ЛОГИКА РАСПОЗНАВАНИЯ А. Д. ЗАКРЕВСКОГО НА ПРИМЕРЕ ЗАДАЧИ ВОССТАНОВЛЕНИЯ ПРОПУСКОВ В ИСТОРИКО–АРХЕОЛОГИЧЕСКИХ ИССЛЕДОВАНИЯХ

A. D. ZAKREVSKIJ LOGIC OF RECOGNITION FOR FILLING DATA GAPS IN HISTORICAL-ARCHAEOLOGICAL INVESTIGATIONS

Шпирко Сергей Валерьевич
Кандидат физико-математических наук, старший научный сотрудник факультета управления и прикладной математики Московского физико-технического института.
E-mail:shpirko@yahoo.com

Sergey V. Shpirko

Рассматривается задача восстановления пропусков в историко-археологических исследованиях. Традиционным подходом к ее решению является применение различных математико-статистических методов. Для решения данной задачи предлагается использовать логико-комбинаторный подход, основанный на логике распознавания А. Д. Закревского. Описывается суть данного подхода, разрабатывается конкретный алгоритм. Работа алгоритма демонстрируется на примере задачи восстановления деталей рисунков граффити Хазарии.

Ключевые слова: бинарная матрица, троичный вектор, запрещенная комбинация, максимальный интервал, кратчайшее строчное покрытие, граффити Хазарии.

In this paper we consider filling data gaps in historical-archaeological investigations. A traditional way for the solution is to apply mathematical-statistical techniques. In this paper we present a logical-combinatorial method, based upon A. D. Zakrevskij logic of recognition. We show how to apply above method for filling gaps in drawings of graffiti of Khazaria.

Keywords: boolean matrix, ternary vector, forbidden combination, maximal interval, shortest row cover, graffiti of Khazaria.

1. ПОСТАНОВКА ЗАДАЧИ

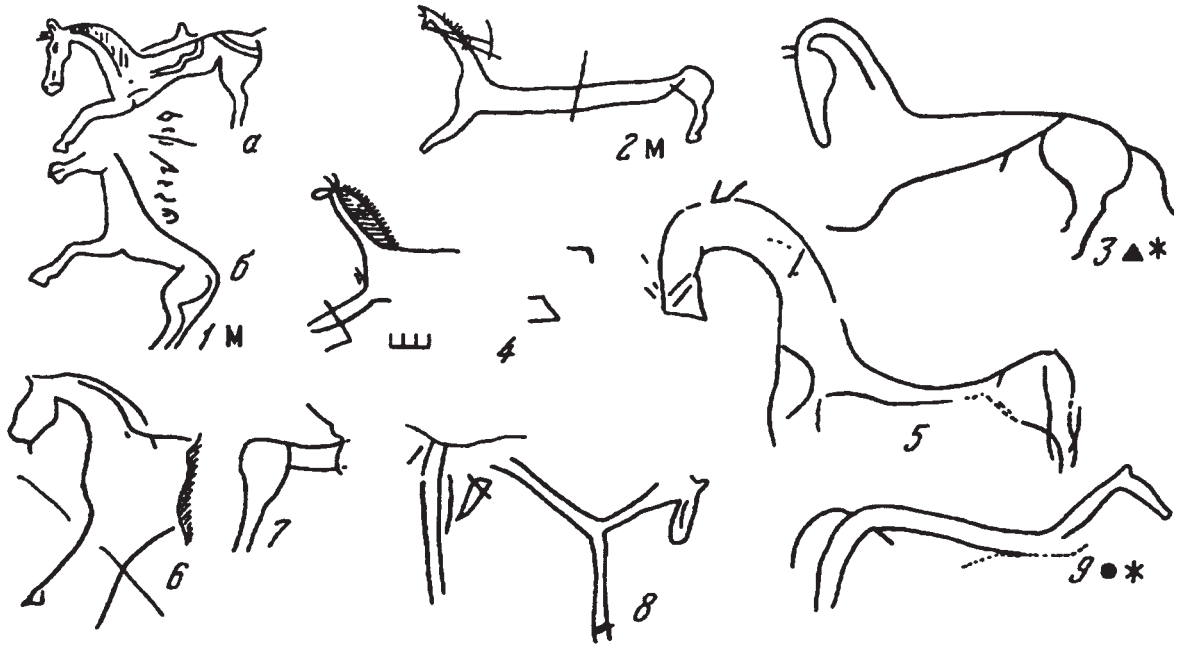
В археологии и изучении памятников изобразительного искусства прошлого весьма часто возникает проблема восполнения данных. В частности, на рисунках, оставленных на камне, кости, дереве, могут быть утрачены детали, порой весьма существенные для последующих задач классификации данных изображений, реконструкции сюжетов.

Первым шагом к решению данной задачи является выявление устойчиво повторяющихся и мало зависящих от содержания образа фрагментов изображений. Таким образом, излагается перечень признаков, по которым рисунки будут сравниваться между собой[1]. В частности, если рисунок характеризуется набором качественных признаков, то его можно представить в виде бинарного вектора. Каждая компонента данного вектора принимает значение либо единицы (рисунок обладает соответствующим признаком), либо нуля (признак отсутствует).

Естественно считать, что чем больше совпадающих компонент у векторов, тем ближе по стилю, по происхождению соответствующие им рисунки.

Предположим теперь, что у нас есть тестовый объект, у которого значение одного из признаков (будем называть его целевым) неизвестно. В содержательном смысле наша задача состоит в том, чтобы по имеющемуся набору объектов с известными значениями признаков (обучающей выборке) попытаться восстановить целевой признак у тестового объекта. Будем считать, что выборка включает сравнительно небольшое число объектов. Этот факт делает затруднительным для решения подобной задачи применение различных математико-статистических методов (ввиду отсутствия массовости данных).

В данной работе для решения подобной задачи предлагается формализованный подход, основанный на применении логики распознавания А. Д. Закревского[2]. Суть данной логики состоит в том, что любую систему можно охарактеризовать множеством запретов. В нашем случае это совокупность таких комбинаций признаков, которыми не могут обладать одновременно ни один из наблюдаемых объектов. Так, анализируя граффити Хазарии (Маяцкое городище), В. Е. Флерова выделяет в отдельный вариант (см. рис. –3, 8, 9) те рисунки, в которых представлены животные с хвостом в виде линии, подробно переданными мордами, подчеркнутыми признаками пола, но без гривы[3]. То есть вышеуказанные четыре признака входят в запрещенную комбинацию.



Рисунки на камнях Маяцкого городища

Понятно, что чем меньше признаков входят в запрещенные комбинации, тем сильнее связь между ними. В данной статье мы ограничились исследованием запрещенных комбинаций лишь двух признаков. Как будет показано ниже, такое предположение является разумным.

2. ПОСТРОЕНИЕ МАТРИЦЫ ЗАПРЕТОВ

Взаимное сочетание признаков у объектов выборки будем выражать с помощью понятия троичного вектора. Смысл его удобно изложить на конкретном примере.

Пример 1. Зафиксируем произвольную пару признаков i и j (будем считать, что объекты выборки имеют n признаков). Будем исследовать взаимосочетаемость этой пары на всей выборке. Предположим, что в результате было установлено, что во всех объектах выборки либо отсутствуют оба эти признака, либо есть признак j и нет признака i. Таким образом, выявлены следующие две запрещенные комбинации пары признаков. Первая запрещенная комбинация — когда оба признака присутствуют у объектов. Вторая — когда признак i присутствует, а признак j отсутствует. Запрещенную комбинацию будем представлять в виде следующего n-мерного вектора. Каждая компонента этого вектора отвечает за присутствие/отсутствие соответствующего признака в запрещенной комбинации. Если исследуемый признак присутствует, то соответствующая компонента равна единице.

Если отсутствует, то соответствующая компонента равна нулю. Все остальные компоненты, не участвующие в рассмотрении, полагаются равными '-'. Построенные таким образом векторы будем называть *троичными*. В нашем примере их будет два:

$$a_1, a_2 \in R^n : a_1[i] = a_1[j] = 1;$$
$$a_2[i] = 1, a_2[j] = 0, a_1[l] = \quad .$$
$$= a_2[l] = '-' \quad \forall l \neq i, l \neq j$$

Каждый троичный вектор задает в бинарном пространстве признаков определенный интервал. Число знаков троичного вектора, отличных от '-', называют рангом соответствующего интервала. В нашем случае ранг $k = 2$. В данный интервал входят всевозможные бинарные векторы, у которых значения двух соответствующих компонент совпадают с заданными.

Допустим, мы сформировали один такой интервал. Будем тестировать данный интервал на выборке из m объектов. А именно, подсчитаем вероятность того, что ни один из объектов выборки не попадет в данный интервал.

Сначала зафиксируем произвольные значения у первых k компонент интервала, например, положим их все равными нулю. Начнем с одного из объектов выборки (бинарного вектора). Чтобы не попасть в заданный интервал, объекту надо иметь хотя бы одно единичное значение в первых k компонентах. Соответствующая вероятность непопадания равна $1 - p$, где p есть вероятность иметь все нули в первых k компонентах. Число всех бинарных комбинаций из n компонент равно 2^n. Тогда нетрудно подсчитать, что

$$p = \frac{2^{n-k}}{2^n} = 2^{-k}.$$

Рассмотрим теперь два объекта из выборки и подсчитаем вероятность того, что ни один из них не попадет в наш интервал с первыми k нулевыми компонентами. Это эквивалентно тому, что и первый, и второй векторы будут иметь хотя бы одну единицу в первых k компонентах. Соответствующая вероятность непопадания равна

$$(1-p)(1-p) = (1-2^{-k})^2.$$

Продолжая аналогичные рассуждения далее, получим вероятность непопадания в данный интервал для всех m объектов выборки:

$$\underbrace{(1-p)(1-p)\times..\times(1-p)}_{m\text{ раз}} = (1-2^{-k})^m.$$

Заметим, что мы фиксировали k нулевых значений. Число бинарных комбинаций произвольных k компонент равно 2^k. Далее, эти k значений мы располагали в фиксированных (первых) компонентах. Число всевозможных способов выбора k компонент равно C_n^k, где

$$C_n^k = \frac{n!}{k!(n-k)!}.$$

Итак, вероятность того, что выборка из m объектов не попадет в интервал ранга k равна

$$C_n^k 2^k (1-2^{-k})^m.$$

Чем меньше данная вероятность, тем труднее найти такой интервал. А если он все же найден, тем выше его ценность.

Зададим некоторое пороговое значение w. Например, положим $w = 0{,}01$. Зафиксируем число признаков и объем выборки n и m соответственно. Выясним, для каких рангов k данная вероятность непопадания будет меньше w. Результаты зависимости ранга от числа признаков и объема выборки приведены в таблице 1[4].

Таблица 1

Зависимость ранга интервала от числа признаков и объема выборки

n	m					
	20	50	100	200	500	1000
10	1	2	3	4	5	6
30	1	2	2	3	4	5
100	1	1	2	3	4	5

Из данной таблицы следует, что при $n = 10$ и $50 < m < 100$ разумно рассматривать интервалы ранга не больше двух. Действительно, если взять интервал с большим рангом, то слишком велика будет вероятность непопадания в него при очередном тестировании. Поэтому такой интервал не несет никакой существенной информации о системе и взаимосочетаемости признаков. Его ценность близка к нулю.

Итак, образуем из совокупности всех интервалов запретов второго ранга матрицу $U \subset R^{m \times n}$. В данной матрице число строк m равно числу интервалов запретов, а число столбцов n равно числу признаков. Чем большем строк в матрице U, тем больше область запрета. И тем больше шансов однозначно восстановить значение целевого признака.

3. НАХОЖДЕНИЕ КРАТЧАЙШЕЙ ФОРМЫ МАТРИЦЫ ЗАПРЕТОВ

На практике размер матрицы запретов может быть очень большим. Поэтому задача построения кратчайшей формы матрицы, т. е. матрицы с минимальным количеством строк,

представляется весьма важной. Разобьем процесс построения такой матрицы на три этапа.

3.1. Построение расширенной матрицы

Обозначим через $M(U)$ множество всех бинарных векторов, входящих в интервалы запрета матрицы U. Пусть в результате некоторых преобразований исходной матрицы U получена новая троичная матрица R. Назовем данные преобразования *эквивалентными*, если $M(R) = M(U)$. Опишем два таких преобразования.

Определение 2. Два троичных вектора называются *смежными*, если найдется одна и только одна компонента, значение которой в одном из векторов равно нулю, а в другом векторе равно единице.

Пример 2. Рассмотрим следующие два вектора a_1, a_2:

$$a_1[i] = a_1[j] = 1, a_1[k] = '-' \quad \forall k \neq i, k \neq j, \qquad (1)$$

$$a_2[j] = 0, a_2[l] = 1, a_2[k] = '-' \quad \forall k \neq j, k \neq l. \qquad (2)$$

Нетрудно видеть, что данные два вектора смежны по компоненте j. Из формулы (1) следует, что комбинация признаков i и j недопустима. Следовательно, если объект обладает признаком i, то он не обладет признаком j. А из формулы (2) следует, что данный объект не может обладать и признаком l. В противном случае он попал бы в интервал запрета, задаваемый вторым вектором.

Таким образом, приходим еще к одному интервалу запрета, задаваемому вектором a_3:

$$a_3[i] = 1, a_3[l] = 1, a_3[k] = '-' \quad \forall k \neq i, k \neq l.$$

Данную операцию называют *обобщенным склеиванием* двух смежных векторов по компоненте. Получающийся в результате данной операции вектор содержит в «склеенной» компоненте (в приведенном примере это компонента j) значение '–'. А значение любой другой компоненты определяется по следующему правилу. Если значения компоненты в обоих смежных векторах равны '–', то и в новом векторе данная компонента равна '–'. Иначе компонента нового вектора принимает то значение соответствующей компоненты смежного вектора, которое отлично от '–'.

Опишем теперь вторую эквивалентную операцию.

Определение 3. Будем говорить, что один вектор *поглощает* другой, если все компоненты первого вектора, значения которых отличны от '–', совпадают с соответствующими компонентами второго вектора.

Пример 3. Рассмотрим два вектора a_1, a_2:

$$a_1[i] = 0, a_1[j] = 1, a_1[l] = '-' \quad \forall \mathrm{l} \neq \mathrm{i}, \mathrm{l} \neq \mathrm{j}; \qquad (3)$$

$$a_2[i] = 0, a_2[l] = '-' \quad \forall l \neq i. \qquad (4)$$

Нетрудно видеть, что второй вектор поглощает первый. Из формулы (3) следует, что не может быть векторов, у которых признак i отсутствует, а признак j присутствует. А из формулы (4) следует, что вообще не может быть объектов, у которых признак i отсутствует. Понятно, что интервал запретов, задаваемый первым вектором, излишен и может быть удален.

Опишем теперь алгоритм построения расширенной матрицы запретов[5]. В качестве начальной возьмем матрицу запретов U. На каждом очередном этапе проверяем текущую матрицу на наличие смежных строк. Если такие строки есть, то проводим операцию обощенного склеивания и добавляем получившуюся строку в текущую матрицу. Далее мы удаляем все поглощаемые строки (если они есть) и переходим к следующему этапу. Данный процесс продолжаем до тех пор, пока в полученной с предыдущего этапа матрице находятся какие-либо смежные строки.

В результате получаем расширенную матрицу R, эквивалентную исходной U. Из приведенного алгоритма следует, что все строки матрицы R являются максимальными (задаваемые ими интервалы нельзя расширить не выходя за пределы множества $M(U)$). Проиллюстрируем работу приведенного алгоритма на следующем примере.

Пример 4. Пусть исходная матрица U состоит из двух строк:

$$U = \begin{pmatrix} 1 & 0 & - & \ldots & - \\ 1 & 1 & - & \ldots & - \end{pmatrix}.$$

Ясно, что обе строки матрицы являются смежными по второй компоненте. Применяя операцию обобщенного склеивания и добавляя новую строку, получаем матрицу

$$\overline{U} = \begin{pmatrix} 1 & 0 & - & \ldots & - \\ 1 & 1 & - & \ldots & - \\ 1 & - & - & \ldots & - \end{pmatrix}.$$

Заметим, что обе верхние строки данной матрицы поглощаются третьей. Удаляя их, приходим к расширенной матрице $R = (1, –, .., –)$, состоящей из единственной строки. Данная строка является максимальной. Действительно, единственный способ расширить задаваемый ее интервал, это присвоить первой компоненте значение '–' вместо 1. Понятно, что тогда соответствующий интервал запрета будет совпадать со всем бинарным пространством и выйдет за пределы исходного $M(U)$.

3.2. Построение матрицы поглощений

Полученная матрица расширений не является минимальной (по числу строк). Могут найтись комбинации максимальных строк из R, которые покроют все множество $M(U)$ (т. е. для любого вектора из $M(U)$ найдется поглощающая его строка из данной комбинации). Так что оставшиеся строки могут быть отброшены. Наша задача состоит в том, чтобы среди имеющихся комбинаций отобрать комбинации с минимальным числом строк. Пусть число всех элементов множества $M(U)$ и число строк в матрице R равно t и r соответственно. Обозначим через $V = \| v_{ij} \|$ матрицу поглощений между элементами множеств R и $M(U)$:

$$v_{ij} = \begin{cases} 1, & \text{если строка } i \text{ из } R \text{ поглощает строку } j \text{ из } M(U) \\ 0, & \text{иначе,} \end{cases}$$

$i = 1, \ .., \ r, \ j = 1, \ .., \ t.$

Таким образом, каждый столбец данной матрицы поглощений есть перечень тех максимальных строк из R, которые поглощают соответствующий данному столбцу элемент множества $M(U)$.

Поскольку при этом нам необходимо найти минимальное число включаемых в решение максимальных строк, то наша задача сводится к задаче нахождения кратчайшего строчного покрытия матрицы поглощений[6].

3.3. Задача нахождения кратчайшего строчного покрытия

Алгоритм приближеннего решения данной задачи приведен в работе А. Д. Закревского[7].

В качестве начальной берется матрица поглощений.

Назовем столбец *минимальным*, если он содержит минимальное число единиц среди всех остальных столбцов данной матрицы.

На очередном шаге среди всех столбцов матрицы, полученной на предыдущем этапе, ищется минимальный столбец. Далее рассматриваются те строки матрицы, которые соответствуют единичным значениям данного столбца. Среди этих строк ищется строка с максимальным числом единиц. Далее из матрицы вычеркивают полученную строку и столбец и переходят к следующему этапу.

Так продолжаем до тех пор, пока не охватим все столбцы исходной матрицы.

Из приведенного алгоритма следует, что мы на каждом этапе решаем самую простую в плане отбора строк задачу. А среди отобранных строк мы ищем оптимальную в плане возможного числа поглощаемых ею столбцов. Понятно, что приведенный алгоритм не гарантирует точного решения. Но в основе данного алгоритма лежат соображения, «повышающие» его оптимальность.

Разобьем исходную задачу восстановления пропусков на два этапа: обучение и распознавание. Первый этап — индуктивный. На нем происходит обработка данных обучающей выборки. На основе полученных результатов строится матрица взаимосвязей реальных объектов, своеобразная «модель мира». Второй этап — дедуктивный. На нем полученная модель используется для определения значения целевого признака тестового объекта. Ниже приведены формализованные алгоритмы для обоих этапов.

Алгоритм этапа обучения

Вход алгоритма

На вход алгоритма подается бинарная матрица с фиксированным числом столбцов n (матрица «объект-признак»). Число строк матрицы произвольно. Фиксируем ранг $k = 2$ интервалов запрета.

Тело алгоритма

1. Построение матрицы запрета U.

2. Эквивалентное преобразование матрицы U с помощью операций обобщенного склеивания смежных векторов и удаления поглощаемых векторов. Построение матрицы R максимальных векторов.

3. Построение множества (матрицы) $M(U)$. Построение матрицы поглощений V между элементами матриц $M(U)$ и R.

4. Нахождение приближенного кратчайшего строчного покрытия матрицы V.

Выход алгоритма

В качестве результата работы алгоритма выдается модифицированная (с минимально допустимым числом строк) матрица расширений $\tilde{R}$, эквивалентная исходной матрице R.

Алгоритм этапа распознавания

Вход алгоритма

На вход алгоритма подается бинарный вектор d с неизвестным (одним) значением целевого признака:

$$\exists l : d[j] = 1 \text{ или } \ d[j] = 0 \quad \forall \ j \neq l \quad \text{и} \quad d[l] = \text{'}-\text{'}.$$

В качестве начальной матрицы берется модифицированная матрица расширений $\tilde{R}$, полученная на предыдущем этапе обучения.

Цикл

На очередном шаге находим в текущей матрице столбец k с максимальным числом нулей и единиц среди всех столбцов, исключая столбец l. Если такого столбца нет, то переходим к пункту ***Выход***.

Иначе идем по всем строкам i текущей матрицы и сравниваем значения соответствующих ком-

понент строк матрицы и вектора d. Если они не совпадают

$$((v_{ik} = 0 \quad \text{и} \quad d[k] = 1) \quad \text{или} \quad (\mathrm{v}_{ik} = 1 \quad \text{и} \quad \mathrm{d[k]} = 0)),$$

то вычеркиваем соответствующую строку i из матрицы. Далее вычеркиваем столбец k из матрицы и возвращаемся к пункту ***Цикл***.

Выход

На выходе алгоритма анализируем столбец l полученной матрицы. Если все значения равны 1, то на выходе алгоритм выдает значение 0. Если все значения равны 0, то на выходе алгоритм выдает значение 1. Иначе значение неопределенно.

Напомним, что на каждом шаге пункта ***Цикл*** отбрасываются из дальнейшего рассмотрения все те строки матрицы, у которых соответствующий компонент не совпадает со значением компонента тестового объекта. Действительно, в этом случае тестовый объект заведомо не попадает в интервалы запрета, задаваемые данными строками. Поэтому эти строки никакой ценной информации не несут и могут быть отброшены.

Понятны также соображения, которыми мы руководствуемся на выходе алгоритма. Если в матрице, оставшейся от исходной матрицы $\tilde{R}$, есть только единицы, то разрешенным для целевого признака является только нулевое значение. И, наоборот, если в данной матрице присутствуют одни нули, то единственное разрешенное значение есть единица. В противном случае никакого определенного ответа мы дать не можем. Например, вследствие недостаточного объема обучающей выборки все значения интересующего нас столбца матрицы могут оказаться равными '–'.

4. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

Работу предложенного алгоритма проверим на материале граффити Хазарии из Маяцкого городища[8]. На основе анализа степени схематизации всех сохранившихся в целости изображений В. Е. Флерова приводит следующий список признаков: а — наличие схемы; б — тело животного передано одной линией; в — схема частично нарушена: отделены ноги, шея или голова; г — круп имеет объемные формы; д — ноги имеют сгибы в районе суставов; е — морда имеет замкнутый контур; ж — уши; з — глаза; и — грива (рога); к — хвост; л — копыта (лапы); м — пасть; н — грива имеет контур (рога); о — хвост имеет контур; п — глаз передан кружком; р — орнаментация фигуры животного; с — уши переданы контуром; т — 2 ноги переданы контуром; у — четыре ноги переданы контуром. Исходя из наличия (значение 1) или отсутствия признака (значение 0) совокупность всех целых изображений представлена в таблице 2.

Таблица 2

Список признаков уровня стилизации рисунков

№	№ рис.	а	б	в	г	д	е	ж	з	и	к	л	м	н	о	п	р	с	т	у
1	1а	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0
2	1б	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	1	0
3	2	0	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	1	0
4	3	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0
5	4	0	0	1	0	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0
6	5	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
7	8	1	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0
8	9	1	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0
9	10	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0
10	11	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
11	12	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
12	13	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
13	14	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
14	15	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
15	16б	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
16	16в	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
17	17	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0
18	18	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	19	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	20	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Окончание таблицы 2

№	№ рис.	а	б	в	г	д	е	ж	з	и	к	л	м	н	о	п	р	с	т	у
21	21а	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
22	22а	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
23	22б	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
24	23	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
25	24а	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	24б	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
27	24б	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
28	26	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
29	27	1	0	0	1	1	1	1	0	1	1	1	0	0	0	0	1	1	0	0
30	28	1	0	0	0	0	1	1	1	1	1	0	0	1	0	0	0	1	0	0
31	29	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1	0
32	30	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	31	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0

Будем восстанавливать каждый из признаков (их всего 19), используя метод *перекрестной проверки с исключением*. Суть данного метода состоит в том, что на каждом проходе из всей совокупности объектов исключается один, который становится тестовым. А оставшиеся объекты используются для построения матрицы запретов (их будет 32). На следующем проходе исключается другой объект и так далее. Всего число проходов будет равно числу объектов. За успех будем считать такое событие, при котором восстановленное алгоритмом значение целевого признака совпадает с реальным. Соответственно, в качестве неуспеха принимается событие, при котором алгоритм либо неверно восстанавливает значение, либо не выдает никакого.

Алгоритм распознавания можно существенно упростить, если с самого начала выбрасывать из рассмотрения все строки матрицы запретов, у которых значение компоненты, соответствующее целевому признаку, неопределенно. Данный алгоритм реализован автором статьи на языке программирования Delphi. Результаты работы алгоритма приведены в таблице 3.

Таблица 3

Результаты работы алгоритма распознавания

1	2	3	4	5	6	7	8	9	10
число испытаний	N признака	число успехов	доля успехов	N признака	число успехов	доля успехов	N признака	число успехов	доля успехов
33	а	14	0,42	з	30	0,91	п	31	0,94
33	б	26	0,79	и	15	0,46	р	32	0,97
33	в	21	0,64	к	7	0,21	с	23	0,7
33	г	30	0,91	л	23	0,7	т	27	0,82
33	д	31	0,94	м	22	0,67	у	33	1
33	е	12	0,36	н	28	0,85			
33	ж	8	0,24	о	33	1			

Здесь доля успешных событий (столбцы 4, 7, 10) рассчитываются как отношение числа успешных событий (столбцы 3, 6, 9) к общему количеству испытаний (столбец 1). Из данной таблицы следует, что средняя доля успешных событий равна 0,71. Заметим, что на некоторых признаках (например, ‘к’) алгоритм выдает крайне низкое число успешных событий. Такой результат можно объяснить тем, что в число неуспешных событий входят и события с неопределенными ответами, число которых в данном случае оказалось велико.

Автор признателен профессору Л. И. Бородкину за полезные консультации и обсуждения.

ПРИМЕЧАНИЯ

1 Шер Я. А. Алгоритм распознавания стилистических типов в петроглифах (К теории стиля в первобытном искусстве) // Математические методы в историко-экономических и историко-культурных исследованиях. М., 1977. С. 127–143.
2 Закревский А. Д. Логика распознавания. М., 2003. С. 144.
3 Флерова В. Е. Граффити Хазарии. М., 2013. С. 174.
4 Закревский А. Д. Указ. соч. С. 144.
5 Закревский А. Д. Алгоритмы синтеза дискретных автоматов. М., 2003. С. 512.
6 Там же.
7 Там же.
8 Флерова В. Е. Указ. соч. С. 174.