ПОИСК IF-THEN ПРАВИЛ В ДАННЫХ: ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ

М. Г. Асеев[1], В. А. Дюк[2]

Санкт-Петербургский институт информатики и автоматизации РАН
199178, Санкт-Петербург, 14-я линия ВО, д.39

[1]<maxim@datadiver.nw.ru>; [2]<duke@datadiver.nw.ru>

УДК 681.3
М. Г. Асеев, В. А. Дюк. **Поиск if-then правил в данных: проблемы и перспективы** // Труды СПИИРАН, Вып. 2, т. 2. — СПб.: Наука, 2005.
Аннотация. *В статье анализируются основные проблемы поиска if-then правил в данных: проблема сегментации признаков, проблема перебора вариантов, отсутствие критериев для оценки отдельных правил, проблема ложных закономерностей в многомерных данных, работа с нечеткими правилами и др. На примере системы Deep Data Diver раскрываются перспективы в рассматриваемой области.* — Библ. 4 назв.

UDC 681.3
M. G. Aseev, V. A. Duke. **If-then rules search in data: problems and perspectives** // SPIIRAS Proceedings. Issue 2, vol. 2. — SPb.: Nauka, 2005.
Abstract. *The article analyses the main problems of if-then rules search in data: the problem of attribute segmentation, the combinatorial problem, the lack of criteria for estimating certain rules, the problem of false laws in multidimensional data, etc. Taking the three unsophisticated tests and Deep Data Diver system as an example we reveal the major perspectives in the considered area.* — Bibl. 4 items.

1. Введение

Среди систем анализа данных, относящих себя к области Data Mining и KDD (Knowledge Discovery in Databases), видное место занимают системы для поиска if-then правил. С помощью таких систем решаются задачи прогнозирования, классификации, распознавания образов, сегментации БД, извлечения из данных «скрытых» знаний, интерпретации данных, установления ассоциаций в БД и др. Методы поиска if-then правил предъявляют минимальные требования к типу данных и применимы для обработки разнородной информации. Их результаты прозрачны для восприятия.

Наиболее популярные подходы в рассматриваемом классе аналитических систем реализуют алгоритмы построения деревьев решений и ограниченного перебора. Как правило, оценка эффективности указанных алгоритмов производится по конечному результату на независимых контрольных выборках или с помощью процедур кросс валидизации. Вместе с тем, из поля зрения нередко выпадают некоторые принципиальные недостатки используемых подходов, существенно снижающие их ценность для решения практических задач.

В настоящей статье мы попытаемся раскрыть некоторые нерешенные проблемы, объяснить их причины, и на примере новейшей системы Deep Data Diver обрисовать видимые перспективы.

Для иллюстрации возникающих проблем воспользуемся тремя тестами.

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

2. Тесты

2.1. ТЕСТ 1 — «Умение решать очевидные задачи»

В этом тесте предлагается задача разбиения на 2 класса множества объектов, равномерно распределенных на плоскости в произвольном квадрате (рис. 1). Квадрат разделен на 4 области линиями, проходящими через середины сторон. Каждый класс располагается в двух областях, симметричных относительно одной из диагоналей квадрата. Особенность подобной конфигурации данных заключается в том, что признаки X_1 и X_2 по-отдельности или интервалы на этих признаках, не обладают самостоятельной дискриминирующей способностью.

Решение представленной тестовой задачи очевидно. Каждый класс описывается двумя логическими правилами (всего 4 правила):

$$IF\ (X_1 > 4)\,\&\,(X_2 < 5)\ THEN\ Класс_1 = крестики$$

$$IF\ (X_1 < 5)\,\&\,(X_2 > 4)\ THEN\ Класс_1 = крестики$$

$$IF\ (X_1 < 5)\,\&\,(X_2 < 5)\ THEN\ Класс_2 = нолики$$

$$IF\ (X_1 > 4)\,\&\,(X_2 > 4)\ THEN\ Класс_2 = нолики$$

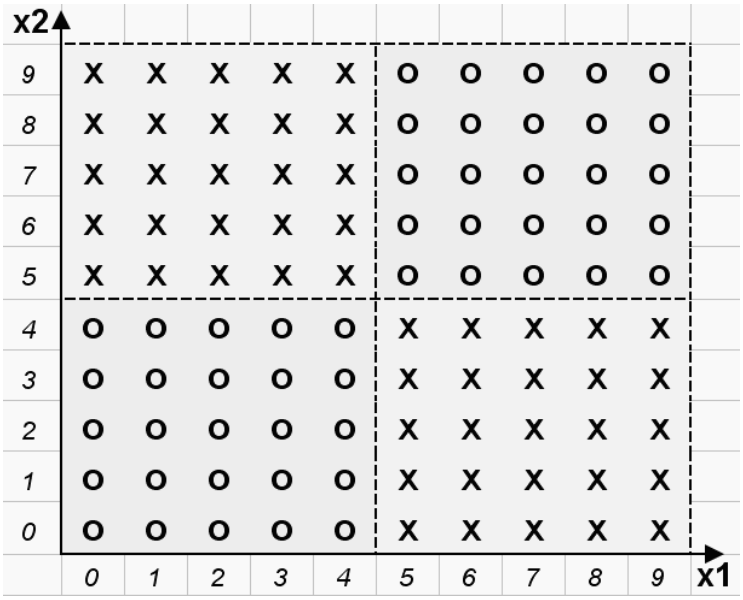

Рис. 1. Распределение объектов на плоскости анализируемых признаков.

2.2. ТЕСТ 2 — «Умение находить наиболее полные и точные правила»

Принцип формирования этого и подобных тестов следующий. Матрица объект-признак размера Np (N — число объектов, p — количество признаков) заполняется нулями и единицами (или любыми другими символами) со случайным равномерным распределением. В этой матрице выбираются участки строк различной длины (комбинации значений признаков), каждый из которых дублируется в матрице определенное число раз строго по вертикали. Тем самым создаются подгруппы объектов, для которых известно логическое правило, опи-

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

сывающее их полностью со 100 % точностью. Наборы подгрупп объединяются в классы, подлежащие распознаванию. Для большей чистоты эксперимента столбцы и строки общей матрицы переупорядочиваются случайным образом. Ставится задача найти в матрице данных введенные *известные* правила.

В конкретном тесте 2 таблица данных имеет следующие характеристики: количество объектов 400 (из них 100 объектов принадлежит 1 классу и 100 — второму, 200 объектов — случайным образом распределенные значения), 100 бинарных признаков, принимающих значения A или B. Требуется найти 4 известных логических правила, по 2 правила на каждый класс. Эти правила представляют собой комбинации от 7 до 15 элементарных логических событий. Фрагмент таблицы данных приведен на рис. 2.

Задача теста 2 далеко не самая трудная из встречающихся на практике. 100 бинарных признаков появляются в анализе, например, когда мы имеем дело всего с 10 исходными количественными признаками, которые при поиске логических закономерностей разбиваются на 10 интервалов каждый. Реальные задачи нередко содержат сотни и даже тысячи количественных, порядковых и категориальных признаков, а логические закономерности могут представлять собой комбинации из десятков и сотен элементарных событий. Если какая-либо система «не умеет» находить правила неограниченной длины, покрывающие максимально возможные количества объектов собственного класса, то аналитик рискует утонуть в море «обрывков» логических правил.

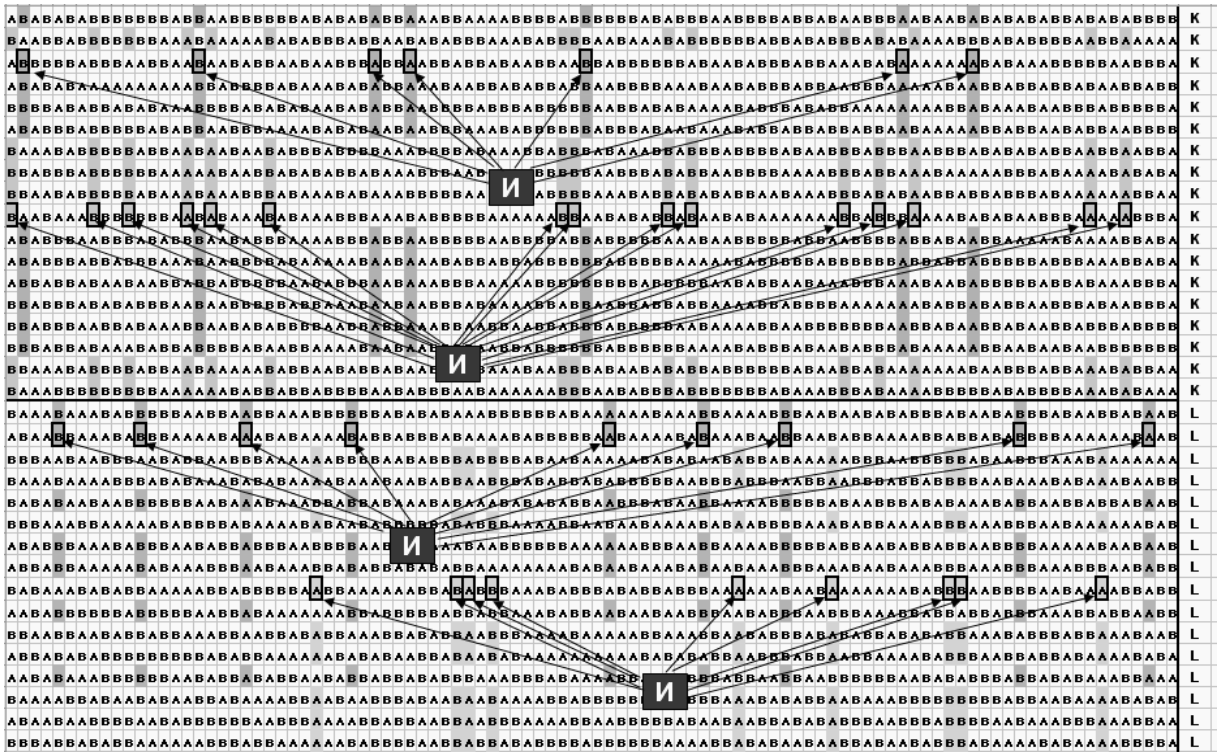

Рис. 2. Небольшой фрагмент бинарных тестовых данных
(выделены искомые комбинации значений признаков).

2.3. ТЕСТ 3 — «Ложные закономерности»

С увеличением количества анализируемых признаков в выборке конечного объема существенно возрастает риск обнаружения ложных закономерностей. Как показали проведенные исследования, указанная проблема слабо разработана в научной литературе, хотя ее высокая актуальность отмечается рядом

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

авторов. Известны лишь отдельные попытки предложить математические формулы для расчета уровня статистической значимости некоторых математических моделей, получаемых на основе анализа высокоразмерного эмпирического материала. Вместе с тем, эти попытки, формулируемые в рамках проблемы множественных сравнений, представляются узко специфичными и не имеют общего характера.

Тест 3 и подобные ему тесты необходимы для оценки достоверности логических правил в условиях данных высокой размерности. Он представляет собой таблицу данных со случайным распределением значений, полученными по схеме Бернулли. В нашем случае, мы будем предъявлять различным алгоритмам анализа бинарные данные (каждый признак может принимать значение A или B) с вероятностью 0,5.

Ниже мы остановимся на популярных подходах в Data Mining, предъявим соответствующим алгоритмам тестовые задачи и вскроем причины их неудовлетворительного функционирования.

3. Результаты тестирования

Для тестирования были выбраны три популярные системы поиска if-then правил See5/C5.0 (RuleQuest, Австралия), AnswerTree (SPSS) — деревья решений, и WizWhy (WizSoft) — ограниченный перебор.

3.1. Тестирование систем для построения деревьев решений

Тест 1 оказывается «непроходимым» для алгоритмов построения деревьев решений. Как уже отмечалось, признаки X_1 и X_2 или любые их интервалы, рассматриваемые по отдельности, не обладают способностью отделить крестики от ноликов. Поэтому уже на первом шаге определения «наилучшего» признака эти алгоритмы «дружно» отказываются от нахождения какого-либо логического правила.

Результат решения **теста 2** с помощью системы AnswerTree v. 2.1 приведен на рис. 3. Распознаваемые классы обозначены буквами A и K, 100 анализируемых признаков обозначены $a_1,\ldots,a_{100}$, каждый признак может принимать два значения – A и B. Тест упрощен — из матрицы данных исключено 200 случайным образом сгенерированных объектов.

Как следует из рисунка, система AnswerTree нашла 7 правил. Они располагаются на концах веток построенного дерева и обведены овалами. При этом только два правила можно считать более или менее удовлетворительными по точности и полноте охвата объектов собственного класса. Так, правило № 2

$$IF\ (a_{62} = B)\,\&\,(a_{72} = A)\ THEN\ Класс\ L$$

со 100 % точностью покрывает 39 из 100 объектов (строк) класса L. В свою очередь, правило № 7

$$IF\ (a_{62} = A)\,\&\,(a_{89} = A)\,\&\,(a_{84} = B)\,\&\,(a_{91} = B)\ THEN\ Класс\ K$$

относит 57 объектов к классу K с одной ошибкой. Таким образом, в целом тест 2 остался нерешенным. Система не сумела найти 4 известных экзаменатору правила, которые покрывают все объекты распознаваемых классов со 100 % точностью. Аналогично с данным тестом не справляются другие системы, реализующие те или иные алгоритмы построения деревьев решений.

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Для решения **теста 3** (ложные закономерности) использовалась система See5 v.1.16, которой была предъявлена таблица со случайными данными (200 бинарных признаков, 4000 объектов, 2 класса L и K по 2000 объектов). Эта система обнаружила в данных 72 if-then правила, некоторые из которых приведены в табл. 1.

При этом, в целом система See5 на основе выявленных логических правил произвела классификацию случайных данных в таблице с вероятностью ошибки 26,6 %.

Таблица 1. Некоторые правила, обнаруженные системой See5 в случайных данных

Rule 1: (23/1, *lift* 1.8)	*Rule* 28 : (24/2, *lift* 1.8)	*Rule* 32 : (32/4, *lift* 1.7)
$a_{56} = A$	$a_{47} = B$	$a_3 = A$
$a_{72} = A$	$a_{74} = A$	$a_4 = A$
$a_{74} = B$	$a_{114} = B$	$a_{39} = B$
$a_{133} = A$	$a_{119} = A$	$a_{140} = A$
$a_{140} = B$	$a_{127} = A$	$a_{157} = B$
$a_{158} = B$	$a_{177} = A$	$a_{171} = B$
$a_{180} = B$	$a_{194} = A$	$a_{182} = B$
$\Rightarrow$ *class L* [0.920]	$\Rightarrow$ *class K* [0.885]	$\Rightarrow$ *class K* [0.853]

В табл. 1 (исходя из существа предъявленного теста) нужно обратить внимание на цифры для каждого правила, заключенные в квадратные скобки. Эти цифры выражают точность полученного правила, оцениваемую с помощью соотношения Лапласа — $\frac{n-m+1}{n+2}$, где n — количество объектов, правильно покрытых правилом; m — количество объектов, ошибочно покрытых правилом. Как следует из полученных цифр, найденные правила обладают вполне удовлетворительной точностью (других оценок правил системой не предлагается). Поэтому у пользователя может возникнуть мнение, что найденные правила отражают устойчивые закономерности в данных, способные составить новое знание. Но это иллюзия. Априорно известно — обрабатывалась таблица со случайным распределением символов A и B. Аналогичные иллюзии нового знания получаются при тестировании системы построения деревьев решений See5 с помощью других вариантов теста 3 с различными количествами признаков и объектов.

Для справедливости следует отметить, что рассмотренные системы действительно функционируют с рекламируемой разработчиками высокой скоростью, имеют удобный интерфейс, развитые средства манипулирования исходными данными и т.п. Но перечисленные свойства, по-видимому, теряют свою привлекательность, когда мы начинаем вникать в принципиальные ограничения используемого аналитического подхода.

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

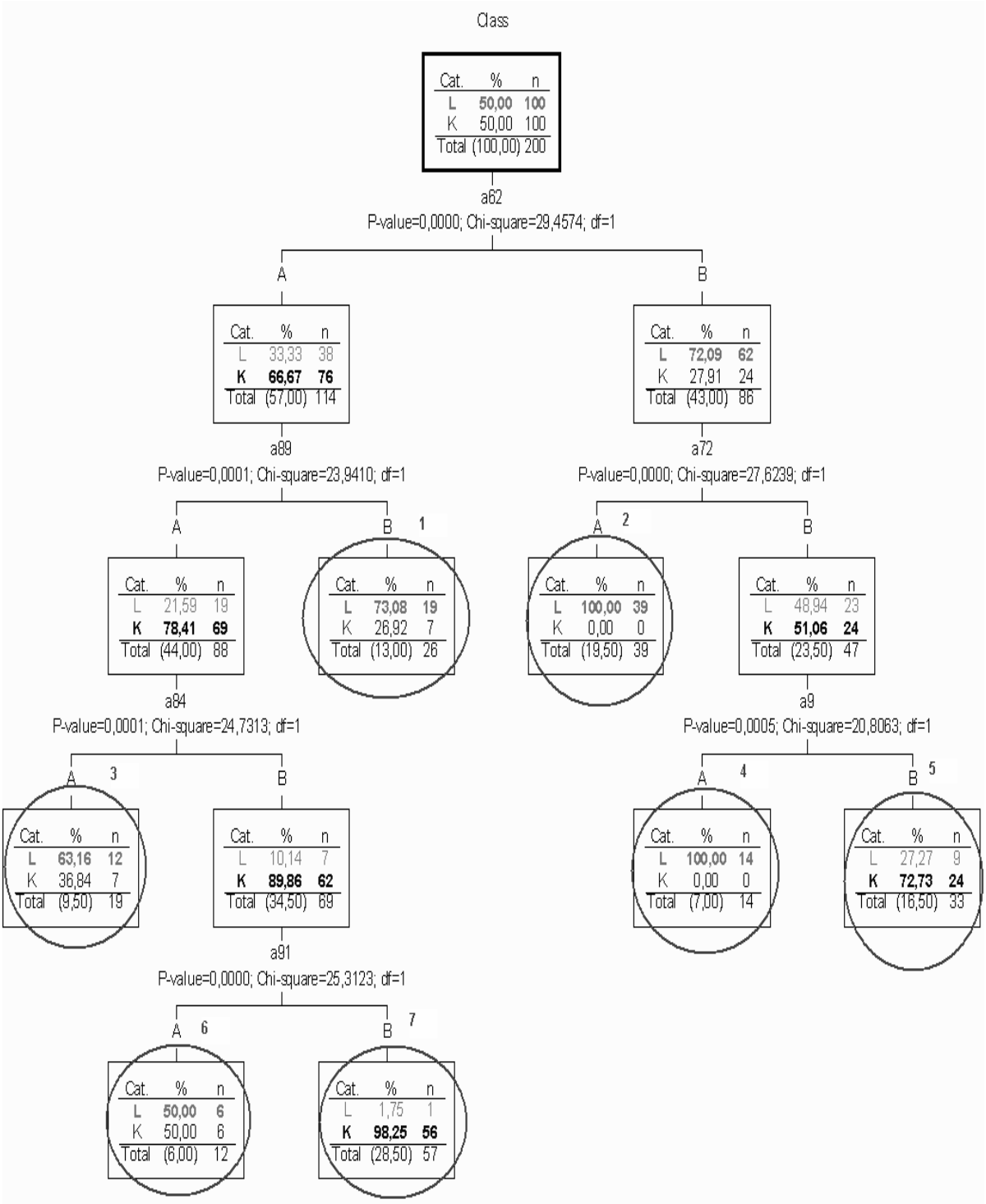

Рис. 3. Дерево решений, построенное системой AnswerTree по данным теста 2.

3.2. Тестирование системы, реализующей ограниченный перебор

Алгоритмы ограниченного перебора были предложены в середине 60-х годов и с тех пор они продемонстрировали свою эффективность при решении многих задач из самых различных областей. Эти алгоритмы вычисляют частоты комбинаций простых логических событий в подгруппах (классах) данных. При-

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

меры простых логических событий: $X = C_1; X < C_2; X > C_3; C_4 < X < C_5$ и др., где X — какой либо параметр (поле), C_i — константы. Ограничением служит длина комбинации простых логических событий. На основании сравнения вычисленных частот в различных подгруппах данных делается заключение о полезности той или иной комбинации для установления ассоциации в данных, для классификации, прогнозирования и пр.

Система WizWhy является современным представителем подхода, реализующего ограниченный перебор. Хотя разработчики системы не раскрывают специфику алгоритма, положенного в основу работы WizWhy, вывод о наличии здесь ограниченного перебора был сделан по результатам тщательного тестирования системы (изучались результаты, зависимости времени их получения от числа анализируемых параметров и др.). По-видимому, в WizWhy ограниченный перебор используется в модифицированном варианте с применением дополнительного алгоритма типа «Apriori». Авторы WizWhy акцентируют внимание на следующих общих свойствах системы:

Выявление ВСЕХ if-then правил; Вычисление вероятности ошибки для каждого правила; Определение наилучшей сегментации числовых переменных; Вычисление прогностической силы каждого признака; Выявление необычных феноменов в данных; Использование обнаруженных правил для прогнозирования; Выражение прогноза в виде списка релевантных правил; Вычисление ошибки прогноза; Прогноз с учетом стоимости ошибок.

В качестве достоинств WizWhy дополнительно отмечают такие:

На прогнозы системы не влияют субъективные причины; Пользователям системы не требуется специальных знаний в прикладной статистике; Более точные и быстрые вычисления, чем у других методов Data Mining.

Для убедительности авторы WizWhy противопоставляют свою систему нейросетевому подходу и алгоритмам построения деревьев решений и утверждают, что WizWhy, обладая более высокими характеристиками, вытесняет другие программные продукты с рынка Data Mining.

Ключевые «смелые» утверждения разработчиков о свойствах системы достаточно легко опровергнуть с помощью наших трех тестов на умение решать очевидные задачи, способность находить наиболее полные и точные логические правила, и отбраковывать ложные закономерности в многомерных данных.

Так, система WizWhy «категорически отказывается» находить какое-либо логическое правило в тесте 1. Это связано с неадекватностью утверждения об «Определении наилучшей сегментации числовых переменных». Здесь совершается классическая ошибка, связанная с непониманием того, что «часть не есть целое». Разработчики стремятся найти наилучшее разбиение количественных признаков на интервалы, рассматривая каждый признак изолированно от всей системы признаков.

В тесте 2 система WizWhy (использованы настройки системы по умолчанию) сумела найти только одно более или менее полное логическое высказывание, которое правильно относит к классу K 52 из 53 покрываемых объектов (записей):

$$IF\ (a_2 = B)\ \&\ (a_6 = B)\ \&\ (a_7 = A)\ \&\ (a_{34} = B)\ \&\ (a_{61} = A)\ THEN\ Класс\ K$$

Причина очевидной оплошности системы здесь кроется в том, что она способна находить логические правила, содержащие не более 6 элементарных событий. Это, собственно говоря, и есть ограничение «ограниченного перебо-

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ра» в действии – в рассмотренном тесте требуется найти заданные экзаменатором логические правила, включающие до 15 элементарных событий ($a_i = A$ или B).

Также следует отметить, что процесс поиска логических закономерностей сильно растянут во времени. В частности, выдачи результатов решения теста 2 на компьютере Intel Pentium III 733 МГц пришлось ожидать более 3 часов. Кроме того, система WizWhy помимо одного ценного правила, которое было приведено выше, выдала «в нагрузку» несколько тысяч правил, обладающих существенно более низкими точностью и полнотой.

Реакция системы WizWhy на тест 3 (ложные закономерности) представляет особый интерес, так как разработчики системы рассчитывают для каждого найденного правила уровень значимости α. Здесь применяется следующая формула (она приведена в руководстве к системе):

$$\alpha = \sum_{k=n}^{m} P_{N,M}(n,k),$$

$$P_{N,M}(n,k) = \frac{\binom{k}{M}\binom{n-k}{N-M}}{\binom{n}{N}},$$

где

m — количество объектов, покрываемых правилом во всей выборке;

n — количество объектов, правильно покрываемых правилом (покрываемых объектов в классе опорного объекта;

N — объем выборки в целом;

M — объем класса, описываемого соответствующим if-then правилом.

Системе WizWhy предъявлялось несколько вариантов теста 3 от (5 признаков на 100 объектов) до (200 признаков на 1000 объектов) — таблицы случайных бинарных данных разделялись ровно пополам на два класса. Вряд ли имеет смысл приводить здесь все полученные результаты (они чрезмерно объемны). Ограничимся наиболее яркими, с нашей точки зрения, примерами.

Так, для случая (100 признаков на 200 объектов) система нашла 14552 статистически значимых if-then правила !, с помощью которых классы объектов распознавались с точностью 95 %. В других тестовых примерах система обнаруживала еще более «статистически достоверные» логические правила. Например, для примера (100 признаков на 1000 объектов) в отчете системы был приведен такой результат:

$$IF\ (a_1 = A)\,\&\,(a_6 = A)\,\&\,(a_{29} = B)\,\&\,(a_{64} = A)\ THEN\ Class\ K$$

Точность правила: **0,865**

Правило покрывает **45** *объектов*

Уровень значимости < **0,0000001**

Примеры говорят сами за себя. Создается впечатление, что система WizWhy, реализующая принцип ограниченного перебора, как-будто бы прямо предназначена для обнаружения ложных закономерностей в случайных данных.

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

4. Выводы по результатам тестирования

1. Наиболее популярные аналитические инструменты Data Mining в ряде случаев оказываются не способны решать даже простейшие очевидные задачи.
2. Применяющиеся подходы к обнаружению знаний в базах данных выявляют лишь неточные фрагменты истинных логических закономерностей.
3. Инструменты для поиска логических правил в данных высокой размерности не способны отличать «ложные закономерности» от устойчивых регулярностей.

Кроме того, уместно добавить, что известные системы для поиска if-then правил не поддерживают функцию обобщения найденных правил и функцию поиска оптимальной композиции таких правил. Вместе с тем, указанные функции являются весьма существенными для построения баз знаний, требующих умения вводить понятия, метапонятия и семантические отношения на основе множества фрагментов знаний о предметной области.

5. Основные проблемы

5.1. Проблема «первого шага» — сегментация признаков

Все протестированные системы независимо от используемого подхода делают принципиальную ошибку уже на первом шаге своей работы.

Алгоритмы построения деревьев решений наивно пытаются найти наилучший признак (корень дерева), который «оптимальным» образом разделяет выборку на части. Никакие математические ухищрения не способны исправить основной дефект подобного подхода, связанный с тем, что признак вырывается из целостной системы описания многомерного объекта (записи базы данных). С нашей точки зрения, алгоритмы построения деревьев решений не выдерживают никакой критики и не заслуживают того необоснованного внимания, которое им уделяется в литературе.

В системах, реализующих переборный подход, принципиальная ошибка первого шага связана с поиском «оптимальной» сегментации количественных признаков. Об этом уже говорилось выше при обсуждении результатов решения теста 1 системой WizWhy — нельзя найти наилучшее разбиение количественных признаков на интервалы, рассматривая каждый признак изолированно от всей системы признаков.

Отмеченная проблема «первого шага» является производной от главной ключевой проблемы поиска if-then правил в данных. Эта ключевая проблема связана с тем, что в принципе здесь мы имеем дело с задачей полного перебора комбинаций элементарных логических условий. Считается, что данная задача при высокой размерности пространства признаков не может быть решена за приемлемое время в обозримом будущем даже с помощью суперкомпьютеров. Поэтому известные подходы к поиску if-then правил вынуждены использовать те или иные эвристических ограничения и их результаты нередко представляют собой «обрывки» истинных регулярностей в данных — «осколки знаний».

Так как заранее нельзя предугадать какие интервалы исходных признаков (элементарные события) окажутся наилучшими для искомых if-then правил, первый шаг работы алгоритма, претендующего на «высокий результат», должен заключаться в максимально мелком (с учетом доступных вычислительных мощностей) разбиении исходных признаков на интервалы. Это, конечно, каса-

Труды СПИИРАН. 2005. Вып. 2. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2005. Issue 2. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ется главным образом количественных признаков. По-видимому, не существует иного пути для решения задачи сегментации признаков.

5.2. Критерий для оценки отдельного правила

Предположим, что существует некий «фантастический» алгоритм Ψ, способный осуществлять полный перебор вариантов за обозримое время. Оказывается, даже в этом случае до сих пор отсутствует формальная постановка переборной задачи при поиске if-then правил. Что считать оптимальным решением при поиске if-then правил? Единственно ли оптимальное решение или существует множество оптимальных решений? Ниже мы попытаемся в какой-то мере прояснить эти вопросы.

Каждую запись базы данных можно «покрыть» множеством различных правил R_{ij} вида:

$$R_{ij} : IF\ (condition\ 1)\ \&\ (condition\ 2)\ \&\ \ldots\ \&\ (condition\ N)\ THEN\ (condition\ M)$$

Примеры условий: $X = C_1; X < C_2; X > C_3; C_4 < X < C_5$ и др., где X — какой либо параметр (поле), C_i — константы.

Будем использовать две характеристики if-then правила — точность и полноту. **Точность** правила R_{ij} это доля случаев B_j среди случаев A_i. **Полнота** правила это доля случаев A_i среди случаев B_j. Графически данные характеристики правил можно проиллюстрировать на рис. 4.

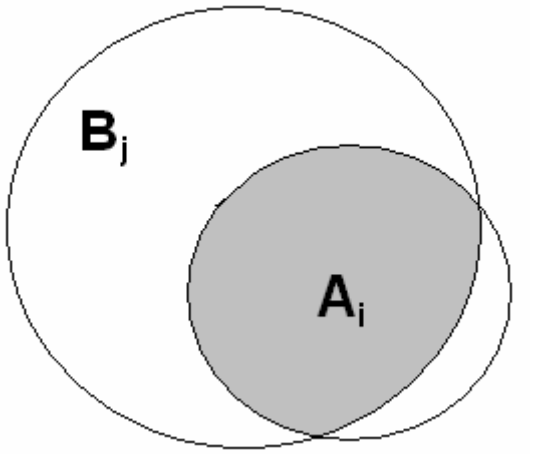

Рисунок 4. Иллюстрация точности и полноты if-then правила.

Литература

[1] *Дюк В. А.* Обработка данных на ПК в примерах. СПб: «Питер», 1997. — 240 с.

[2] *Duke V. A.* Latent knowledge extraction by methods of local geometry: development of expert system for keen appendicitis diagnostics. SPb.: Proc. Int. Conf. On Informatics and Control (ICI&C 97), vol.2. — P. 663–668.

[3] *Дюк В. А.* Формирование знаний в системах искусственного интеллекта: геометрический подход (ч. 4, глава 2) // В кн. Телемедицина. Новые информационные технологии на пороге XXI века. СПб.: Анатолия, 1998. — С. 367–389.

[4] *Дюк В. А.* От данных к знаниям — новые возможности обработки баз данных// Тр. Межд. научн. конф. «Интеллектуальные системы и информационные технологии управления (Псков, 19-23 июня 2000 г.). СПб.: Изд-во СПбГТУ, 2000. — С. 438–440.