная колонка $(i \mid \{i_1, \ldots, i_m\})$ и m колонок $(i_k \mid p_k)$. Действуя аналогичным образом, можно привести конечное множество колонок C^2 к виду, когда классы эквивалентности будут встречаться только среди имен.

2.4. ИНДЕКС

Индексом называется любое конечное множество колонок. Если это специально не оговаривается, то считается, что индекс – неупорядоченное множество колонок

Состав любого индекса может меняться за счет добавления или удаления колонок. Одна колонка – это также индекс, следовательно, множество колонок C^2 входит в множество индексов. Поэтому добавление и удаление колонок индекса – это операции на множестве индексов. Эти операции будут называться сложением и вычитанием индексов и обозначаться через + и –.

Как уже говорилось, любая колонка – это индекс. Интересно, что справедливо и обратное – индекс можно представить как колонку. Действительно, как и любое конечное множество колонок, индекс можно наименовать, используя некоторое чистое имя i для образа, который состоит из имен входящих в индекс колонок. Имя этой колонки-индекса можно интерпретировать как имя индекса.

$$\underbrace{\overset{\overset{p_k}{\uparrow} \quad \overset{p_l}{\uparrow} \quad \overset{p_m}{\uparrow}}{k, \ldots, l, \ldots, m}}_{\text{Индекс как безымянное множество колонок}} \quad \Rightarrow \quad \underbrace{\overset{\overset{p_k}{\uparrow} \quad \overset{p_l}{\uparrow} \quad \overset{p_m}{\uparrow}}{i \to \{k, \ldots, l, \ldots, m\}}}_{\text{Индекс как колонка – наименованное множество колонок}}$$

Из этих колонок-индексов в свою очередь могут образовываться индексы и т.д. Можно выделить *уровни индекса по построению*: L^1 – индексы из колонок, L^2 – индексы из колонок-

индексов уровня L^1, L^3 – индексы из колонок-индексов уровня L^2 и т.д. Принадлежность колонки-индекса к одному из уровней может интерпретироваться как положение индекса в иерархии индексов (рис. 2).

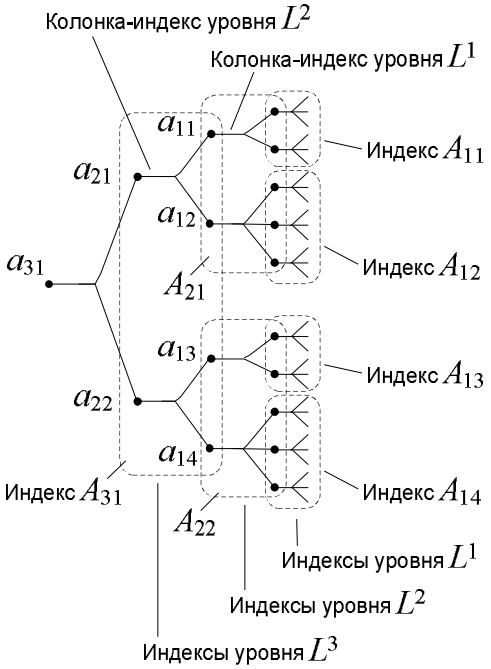

Рис. 2. Уровни индекса по построению

В качестве примера рассмотрим конечное множество $A = \{A_1, \ldots, A_n\}$, где A_k – индексы уровня L^1. Легко видеть, что A – это индекс уровня L^2. Для этого достаточно воспринимать *наши обозначения* A_k как имена соответствующих колонок-индексов. Можно явным образом наименовать индексы A_k, используя для этого чистые имена a_k, $k = 1, \ldots, n$. В результате будем иметь индекс A уровня L^2, состоящий из колонок-индексов $a_k \to A_k$, где ссылка указывает на образ $\{i_1^k, \ldots, i_{m_k}^k\}$, состоящий из имен колонок индекса A_k:

$$\begin{array}{c} A_1 \quad A_n \\ \uparrow \ \ldots \ \uparrow \\ A = \{a_1, \ldots, a_n\}, \end{array} \qquad \begin{array}{c} p_1^k, \ldots, p_{m_k}^k \\ \uparrow \ \ldots \ \uparrow \\ a_k \to A_k = a_k \to \underbrace{\{i_1^k, \ldots, i_{m_k}^k\}}_{A_k} \end{array}$$

Любой из уровней индекса является множеством колонок и, следовательно, может быть наименован, т.е. представлен в виде колонки. Образ колонки-уровня L^k состоит из имен всех колонок-индексов, образующих уровень L^k. Имя колонки может интерпретироваться как имя уровня L^k.

Вообще, какие бы причины ни приводили к формированию тех или иных множеств колонок, не будет получено ничего кроме колонок. Это одна из основных особенностей модели, основанной на колонках.

2.5. ПРЕДСТАВЛЕНИЕ ИНДЕКСА В ВИДЕ ТАБЛИЦЫ

Индекс A может быть представлен в виде таблицы, состоящей из вертикальных колонок (столбцов) переменной высоты. В нижней строке таблицы, под чертой, имена колонок. Над именем каждой колонки перечислены все имена, входящие в образ колонки. По умолчанию считается, что в таблице имена колонок и имена в образах принадлежат различным областям имен.

Для индекса в виде неупорядоченного множества колонок порядок записи колонок в таблице может быть произвольным. Если образы представляют собой неупорядоченные множества имен, то и порядок записи имен в образах колонок также может быть произвольным.

Если образы в колонках индекса представляют собой упорядоченные множества, то запись имен в образах колонок выполняется в определенном порядке, например, снизу вверх, т.е. первое имя образа в первой строке над чертой, второе – во второй и т.д.

Индекс, состоящий из первичных колонок, может быть сразу записан в виде таблицы. Для более сложных колонок могут потребоваться дополнительные преобразования.

Пусть A – представленный в виде таблицы индекс с неупорядоченными образами, не содержащий пустых колонок. Тогда для него можно построить *обратный индекс*, который будет обозначаться как A^{-1}. Для этого берется множество U_p всех имен, содержащихся в образах всех колонок индекса A. Каждое имя, принадлежащее U_p, используются как имя одной из колонок формируемого обратного индекса A^{-1}. При этом в образ колонки индекса A^{-1} с именем $k \in U_p$ включаются те и только те имена колонок индекса A, которые содержат k в своем образе. Другими словами, если имя $k \in U_p$ содержится только в образах колонок $(i_1 \mid p_1), \ldots, (i_m \mid p_m) \in A$, то колонка обратного индекса A^{-1} с именем k будет иметь вид $(k \mid \{i_1, \ldots, i_m\})$.

Например, в колонку с именем 4 приведенного ниже обратного индекса A^{-1} включены имена колонок 2 и 3 – имена всех колонок индекса A, которые содержат в образе имя 4.

A			$\Rightarrow$	A^{-1}					$\Rightarrow$	$(A^{-1})^{-1}$		
	5							3			5	
5	4									5	4	
3	3	5				2	3	2		3	3	5
1	2	4		1	2	1	2	1		1	2	4
1	2	3		1	2	3	4	5		1	2	3

Если для приведенного в примере индекса A^{-1} построить обратный индекс $(A^{-1})^{-1}$, то снова будет получен индекс A. Можно показать [2], что для представленного в виде таблицы индекса A с неупорядоченными образами, не содержащего пустых колонок, существует единственный обратный индекс A^{-1}, причем $(A^{-1})^{-1} = A$.

2.6. ВРЕМЯ, СОБЫТИЯ

С помощью образов могут быть представлены временные процессы. Для этого используются образы в виде конечных последовательностей вида $p = (i_1, \ldots, i_k)$, $i_l \in U_l$, $1 \le k \le n$, где U_k – область имен k-й координаты, n – максимальная размерность рассматриваемых конечных последовательностей. Для

того чтобы последовательность $p = (i_1, \ldots, i_n)$ можно было рассматривать как временной процесс, достаточно сопоставить порядковым номерам координат дискретные моменты времени $k \to t_k$, $k = 1, \ldots, n$.

Абстрактный образ, который может быть привязан к любому конечному отрезку дискретного времени соответствующей длины, получил название *шкалы времени*. Более точно под шкалой времени $\tau = (t_1, t_2, \ldots, t_n)$ понимается образ в виде конечной последовательности с именами из области имен моментов времени T. При этом имена, являющиеся координатами шкалы τ, образуют упорядоченную последовательность, т.е. из $k > i$ следует, что $t_k > t_i$, где $t_k, t_i \in T$. Также возможны шкалы с обратным порядком, где из $k > i$ следует, что $t_k < t_i$. Тот факт, что момент времени t_k является координатой шкалы времени τ, будет обозначаться как $t_k \in \tau$ и про него будет говориться, что это элемент или момент шкалы τ.

Может существовать несколько шкал времени. По умолчанию считается, что элементы различных шкал принадлежат различным областям имен.

Для того чтобы привязать колонку $(i \mid p)$ к моменту времени t шкалы τ, достаточно сформировать колонку $(t \mid i)$. В результате образуется цепочка ссылок $t \to i \to p$. При этом факторизация по моменту времени t, т.е. факторизация по имени t, приведет к формированию образа из имен всех колонок, относящихся к этому моменту времени:

$$\begin{array}{cccc} p_1 & p_2 & & p_m \\ \uparrow & \uparrow & \ldots & \uparrow \\ i_1 & i_2 & & i_m \\ \uparrow & \uparrow & \ldots & \uparrow \\ t & t & & t \end{array} \quad \Rightarrow \quad t \to \begin{array}{cccc} p_1 & p_2 & & p_m \\ \uparrow & \uparrow & \ldots & \uparrow \\ \{i_1, & i_2, & \ldots, & i_m\} \end{array}$$

Любая колонка вида $(t \mid p_t)$ называется *событием*, а образ p_t – *состоянием* в момент времени $t \in \tau$. На приведенной выше схеме событиями являются колонки $(t \mid i_k)$ и $(t \mid \{i_1, i_2, \ldots, i_m\})$.

2.7. УПОРЯДОЧЕННЫЙ ИНДЕКС

Наличие образов в виде конечных упорядоченных множеств, приводит к появлению упорядоченных множеств колонок, т.е. *упорядоченных индексов* или *индексов с упорядоченным множеством колонок*. Ниже показан безымянный упорядоченный индекс, состоящий из колонок $(i_k \mid p_k)$:

$$\begin{matrix} p_1\ p_2 & & p_m \\ \uparrow\ \uparrow & \dots & \uparrow \\ (i_1,\ i_2,\ \dots,\ i_m) & & \end{matrix} = \begin{pmatrix} p_1 & p_2 & & p_m \\ \uparrow & \uparrow & \dots & \uparrow \\ i_1, & i_2, & \dots, & i_m \end{pmatrix}$$

Практический интерес представляет упорядоченный индекс, колонки которого представляют собой m последних событий.

Пусть имеется некоторая шкала времени $\tau = (1, 2, \dots, n)$. Рассмотрим упорядоченный индекс A_m, число колонок которого не превосходит m, где $1 < m < n$. Первоначально индекс $A_m = \varnothing$. В момент времени 1 в индекс A_m добавляется колонка $(1 \mid s_1)$, в момент времени 2 – колонка $(2 \mid s_2)$ и т.д., где s_k – состояние в момент времени $k \in \tau$. В момент времени m индекс A_m будет состоять из колонок $(1 \mid s_1)$, $(2 \mid s_2)$, …, $(m \mid s_m)$, т.е. будет содержать m последних событий. Так как индекс A_m не может содержать более чем m колонок, то в момент времени $m + 1$ из него сначала необходимо удалить самую старую колонку $(1 \mid s_1)$, и только потом можно будет добавить новую колонку $(m + 1 \mid s_{m+1})$. Индекс A_m по-прежнему будет содержать m последних событий. Аналогично, в момент времени $m + 2$ будет удалена самая старая колонка $(2 \mid s_2)$ и добавлена новая $(m + 2 \mid s_{m+2})$ и т.д. В любой момент времени k, $m \le k < n$, индекс A_m будет содержать m колонок $(k - m + 1 \mid s_{k-m+1})$, …, $(k \mid s_k)$, представляющих собой m последних событий.

3. Системы на основе колонок, прямая и обратная задачи

В общем виде интеллектуальную систему на основе колонок можно представить в виде одного или нескольких конечных

множеств колонок, т.е. индексов, и работающего с ними механизма, который, получая информацию о внешним мире в виде образов, формирует новые колонки, изменяет уже существующие, удаляет ненужные и выполняет другие необходимые операции (рис. 3).

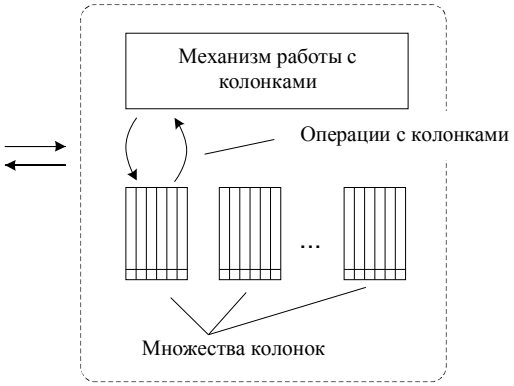

Рис. 3. Система на основе колонок

Знания в рассматриваемых системах представлены с помощью колонок, а в основе процесса накопления знаний лежит запоминание новых образов под определенными именами. Очевидно, базовыми задачами для подобных систем являются *прямая задача* – по образу получить его имя, и *обратная задача* – по имени получить соответствующий образ.

Существует общий метод решения этих задач, применимый к любым типам образов. Этим методом является метод, использующий поэлементное сравнение.

Известные системе образы хранятся в индексе A, который представляет собой множество колонок вида $(i \mid p_i)$, где p_i – образ, известный под именем i. В исходном состоянии $A = \varnothing$. Любой образ p, для которого надо решить прямую задачу поэлементно сравнивается с образами всех колонок индекса A. Если найден совпадающий образ p_i, то его имя, т.е. имя колонки $(i \mid p_i) \in A$, является именем образа p и решением прямой задачи. В противном случае выбирается некоторое чистое имя i_p из

соответствующей области имен и выполняется сложение $A + (i_p \mid p)$. Если образ p появится снова, то поэлементное сравнение даст для него имя i_p.

Еще более просто решается обратная задача. Если имеется имя i, для которого необходимо решить обратную задачу, то соответствующий образ равен образу p_i колонки $(i \mid p_i) \in A$. Если колонки с таким именем не существует, то i – чистое имя.

Очевидно, для образов тех или иных типов могут существовать более эффективные методы решения прямой и обратной задачи. Например, применение хэш-функций для образов в виде конечных последовательностей или векторов может значительно сократить объемы поэлементного сравнения. Для этих же образов более эффективным оказывается и метод пересечений.

Идея метода пересечений восходит к книжному алфавитному указателю. В нем для каждой рубрики приведено множество указателей на те страницы книги, где эту рубрику можно обнаружить. Запросу из нескольких рубрик, очевидно, соответствует пересечение множеств указателей для этих рубрик. Подобные методы с таблично заданными индексами используются в поисковых машинах интернета, в частности, в Google [3]. В работах [1, 5] метод пересечений применяется для решения задач распознавания.

Далее в разделе метод пересечений рассматривается применительно к базовым задачам для наиболее очевидных и простых типов образов – образов в виде конечных неупорядоченных множеств и образов в виде конечных последовательностей или векторов. Предлагаемый в данной работе метод характеризуется полным отсутствием необходимости в поэлементном сравнении. Кроме того, для обратной задачи предложено решение, в котором для того, чтобы обеспечить компромисс между быстродействием и объемом требуемой памяти, используется упорядоченный индекс.

3.1. МЕТОД ПЕРЕСЕЧЕНИЙ ДЛЯ ОБРАЗОВ В ВИДЕ КОНЕЧНЫХ НЕУПОРЯДОЧЕННЫХ МНОЖЕСТВ

3.1.1. ПРЯМАЯ ЗАДАЧА

Будем полагать, что образы представляют собой конечные неупорядоченные множества имен вида $p = \{i_1, \ldots, i_m\}$, $m \geq 1$, где $i_k \in U'$, U' – некоторая область имен. Очевидно, любой такой образ $p \in P$, где P – множество всех подмножеств множества U', исключая пустое.

Для решения прямой задачи будет использоваться индекс A. В исходном состоянии индекс $A = \varnothing$.

Пусть на вход системы пришел некоторый образ $p_1 \in P$. Так как система еще ничего не знает, то p_1 – новый неизвестный образ. У системы есть область имен U'', которые она может использовать для наименования неизвестных образов. Если через $U_p \subset U''$ обозначить множество имен всех известных образов, то для наименования образа p_1 система может взять любое чистое имя из $U''\backslash U_p$. Пусть выбрано имя i_1. Для запоминания образа p_1 под именем i_1 выполняется сложение $A + (p_1 \mid \{i_1\})$, т.е. к индексу A добавляется $n_1 = |p_1|$ колонок $(k \mid \{i_1\})$ для всех элементов $k \in p_1$.

Если системе опять будет предъявлен образ p_1, то пересечение n_1 образов колонок индекса A для всех имен $k \in p_1$, очевидно, даст множество, состоящее из одного имени i_1, под которым системе известен образ p_1, т.е. $\bigcap\limits_{k \in p_1} a_k = \{i_1\}$, где a_k – образ колонки $(k \mid a_k) \in A$.

Обозначим через $\eta(p)$ пересечение образов колонок $(k \mid a_k) \in A$ для всех имен $k \in p$, т.е. $\eta(p) = \bigcap\limits_{k \in p} a_k$. При этом будем полагать, что если в индексе A отсутствует колонка с именем k, то $a_k = \varnothing$.

Предположим, что теперь на вход поступил некоторый образ $p_2 \in P$. Рассмотрим для него пересечение $\eta(p_2)$. Очевидно, равенство $\eta(p_2) = \{i_1\}$ возможно только в том случае, если $p_2 \subset p_1$, $|p_2| \leq |p_1|$. Причем при $|p_2| = |p_1|$ имеет место равенство

$p_2 = p_1$. Если же $p_2 \not\subset p_1$, то всегда $\eta(p_2) = \varnothing$, так как в индексе A просто отсутствуют колонки с именами, принадлежащими множеству $p_2 \setminus p_1$.

Следовательно, для образа p_2 возможны три случая:

1. $\eta(p_2) = \{i_1\}$, $|p_2| = |p_1|$ при $p_2 = p_1$;
2. $\eta(p_2) = \{i_1\}$, $|p_2| < |p_1|$ при $p_2 \subset p_1$;
3. $\eta(p_2) = \varnothing$ при $p_2 \not\subset p_1$.

Если рассматривается вариант задачи, когда система работает с образами одинаковой мощности, возможны только случаи 1 и 3. При этом равенство $\eta(p_2) = \varnothing$ является признаком того, что образ p_2 неизвестен системе и его надо запомнить. При $\eta(p_2) = \{i_1\}$ образ p_2 известен системе под именем i_1 и $p_2 = p_1$.

Если же на вход системы поступают образы различной мощности, то равенство $\eta(p_2) = \varnothing$ также является признаком того, что образ p_2 неизвестен системе и его надо запомнить. Однако при $\eta(p_2) = \{i_1\}$ возможны случаи 1 и 2 – либо входной образ известен под именем i_1, либо входной образ неизвестен, но представляет собой подмножество образа, известного под именем i_1. Чтобы выделить случай 2, необходимо выполнить сравнение мощностей образов. Для этого в системе используется функция $n(i)$, которая определена на множестве U_p и задана с помощью множества пар (аргумент, значение). Каждый раз, когда система запоминает новый образ p, в определение функции $n(i)$ добавляется новая пара $(i, |p|)$, где i – имя, под которым запоминается образ p, $|p|$ – его мощность.

Вернемся к моменту, когда на вход системы поступил некоторый образ p_2. Если $\eta(p_2) = \{i_1\}$ и $n(i_1) = |p_2|$, то образ p_2 известен системе под именем i_1 и $p_2 = p_1$.

Если $\eta(p_2) = \varnothing$ или если $\eta(p_2) = \{i_1\}$, но $n(i_1) \neq |p_2|$, то образ p_2 является новым и его надо запомнить. Системой выбирается любое чистое имя $i_2 \in U'' \setminus U_p$, под которым теперь будет известен образ p_2, и выполняется сложение $A + (p_2 \mid \{i_2\})$. В определение функции $n(i)$ добавляется новая пара $(i_2, |p_2|)$, а решением прямой задачи для образа p_2 будет имя i_2.

Очевидно, при выполнении сложения $A + (p_2 \mid \{i_2\})$ факторизация по имени приведет к тому, что любая колонка индекса A с именем $k \in p_1 \bigcap p_2$ будет иметь вид $(k \mid \{i_1, i_2\})$. Если рассмотреть все возможные образы индекса A, то факторизация по образу показывает, что полученный в результате сложения индекс A состоит из трех колонок $(p_1 \setminus p_2 \mid \{i_1\})$, $(p_1 \bigcap p_2 \mid \{i_1, i_2\})$ и $(p_2 \setminus p_1 \mid \{i_2\})$, где $p_1 \setminus p_2$, $p_1 \bigcap p_2$ и $p_2 \setminus p_1$ – классы эквивалентности.

Аналогично тому, как это делалось для образа p_2, будут запоминаться и другие новые образы p_3, p_4, p_5 и т.д. Запоминание l новых образов можно представить в виде цепочки сложений:

$$A = \sum_{k=1}^{l} (p_k \mid \{i_k\}),\ n(i) = \bigcup_{k=1}^{l} (i_k, \mid p_k \mid),$$

где i_k – имя образа p_k.

Запоминание любого образа p будет выполняться только при условии, что $\eta(p) = \varnothing$ или $\eta(p) \neq \varnothing$ и $n(i) \neq |p|$ для $\forall i \in \eta(p)$. Покажем, что в этом случае при появлении любого образа $p \in P$ всегда можно однозначно сказать, известен ли этот образ, и определить его имя.

Рассмотрим любой образ $p \in P$. Пусть $\eta(p) = \varnothing$. Это означает, что образ p не предъявлялся системе, так как в противном случае по построению существовали бы колонки для $\forall k \in p$ и пересечение $\eta(p)$ содержало бы имя, под которым известен образ p.

Пусть теперь $\eta(p) \neq \varnothing$ и $n(i) \neq |p|$ для $\forall i \in \eta(p)$. Это также означает, что образ p не предъявлялся системе, так как в противном случае по построению пересечение $\eta(p)$ содержало бы имя i, под которым известен образ p, и выполнялось бы равенство $n(i) = | p |$.

Наконец, пусть $\eta(p) \neq \varnothing$ и существует по крайней мере одно имя $i \in \eta(p)$, для которого $n(i) = | p |$. Это означает, что образ p известен системе под именем i. Покажем, что такое имя является единственным.

Предположим, что $\exists i, i' \in \eta(p)$, $i \neq i'$ и $n(i) = n(i') = |p|$. Так как образ p не может одновременно получить сразу два имени,

то это означает, что он сначала получил одно имя, а потом в одном из повторных предъявлений – второе. Это невозможно по построению, т.е. $i = i'$, и если существует такое имя $i \in \eta(p)$, что $n(i) = |p|$, то это имя является единственным.

Таким образом, рассмотренный метод дает решение прямой задачи для любого образа $p \in P$. Схему его работы можно описать следующим образом.

В исходном состоянии $A = \varnothing$ и $n(i) = \varnothing$.

1. Для предъявленного образа $p \in P$ вычисляется пересечение $\eta(p)$.

2. Если $\eta(p) \neq \varnothing$ и $\exists i \in \eta(p)$ такое, что $n(i) = |p|$, то имя i единственное, представляет собой имя, под которым известен образ p, и является решением прямой задачи.

3. В противном случае, если $\eta(p) = \varnothing$ или $\eta(p) \neq \varnothing$, но $n(i) \neq |p|$ для $\forall i$, образ p является новым. Для него выбирается любое чистое имя $i \in U'' \backslash U_p$, выполняется сложение $A + (p \mid \{i\})$, а в определение функции $n(i)$ добавляется пара $(i, |p|)$. Имя i, под которым теперь будет известен образ p, является решением прямой задачи.

Здесь необходимо сделать одно важное замечание. Выбор для нового образа чистого имени из множества $U'' \backslash U_p$ обеспечивает выполнение условия, которое для правильной работы приведенного метода является необходимым. Это условие состоит в том, что каждый образ получает единственное уникальное имя, т.е. разные образы не могут иметь одно и то же имя.

Легко показать, что при нарушении этого условия рассмотренный метод не сможет различать образы.

Действительно, пусть два разных образа p_1 и p_2 имеют одно и тоже имя i и $|p_1| = |p_2| = n$. Рассмотрим любой образ p_3, $|p_3| = n$, который имеет часть элементов из образа p_1, а оставшуюся часть из образа p_2. Очевидно, образ p_3 не равен образам p_1 и p_2, но при этом $\eta(p_3) = \eta(p_1) = \eta(p_2) = i$ и $|p_3| = |p_1| = |p_2| = n(i)$, т.е данный метод не сможет отличить образ p_3 от образов p_1 и p_2.

3.1.2. ОБРАТНАЯ ЗАДАЧА

Обратная задача состоит в том, чтобы по имени образа i получить сам образ p_i. Она будет решаться обычным образом. Для этого кроме индекса A будет использоваться дополнительный индекс B, в который при запоминании образа p под именем i будет добавляться колонка $(i \mid p)$.

В начале работы $A = \varnothing$, $B = \varnothing$ и $n(i) = \varnothing$.

Если образ p является новым, то для него выбирается некоторое чистое имя $i \in U'' \backslash U_p$ и выполняются сложения $A + (p \mid \{i\})$, $B + (i \mid p)$. В определение функции $n(i)$ добавляется пара $(i, |p|)$. Таким образом, процесс запоминания l новых образов можно представить цепочкой сложений:

$$A = \sum_{k=1}^{l} (p_k \mid \{i_k\}),\ B = \sum_{k=1}^{l} (i_k \mid p_k),\ n(i) = \bigcup_{k=1}^{l} (i_k, \mid p_k \mid).$$

Пусть решается обратная задача и предъявлено некоторое имя $i \in U''$. Если имя i представляет собой имя известного образа, т.е. $i \in U_p$, то решением обратной задачи является образ p_i колонки $(i \mid p_i) \in B$. В противном случае имя i является чистым.

Нетрудно заметить, что $B = A^{-1}$. Для доказательства достаточно использовать определение обратного индекса по отношению к колонкам $(p_k \mid \{i_k\})$ и $(i_k \mid p_k)$.

ПРИМЕР. Пусть индексы A, B и функция $n(i)$ имеют вид:

A

3	3	3	
1	2	1	2
1	2	3	4

$n(i)$

i	1	2	3
n	2	2	3

B

		3
3	4	2
1	2	1
1	2	3

Легко видеть, что системе известны три образа: образ $\{1, 3\}$ под именем 1, образ $\{2, 4\}$ под именем 2 и образ $\{1, 2, 3\}$ под именем 3.

Пусть на входе появился образ $p = \{1, 2, 3\}$. Пересечение $\eta(p) = \{3\}$ и $n(3) = |p| = 3$. Следовательно, образ p известен системе под именем 3.

Пусть теперь пришел образ $p = \{2, 3\}$. Пересечение $\eta(p) = \{3\}$, но $n(3) = 3 \neq |p|$. Следовательно, образ p является новым. Для него выбирается имя $4 \in U'' \backslash U_p$. Выполняется сложение $A + (p \mid \{4\})$ и $B + (4 \mid p)$, а в определение функции $n(i)$ добавляется пара (4, 2). В результате получим:

A

	4	4	
3	3	3	
1	2	1	2
1	2	3	4

$n(i)$

i	1	2	3	4
n	2	2	3	2

B

		3	
3	4	2	3
1	2	1	2
1	2	3	4

Если на входе опять появится образ $p = \{2, 3\}$, то $\eta(p) = \{3, 4\}$, причем $n(4) = |p| = 2$, что означает, что образ p известен системе под именем 4.

Пусть теперь решается обратная задача и предъявлено имя $i = 3$. Так как $i \in U_p$, то образ p_i равен образу колонки 3 индекса B, т.е. $p_i = \{1, 2, 3\}$. ■

Приведенное выше решение обратной задачи имеет один недостаток – в индексе B должны сохраняться все известные образы. В результате значительно возрастает объем требуемой памяти. Очевидным выходом является сохранение в индексе B лишь части известных образов, например, тех, которые встречаются на отрезке из m последних моментов времени. При этом индекс B будет представлять собой упорядоченный индекс, содержащий m последних событий, каждое из которых связано с появлением некоторого образа на входе системы (см. раздел 2.7). Для всех остальных образов может использоваться более медленная схема восстановления образа путем просмотра колонок индекса A [2].

3.2. МЕТОД ПЕРЕСЕЧЕНИЙ ДЛЯ ОБРАЗОВ В ВИДЕ КОНЕЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ИЛИ ВЕКТОРОВ

Рассмотрим теперь прямую и обратную задачи для образов в виде конечных последовательностей или векторов. Будем полагать, что в этом случае, образы $p = (i_1, i_2, \ldots, i_m)$ принадлежат множеству $P = \bigcup_{m=1}^{n} P^m$, где $P^m = U_1 \times \ldots \times U_m$, U_k – область имен k-й координаты. Длина или размерность последовательности p будет обозначаться через $|p|$.

При решении прямой задачи будет использоваться индекс $A = \{A_1, A_2, \ldots, A_n\}$, который представляет собой индекс уровня L^2 (см. раздел 2.4), где A_k – индекс для k-й координаты.

Кроме индекса A также будет использоваться функция $m(i)$, которая будет содержать длину известных последовательностей. В исходном состоянии $A_k = \varnothing$ ($k = 1, \ldots, n$), $m(i) = \varnothing$. Запоминание нового образа $p = (i_1, i_2, \ldots, i_m)$ под некоторым именем i выполняется как покоординатное сложение $A_k + (i_k \mid \{i\})$, $k = 1, \ldots, m$, где i_k – имя, являющимся k-й координатой входного образа p:

$$A + (p \mid \{i\}) = \{A_1 + (i_1 \mid \{i\}),\ \ldots,\ A_m + (i_m \mid \{i\})\}.$$

Кроме этого, в определение функции $m(i)$ добавляется пара $(i, \mid p \mid)$. Таким образом, запоминание системой l новых образов можно представить в виде цепочки сложений:

$$A = \sum_{k=1}^{l} (p_k \mid \{i_{p_k}\}),\ m(i) = \bigcup_{k=1}^{l} (i_{p_k}, \mid p_k \mid),$$

где i_{p_k} – имя образа p_k.

Обозначим через $\eta(p)$ покоординатное пересечение $\eta(p) = \bigcap_{k=1}^{m} a_{i_k}$, где a_{i_k} – образ колонки $(i_k \mid a_{i_k}) \in A_k$, i_k – имя, являющееся k-й координатой входного образа p.

Пусть на вход системы пришел некоторый образ $p = (i_1, i_2, \ldots, i_m)$.

Очевидно, если образ p неизвестен и не является подпоследовательностью известных образов, то $\eta(p) = \varnothing$. Если $\eta(p) \neq \varnothing$

и $|p| \neq m(i)$ для $\forall i \in \eta(p)$, то образ p также неизвестен, но его координаты совпадают с частью координат известных образов, имена которых принадлежат множеству $\eta(p)$. Наконец, если $\eta(p) \neq \varnothing$ и существует имя $i \in \eta(p)$, такое, что $|p| = m(i)$, то образ p известен под именем i. Причем в последнем случае такое имя единственное, так как повторное запоминание известного образа под другим именем невозможно.

Схема работы метода имеет вид.

В исходном состоянии $A_k = \varnothing$ ($k = 1, \ldots, n$), $m(i) = \varnothing$.

1. Для появившегося образа $p \in P$ вычисляется покоординатное пересечение $\eta(p)$.

2. Если $\eta(p) \neq \varnothing$ и $\exists i \in \eta(p)$ такое, что $n(i) = |p|$, то имя i единственное, представляет собой имя образа p и является решением прямой задачи.

3. В противном случае, если $\eta(p) = \varnothing$ или $\eta(p) \neq \varnothing$, но $n(i) \neq |p|$ для $\forall i \in \eta(p)$, образ p является новым. Для него выбирается любое чистое имя $i \in U'' \backslash U_p$, выполняется сложение $A + (p \mid \{i\})$, а в определение функции $m(i)$ добавляется пара $(i, |p|)$. Имя i, под которым теперь будет известен образ p, является решением прямой задачи.

Как и ранее необходимым условием для правильной работы метода является то, что каждый образ получает единственное уникальное имя, что обеспечивается выбором чистого имени из множества $U'' \backslash U_p$. При нарушении этого условия метод не сможет различать образы.

Действительно, пусть образы p_1 и p_2, имеющие одно и то же имя i, отличаются всеми координатами и $|p_1| = |p_2|$. Тогда, взяв образ той же размерности, у которого одна часть координат равна координатам образа p_1, а вторая часть – координатам образа p_2, получим образ p_3, отличный от p_1 и p_2. При этом, очевидно, $\eta(p_3) = \eta(p_1) = \eta(p_2) = i$ и $|p_3| = |p_1| = |p_2| = m(i)$, т.е метод не сможет отличить образ p_3 от образов p_1 и p_2.

Обратная задача решается аналогично тому, как это делалось для образов в виде неупорядоченных множеств. При этом используется дополнительный индекс B, с которым при запоми-

нании образа p складывается колонка $(i \mid p)$. Запоминание l новых образов можно представить цепочкой сложений:

$$A = \sum_{k=1}^{l} (p_k \mid \{i_{p_k}\}),\ B = \sum_{k=1}^{l} (i_{p_k} \mid p_k),\ m(i) = \bigcup_{k=1}^{l} (i_{p_k}, \mid p_k \mid),$$

где i_{p_k} – имя образа p_k.

ПРИМЕР. Пусть для $n = 3$ имеются следующие индексы A, B и функция $m(i)$:

A_1			A_2			A_3			B			
									3	2	2	
4		2	2				4		3	1	1	2
1		3	3	4	1		3	2	1	3	3	1
1	2	3	1	2	3	1	2	3	1	2	3	4

$m(i)$

i	1	2	3	4
m	2	3	3	3

Легко видеть, что в индексе $A = \{A_1, A_2, A_3\}$ под именем 1 запомнен образ (1, 3), под именем 2 – образ (3, 1, 3), под именем 3 – образ (3, 1, 2), а под именем 4 – образ (1, 2, 2).

Пусть на вход пришел образ $p = (3, 1, 2)$. Пересечение $\eta(p)$, равное пересечению образа колонки индекса A_1 с именем 3, образа колонки индекса A_2 с именем 1 и образа колонки индекса A_3 с именем 2, будет состоять из одного имени 3, при этом $m(3) = 3$. Это означает, что на входе образ, известный под именем 3.

Пусть теперь на входе появился образ $p = (3, 1)$. Пересечение $\eta(p) = \{2, 3\}$, но $m(2) = m(3) \neq 2$. Следовательно, p – это неизвестный новый образ. Из множества $U'' \backslash U_p$ для него выбирается имя 5. Затем выполняются сложения $A + ((3, 1) \mid 5)$ и $B + (5 \mid (3, 1))$, а в определение функции $m(i)$ добавляется пара (5, 2). В результате получим:

A_1			A_2			A_3			B				
		5	5							3	2	2	
4		2	2			4			3	1	1	2	1
1		3	3	4	1		3	2	1	3	3	1	3
1	2	3	1	2	3	1	2	3	1	2	3	4	5

$m(i)$

i	1	2	3	4	5
m	2	3	3	3	2

Если на входе снова появится образ $p = (3, 1)$, то пересечение $\eta(p) = \{2, 3, 5\}$, причем $m(5) = 2$. Следовательно, на входе образ, известный под именем 5.

Пусть теперь решается обратная задача и на вход пришло имя $i = 3$. Так как $i \in U_p$, то образ p_i равен образу колонки 3 индекса B, т.е. $p_i = (3, 1, 2)$.

3.3. ВЫЧИСЛИТЕЛЬНАЯ ЭФФЕКТИВНОСТЬ МЕТОДА ПЕРЕСЕЧЕНИЙ

Пусть решается прямая задача для образов в виде конечных последовательностей или векторов одинаковой размерности. Пусть также l – это число запомненных образов, n – размерность вектора, m – число возможных значений каждой координаты.

Рассмотрим число операций, которые необходимо выполнить при решении прямой задачи методом поэлементного сравнения и с помощью рассмотренного выше метода пересечений.

При поэлементном сравнении известные системе образы хранятся в индексе A, который представляет собой множество колонок вида $(i \mid p_i)$, где p_i – образ, известный под именем i. Если на входе появился некоторый вектор $p = (i_1, i_2, \ldots, i_n)$, то для решения прямой задачи выполняется его покоординатное сравнение с образами всех колонок индекса A. Если найден совпадающий образ p_i, то его имя, т.е. имя колонки $(i \mid p_i) \in A$, является именем образа p и решением прямой задачи. Максимальное

число операций, которые при этом необходимо выполнить, равно $O_M = nl$.

Оценим теперь число операций, необходимое для решения прямой задачи при помощи метода пересечений с использованием индекса $A = \{A_1, A_2, \ldots, A_n\}$. Каждая из m колонок индекса A_k ($k = 1, \ldots, n$) содержит в среднем l / m имен. При вычислении пересечения $\eta(p)$ необходимо выполнить $O_A = nl / m$ операций, т.е. в m раз меньше чем при покоординатном сравнении. Другими словами, *чем больше имен использует система для представления значений координат, т.е. чем точнее она отражает реальные объекты, тем она быстрее работает*[1]. При большой величине m выигрыш по сравнению с покоординатным сравнением может быть довольно значительным. Например, пусть запомнено $l = 10^5$ образов размерности $n = 10^2$, каждая координата которых может иметь $m = 10^3$ значений. Тогда при покоординатном сравнении необходимо выполнить 10 млн. операций, а при вычислении пересечения $\eta(p)$ только 10 тыс. операций. Кроме этого, следует отметить, что метод пересечений хорошо распараллеливается.

Оценки памяти, необходимой для этих двух методов, совпадают. И в том, и в другом случае в памяти должно храниться nl целых чисел.

4. Классы задач, для решения которых могут использоваться системы на основе колонок

Базовые задачи служат основой для решения других задач. В данном разделе это сначала будет показано на примере задачи классификации, а затем будет доказана возможность реализации в рассматриваемых системах произвольных булевых функций. Эти результаты позволят оценить те клас-

[1] *Аналогичное утверждение справедливо и для образов в виде неупорядоченных множеств. В этом случае m – число различных имен, участвующих в формировании образов.*

сы задач, для которых могут применяться интеллектуальные системы на основе колонок.

4.1. ЗАДАЧА КЛАССИФИКАЦИИ

Рассмотрим задачу классификации, в которой любой объект из некоторой проблемной области описывается с помощью образа $p \in P$ и принадлежит одному или нескольким классам объектов. Задача состоит в том, чтобы по входному образу $p \in P$ определить, к какому классу или классам он принадлежит.

Одно из основных отличий данной задачи от рассмотренных выше связано с наименованием неизвестных образов. Если ранее система присваивала новому образу любое имя из имеющихся у нее чистых имен, то теперь она должна связать этот образ с совершенно определенным множеством имен классов.

Будем полагать, что для каждого нового образа система имеет возможность получить множество имен классов, к которым этот образ принадлежит. В этом случае для решения задачи классификации можно применить любой из методов решения прямой задачи, используя дополнительный индекс B для запоминания имен классов. Действительно, если входной образ p является новым, то достаточно его запомнить под именем i_p – прямая задача, а в индекс B добавить колонку $(i_p \mid q_p)$, где $q_p = \{c_1, ..., c_{n_p}\}$ – множество имен классов, к которым принадлежит образ p, $c_k \in U_q$, U_q – область имен классов. Тогда при предъявлении образа p сначала определяется его имя i_p, а затем по нему колонка $(i_p \mid q_p)$ индекса B. Образ этой колонки q_p и есть множество имен классов, к которым принадлежит входной образ p.

4.2. БУЛЕВЫ ФУНКЦИИ

Рассмотрим произвольную булеву функцию $f\colon B^n \to B$, $B = \{0, 1\}$. Она определяет разбиение B^n на классы эквивалентности:

$$B_0^n = \{x \in B^n \mid f(x) = 0\},$$

$$B_1^n = \{x \in B^n \mid f(x) = 1\},$$

где $B_0^n \bigcup B_1^n = B^n$, $B_0^n \bigcap B_1^n = \varnothing$.

Очевидно, для того чтобы реализовать функцию f с помощью системы на основе колонок, достаточно решить задачу классификации для двух классов B_0^n и B_1^n. Положим для наглядности, что классы B_0^n и B_1^n получили имена 0 и 1 из некоторой области имен U_q. Тогда для любого вектора $p \in B^n$ будем иметь: если p принадлежит классу с именем 0, то функция $f(p) = 0$; если же p принадлежит классу с именем 1, то $f(p) = 1$.

Так как мы имеем дело с разбиением множества B^n всего на два класса, то задачу можно значительно упростить, оставив только один класс, например, B_1^n. Тогда для $\forall p \in B^n$, если p принадлежит классу 1, то $f(p) = 1$, в противном случае $f(p) = 0$.

Данная задача классификации может быть решена так, как это описывалось в предыдущем разделе. При этом может использоваться любой подходящий метод решения прямой задачи. Таким образом, любая булева функция $f: B^n \to B$, $B = \{0, 1\}$ может быть реализована в системе на основе колонок.

Покажем более подробно, как это будет выглядеть при использовании метода пересечений.

Итак, пусть образы представляют собой вектора $p \in P$, $P = U_B^n$, где $U_B = \{0, 1\}$ – область имен для любой координаты.

Так как все вектора имеют одну и ту же размерность, а класс всего один, то функцию $m(i)$ и индекс B можно опустить.

Сначала при помощи индекса $A = \{A_1, \ldots, A_n\}$ запоминаются все вектора класса B_1^n:

$$A = \sum_{k=1}^{l} (p_k \mid \{i_k\}),$$

где $p_k \in B_1^n$, i_k – имя вектора p_k, $l = \mid B_1^n \mid$.

В результате системе известны только вектора класса B_1^n. Других векторов она не знает.

Далее система работает только в режиме классификации. Пусть $p \in P$ – произвольный входной вектор. Если $\eta(p) \neq \varnothing$, то

вектор p известен системе, $p \in B_1^n$ и $f(p) = 1$. В противном случае $f(p) = 0$. ■

Этот результат показывает возможность решения с помощью рассматриваемых систем тех задач, для которых может применяться логический подход. Это относится и к логическому выводу, используемому, в частности, в синтетических задачах. Вместе с результатами раздела 4.1 это позволяет говорить о принципиальной возможности применения интеллектуальных систем на основе колонок для решения различных задач классификационного и синтетического типа.

Литература

1. МИХАЙЛОВ А.М. *Распознавание образов с помощью их индексирования* // Автоматика и телемеханика. – 2012. – №4. – С. 151–161.
2. ЧЕСНОКОВ А.М. *Введение в общую теорию колонок*. – М.: ИПУ РАН. – 2012. – 141 с.
3. BRIN S., PAGE L. *The Anatomy of a Large-Scale Hypertextual Web Search Engine* // Proc. Seventh International Conference on World Wide Web (WWW7). – Amsterdam: Elsevier, 1998. – P. 107–117.
4. HAWKINS J. *On Intelligence*. – New York: Henry Holt and Company, 2005. – 262 p.
5. MIKHAILOV A. *Biologically Inspired Artificial Neural Cortex and its Formalism* // World Academy of Science, Engineering and Technology. – August 2009. – Vol. 56. – P. 121.

COLUMNS-BASED INTELLIGENT SYSTEMS

Alexander Chesnokov, Institute of Control Sciences of RAS, Moscow, Cand. Sc. (alex-ches@yandex.ru).

Abstract: *The paper considers columns-based intelligent systems and introduces basic definitions and notions. The direct and inverse problems of patterns are introduced. Solutions to these problems and their computational complexity are discussed. Also, a solution to pattern classification problem and a method of Boolean functions implementation are introduced. Finally, a class of problems that can be solved with columns-based intelligent systems is identified.*

Keywords: artificial intelligence, columns-based intelligent systems, column.

Статья представлена к публикации
членом редакционной коллегии О.П. Кузнецовым

Поступила в редакцию 13.03.2013.
Опубликована 30.11.2013.