Лекция 1

1. ВВЕДЕНИЕ

В различных областях человеческой деятельности (экономике, финансах, медицине, бизнесе, геологии, химии, и др.) повседневно возникает необходимость решения задач анализа, прогноза и диагностики, выявления скрытых зависимостей и поддержки принятия оптимальных решений. В настоящее время вследствие бурного роста объема информации, развития технологий ее сбора, хранения и организации в базах и хранилищах данных (в том числе Интернет-технологий), точные методы их анализа и моделирования зачастую отстают от потребностей реальной жизни. Здесь требуются универсальные, простые и надежные подходы, пригодные для обработки информации из различных областей, в том числе для решения проблем, которые могут возникнуть в ближайшем будущем. В качестве подобного базиса могут быть использованы технологии и подходы математической теории распознавания и классификации [1, 2, 8].

Действительно, данные подходы в качестве исходной информации используют лишь выборки описаний-наблюдений объектов, предметов, ситуаций или процессов (выборки прецедентов), при этом каждое отдельное наблюдение-прецедент записывается в виде вектора числовых значений отдельных его свойств-признаков. Выборки признаковых описаний являются обычно первичными исходными данными, которые повседневно возникают в различных предметных областях, и которые могут быть использованы для решения следующих задач:

- распознавание (классификация, диагностика) ситуаций, явлений, объектов или процессов с обоснованием решений;
- прогнозирование ситуаций, явлений, процессов и состояний по выборкам динамических данных;
- кластерный анализ и исследование структуры данных;
- выявление существенных признаков и минимизация описаний объектов;
- нахождение эмпирических закономерностей различного вида;

- нахождение нестандартных или критических случаев;
- формирование эталонных описаний образов.

Данные задачи возникают в различных предметных областях. Приведем некоторые примеры подобных приложений:

- обработка данных социологических опросов;
- прогнозирование тенденций изменения макроэкономических показателей;
- анализ финансовых данных и прогноз финансовых показателей;
- оценка экономического состояния предприятий и перспектив их инвестирования;
- проблемы прогнозирования экологических последствий по малым выборкам прецедентов;
- широкий круг задач медицины, связанных с созданием систем поддержки принятия диагностических решений, обработкой медицинской статистики, анализа эффективности лекарств и прогноза последствий лечения;
- задачи геологического прогнозирования;
- задачи экспериментальной физики, связанные с анализом накопленного экспериментального материала на этапах выявления качественных взаимосвязей между физическими параметрами и созданием приближенных математических моделей;
- задачи прогнозирования свойств новых органических соединений в химии на основе имеющегося банка исследованных органических соединений;
- обработка и анализ данных в биологии, с целью оптимизации селекционных и генетических исследований;
- обширный круг задач распознавания изображений.

1. Основные задачи анализа данных, распознавания, классификации и прогноза по прецедентам.

Исходной информацией являются описания объектов (ситуаций , предметов, явлений или процессов) S в виде векторов значений признаков $S = (x_1(S), x_2(S), ..., x_n(S))$, где признаки $x_i, i = 1,2,...,n$, характеризуют различные

стороны-свойства *S*. Одно из «свойств» *y(S)* объектов *S* (не входящее в состав признаков) считается «основным». Свойство *y(S)* принимает конечное число значений и для некоторых объектов $S_1, S_2, ..., S_m$ считается известным. Предполагается, что существует прямая связь между признаками и основным свойством (неизвестная пользователю).

Задача распознавания (прогноза, идентификации, «классификации с учителем») по прецедентам состоит в определении значения свойства *y(S)* объекта *S* по информации $S_1, S_2, ..., S_m, y(S_1), y(S_2), ..., y(S_m)$ (обучающей или эталонной выборке). Обычно вместо термина «основное свойство объекта» используют термин «класс объекта». Объекты, имеющие равные значения основного свойства считаются принадлежащими одному множеству (образу, классу объектов), и задача распознавания по прецедентам формулируется как задача отнесения объекта к одному из классов.

Задачу распознавания далее мы будем рассматривать далее как задачу классификации с учителем, и использовать следующую постановку и обозначения.

1. Пусть некоторое множество объектов является объединением конечного числа непересекающихся подмножеств, именуемых классами: $M = \bigcup_{i=1}^{l} K_i, \ K_i \cap K_j = \varnothing, \ i \neq j$. Данное разбиение известно лишь частично в виде выборки объектов $S_1, S_2, ..., S_m$ из данного множества, содержащей представителей всех классов. Для определенности будем считать, что $S_{m_{i-1}}, S_{m_{i-1}+1}, ..., S_{m_i} \in K_i, \ m_0 = 0, m_l = m, \ i = 1,2,...,l$.

2. Описание произвольного объекта S из M задается в виде совокупности из *n* значений признаков $X_1, X_2, ..., X_n$: $x_1(S), x_2(S), ..., x_n(S),$ где $x_i(S) \in M_i$ - значение признака X_i на объекте S. Здесь множества M_i задают область допустимых значений признака. Признак, как некоторое

свойство объекта, может быть произвольной природы (некоторая числовая характеристика, наличие или отсутствие какого-то свойства, изображение, функция, и т.д.). Мы будем рассматривать случаи числовых признаков, а именно:

а) $M_i = \{0,1\}$ - признак бинарный, обозначает отсутствие или наличие какого-либо свойства;

б) $M_i = \{0,1,...,k-1\}$ - признак k – значный, выражает степень выраженности некоторого свойства с конечным числом значений;

в) $M_i = [a_i, b_i]$, где $a_i, b_i -$ числа, либо символы $\pm\infty$.

Числовые признаки являются наиболее простыми и распространенными. Признаки номинальные (при сравнении которых нельзя использовать отношения «больше», «меньше», например «цвет», «социальное положение», «пол»), порядковые (где существенны или известны лишь отношения <, >, но не сама величина различия между значениями признаков), и другие более «сложные » признаки рассматриваться не будут. На практике, данные признаки сводятся к числовым, или для задач со сложными признаками создаются специальные методы. В качестве подобных примеров можно привести задачи распознавания зрительных и слуховых образов. Далее, для простоты записи, мы будем отождествлять объект с его описанием: $S = (x_1(S), x_2(S), ..., x_n(S))$.

Обучающая выборка будет задаваться таблицей обучения T_{nm} из m строк и n столбцов, в которой строками являются признаковые описания объектов, причем первые m_1 объектов из первого класса, следующие $(m_2 - m_1)$ - из второго, и т.д. Т.е. класс K_j представлен $(m_j - m_{j-1})$ эталонами, $m_0 = 0, m_l = m$. Строка $(x_1(S_j), x_2(S_j), ..., x_n(S_j))$ таблицы является признаковым описанием эталонного объекта S_j, а столбец $(x_i(S_1), x_i(S_2), ..., x_i(S_m))^t$ содержит значения признака x_i на эталонной выборке.

Примерами подобных задач являются:

- задачи медицинской диагностики, в которых по совокупности симптомов, данных лабораторных обследований и т.п. требуется поставить диагноз при заданном конечном наборе возможных их вариантов (здесь «основное свойство» есть наличие/отсутствие определенного заболевания);

- задачи технической диагностики, когда по набору значений косвенных технических параметров, показаниям датчиков и приборов требуется определить наличие или вид неисправности;

- прогноз эффективности инвестирования предприятия по его финансово-экономическим показателям (здесь «основное свойство» есть оценка эффективности инвестирования, качественная или в баллах);

- прогноз тенденций в политике, финансах и экономике, выявление и оценивание скрытых факторов;

- прогноз свойств органических/неорганических химических соединений и сплавов по составляющим компонентам и технологии производства;

- прогноз урожайности (интервала сбора культуры с единицы площади) сельскохозяйственных культур по описанию их состояния на различных стадиях роста и климатических условий;

- распознавание изображений, рукописных и других символов, подписей.

Задача распознавания объекта *S* состоит в определении класса $K_j, j = 1,2,...,l,$ которому принадлежит объект, на основе описания объекта $(x_1(S), x_2(S),..., x_n(S))$ и таблице обучения T_{nm}. Данная задача обычно решается в два этапа. Сначала по таблице обучения подбирается алгоритм, который наилучшим образом соответствует в каком-либо смысле таблице обучения. Данный этап называют этапом обучения распознаванию. На втором этапе, подобранный алгоритм непосредственно применяется для классификации нового объекта.

Данная постановка задачи распознавания имеет простую геометрическую интерпретацию. Множеству *M* (соответственно классам)

соответствуют область (подобласти) *n*-мерного векторного пространства признаковых описаний. Исходная информация об областях представлена в виде отдельных их точек. По данной исходной информации требуется определять принадлежность новых точек к одной из подобластей.

В практическом распознавании, в качестве допустимых решений, принимаются «отказы от распознавания», когда распознаваемый объект не похож на все предыдущие прецеденты, или когда он похож приблизительно в равной степени на объекты двух и более классов.

Задача **автоматической классификации (классификации без учителя, кластерного анализа, таксономии)** состоит в автоматическом разбиении заданной выборки объектов на классы (группировки) так, чтобы по совокупности значений признаков объекты одной группировки были близки друг другу, а объекты разных группировок – далеки. Полученные группировки являются приближенным макроописанием исходной выборки. Для простоты изложения, чтобы не возникало разночтений и путаницы между задачами классификации с учителем (распознавания) и классификации без учителя, для последней далее будут использоваться как правило термины «кластерный анализ», «кластеризация», и вместо терминов «классы» - термин «кластеры».

Задача **оценки информативности признаков и объектов** состоит в вычислении относительного вклада признака (объекта) в процесс распознавания.

Задача **минимизации признакового пространства** состоит в нахождении минимального набора признаков, обеспечивающего незначительное ухудшение качества (точности) распознавания относительно исходного набора признаков.

Задача поиска **логических закономерностей (логических зависимостей, извлечения знаний, data mining**) состоит в нахождении таких значений (интервалов значений) признаков, которые свойственны

многим объектам одного класса (с одинаковым значением свойства *y*). Это выражается в правилах следующего вида:

1. «для 80% эталонных объектов $S = (x_1(S), x_2(S), ..., x_n(S))$ второго класса ($y(S)=2$) выполнены условия: $(1.3 \le x_2(S) \le 5.2) \,\&\, (6.7 < x_5(S) \le 22.2) \,\&\, (x_6(S) = 1) \,\&\, (x_9(S) < 11)$».

2. «если $(3 \le x_1(S) \le 7.2) \,\&\, (1.9 \le x_4(S) \le 2.2) \,\&\, (5 < x_6(S)) \,\&\, (x_{11}(S) = 1)$, то с вероятностью 0.9 выполнено $y(S)=1$ (объект *S* принадлежит первому классу)».

Существуют и другие функции, параметры, величины, которые могут быть вычислены (хотя бы приближенно) по эталонным выборкам, и которые имеют интерпретацию и практическую ценность для пользователя (логические описания классов, логические корреляции, и др.).

Лекция 2

2. Алгоритмы распознавания, основанные на принципе частичной прецедентности

Принципиальная идея алгоритмов частичной прецедентности состоит в отнесении распознаваемого объекта в тот класс, в котором имеется большее число «информативных» фрагментов эталонов (частичных прецедентов), приблизительно равных соответствующим фрагментам объекта *S* [1, 2]. Вычисляются близости – «голоса» (равные 1 или 0) распознаваемого объекта к эталонам некоторого класса по определенным различным подмножествам признаков. Данные близости («голоса») суммируются и нормируются на число эталонов класса. В результате вычисляется нормированное число голосов, или оценка объекта *S* за класс $\Gamma_j(S)$ – эвристическая степень близости объекта *S* к классу K_j. После вычисления оценок объекта за каждый из классов, осуществляется отнесение объекта к одному из классов (т.е. распознавание класса объекта) с помощью решающего правила. Оптимальные значения параметров алгоритмов

распознавания (если он содержит некоторые неизвестные параметры), определяются из решения задачи оптимизации данной модели распознавания - находятся такие значения параметров, при которых точность распознавания является максимальной на некоторой заданной контрольной выборке.

2.1. Тестовый алгоритм распознавания

Тестовый алгоритм является одним из первых представителей широкого класса алгоритмов распознавания, основанных на принципе частичной прецедентности, в которых сравнение распознаваемого объекта с эталонными осуществляется по различным «информативным» и «сложно» вычисляемым подмножествам признаков. В качестве подобных подсистем признаков используются тупиковые тесты и их аналоги [1,2,4].

Для случая признаков целочисленных (k -значных) $x_i(S) \in \{0,1,...,k-1\}$ известны понятия теста и тупикового теста [21]. **Определение.** Подмножество столбцов $(i_1, i_2, \ldots, i_k)$ таблицы эталонов T_{nml} называется тестом, если любые две строки подтаблицы, образованной данными столбцами, различны при условии их принадлежности разным классам. Тупиковым называется тест, любое собственное подмножество которого не является тестом.

Для веществennозначных таблиц T_{nml} легко вводится аналог тестам таблиц конечнозначных, если строки подтаблиц считать различными при их различии с точностью до параметров $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$.

Тупиковый тест - минимальная подсистема признаков, разделяющая эталоны разных классов.

Пример 1. *Таблица* $T_{4,4,2}$ *:*

$$\left.\begin{matrix} S_1 & 0111 \\ S_2 & 1010 \end{matrix}\right\} \in K_1$$

$$\left.\begin{matrix} S_3 & 0011 \\ S_4 & 1000 \end{matrix}\right\} \in K_2$$

Здесь {1,2,3,4}, {2,3,4} – тесты; {1,2} – не является тестом. Таблица имеет два тупиковых теста: {2,3,4},{1,2,3}.

Пример 2. *Таблица* $T_{5,4,2}$ *:*

$$\left.\begin{matrix} S_1 & 10110 \\ S_2 & 10000 \end{matrix}\right\} \in K_1$$
$$\left.\begin{matrix} S_3 & 01101 \\ S_4 & 00011 \end{matrix}\right\} \in K_2$$

Здесь тупиковыми тестами являются наборы {1},{3,4},{4}.

Пример 3. *Таблица* $T_{6,4,2}$:

0	1	1	1	0	0
1	0	0	1	1	0
0	1	1	0	0	0
0	0	1	1	0	0

Здесь 6-й столбец не входит ни в один тупиковый тест. Столбец 4-й входит во все тесты, поскольку после его удаления будут равны строки разных классов (1-я и 4-я). Множество тупиковых тестов образуют наборы наборы {1,2,4},{2,3,4},{2,4,5}.

Рассмотрим кратко вопрос нахождения тупиковых тестов. Пусть

$T_{nml} = \left\| a_{ij} \right\|_{m \times n}$, где $a_{ij} = x_j(S_i) \in \{0,1,...,k-1\}$. Таблице T_{nml} ставится в

соответствие матрица сравнения $C = \left\| c_{ij} \right\|_{N \times n}$, где $c_{ij} = \begin{cases} 1, & a_{\nu j} \neq a_{\mu j}, \\ 0, & a_{\nu j} = a_{\mu j}, \end{cases} i = i(\nu, \mu)$

, $S_\nu \in K_u, S_\mu \in K_v, u \neq \nu$, $N = \sum_{\substack{i>j \\ i,j=1,2,...,l}} (m_i - m_{i-1})(m_j - m_{j-1})$.

Пример 3.

0	1	1	0
1	0	1	1
0	0	0	0
0	1	1	1
0	0	0	1

0	1	1	0
0	0	0	1
0	1	1	1
1	0	1	1
1	1	0	0
1	0	1	0

Таблица обучения $T_{4,5,2}$ *Матрица сравнения* $C_{6\times4}$

Столбцы $(i_1, i_2, \ldots, i_k)$ образуют покрытие строк матрицы $C = \|c_{ij}\|_{N\times n}$, если $\forall i = 1,2,...,N,\ \exists j \in \{i_1, i_2, ..., i_k\}: c_{ij} = 1$.

Покрытие называется тупиковым, если произвольное его собственное подмножество не является покрытием. Нетрудно убедиться, что наборы {1,2,4}, {1,3,4}, {2,3,4} образуют все тупиковые покрытия матрицы сравнения из примера 3. Каждому тупиковому тесту соответствует тупиковое покрытие строк матрицы сравнения и наоборот.

Тупиковый тест, состоящий из минимального числа столбцов, называется минимальным тупиковым тестом. Задача поиска минимального тупикового теста может быть сформулирована как задача целочисленного линейного программирования.

Рассмотрим следующую оптимизационную задачу:

$$\sum_{j=1}^{n} x_j \to \min,$$

$$\sum_{j=1}^{n} c_{ij} x_j \geq 1, i = 1,2,...,N, \qquad (1)$$

$$x_j \in \{0,1\}.$$

Легко видеть, что единичные компоненты решения данной задачи определяют минимальный тест. Если понимать под локальным минимумом данной задачи любой допустимый бинарный набор $(x_1, x_2, ..., x_n)$, в котором любая замена некоторой единицы на ноль делает его недопустимым, то множество локально-оптимальных решений задачи определяет множество тупиковых тестов.

Тестовый алгоритм распознавания определяется следующим образом.

Пусть заданы значения неотрицательных числовых параметров $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$. Значения числовых параметров $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$ задают пороги близости соответствующих признаков и могут вычисляются, например, как средний модуль разности значений признака по обучающей выборке : $\varepsilon_v = \frac{2}{m(m-1)} \sum_{i,j=1,i>j}^{m} |x_v(S_i) - x_v(S_j)|$. Пусть для заданных значений параметров $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$ и таблицы обучения T_{nml} найдено множество $\{T\}$ всех тупиковых тестов. Пусть $T = \{i_1, i_2, \ldots i_k\}$ - некоторый тупиковый тест таблицы T_{nml}, заданный соответствующими номерами ее столбцов. Для простоты обозначений тупиковые тесты будем записывать как множество номеров образующих тест столбцов.

Определим функцию близости между частями описаний некоторого эталонного объекта S_v и S, соответствующим данному тупиковому тесту:

$$B_T(S_v, S) = \begin{cases} 1, & |x_i(S) - x_i(S_v)| \le \varepsilon_v, \forall i \in T, \\ 0, & \text{иначе.} \end{cases} \qquad (2)$$

Назовем оценкой объекта S за класс K_j («мерой близости» к классу) следующую величину:

$$\Gamma_j(S) = \frac{1}{(m_j - m_{j-1})} \sum_{v=m_{j-1}+1}^{m_j} \sum_{T \in \{T\}} B_T(S_v, S). \qquad (3)$$

Таким образом, близость объекта к классу определяется как нормированная на число эталонов класса сумма близостей объекта ко всем эталонам данного класса по всем тупиковым тестам. Пусть вычислены оценки объекта S за каждый из классов. Объект S считается принадлежащим классу K_j, если $\Gamma_j(S) > \Gamma_i(S), i = 1,2,\ldots,l$. В случае наличия двух или более равных максимальных оценок, алгоритм отказывается от классификации *S*.

ЛЕКЦИЯ 3

(тестовый алгоритм, продолжение)

Существуют эффективные алгоритмы поиска тупиковых тестов [17,18]. Тем не менее, задача нахождения множества всех тупиковых тестов является вычислительно сложной комбинаторной задачей и не может быть решена на современных компьютерах даже для относительно небольших таблиц обучения (сотни объектов и признаков). Поэтому при решении практических задач вычисляют и используют в процедурах голосования обычно лишь часть тупиковых тестов [17,18] .

В Системе РАСПОЗНАВАНИЕ реализован стохастический вариант идеи тестового алгоритма. Из таблицы обучения выбираются случайно *N* подтаблиц, каждая из которых состоит из *3* строк таблицы обучения, *N* подтаблиц, состоящих из *4* строк таблицы обучения, и т.д., *N* подтаблиц, состоящих из *k* строк таблицы обучения (здесь *N* и *k* – управляющие параметры программы). Каждая подтаблица не обязана содержать эталоны из каждого класса, т.е. допускаются подтаблицы с числом строк меньшим числа классов. Это не приводит к дальнейшему использованию «плохих» тестов, так как каждому тесту впоследствии сопоставляется вес (качество) уже по полной обучающей выборке. Для каждой подтаблицы находятся все тупиковые тесты либо один минимальный тест в зависимости от выбранного алгоритма поиска. В последнем случае для таблицы обучения находится не более *N* (k-2)* минимальных тестов случайных подтаблиц.

Обозначим множество всех найденных тупиковых тестов для подтаблиц как и ранее через $\{T\}$. Пусть $M_1 = \{S_i, S_j\}$ множество пар строк таблицы обучения, принадлежащих равным классам, а M_2 - множество пар строк из разных классов. Число элементов множеств M_1 и M_2 обозначим, соответственно, через n_1 и n_2. Антиблизостью объектов по опорному множеству $T \in \{T\}$ назовем бинарную величину

$$D_T(S_\nu, S_\mu) = 1 - B_T(S_\nu, S_\mu) .$$

Определим «вес» опорного множества (в нашем случае теста T) согласно выражению (4)

$$Q_T = \frac{1}{n_2} \sum_{(S_i, S_j) \in M_2} D_T\left(S_i, S_j\right) - \frac{1}{n_1} \sum_{(S_i, S_j) \in M_1} D_T\left(S_i, S_j\right) \qquad (4)$$

, а через $w_T = \frac{Q_T}{\sum_{T \in \{T\}} Q_T}$ – его удельный вес. Данные величины показывают, как часто бывают близки эталонные объекты одного класса и далеки объекты разных классов по выбранному опорному множеству.

Окончательно, оценки распознаваемого объекта за классы K_j, *j=1,2,…,l,* вычисляются согласно следующей формуле:

$$\Gamma_j(S) = \frac{1}{2} \sum_{T \in \{T\}} w_T \left(\left|K_j\right|^{-1} \sum_{S_i \in K_j} B_T(S, S_i) + \left(m - \left|K_j\right|\right)^{-1} \sum_{S_i \notin K_j} D_T(S, S_i) \right).$$

Классификация осуществляется с помощью простейшего решающего правила.

В случаях практических задач с плохой отделимостью классов тупиковые тесты будут иметь большое число столбцов или могут вообще отсутствовать. Для управления отделимостью классов введен управляющий параметр программы (делитель ε - порогов), позволяющий увеличивать-

уменьшать близость объектов. Для небольших по количеству признаков таблиц обучения возможно вычисление всех тупиковых тестов и, соответственно, голосование по всем тупиковым тестам. Для реализации данного варианта в Системе предусмотрена кнопка «переборный алгоритм» .

2.2. Алгоритмы распознавания, основанные на вычислении оценок

Тестовый алгоритм стал первым широко известным в теории распознавания подходом, основанным на принципе частичной прецедентности. Ю.И.Журавлевым была предложена более общая формализация с различными способами выбора информативных подсистем признаков и формулами для вычисления оценок. Данный класс алгоритмов получил название «алгоритмы вычисления оценок» и включал «тестовый алгоритм» как частный случай. В настоящем разделе будут рассмотрены наиболее используемые алгоритмы, при этом тестовый алгоритм будет как и ранее специально выделенным.

Алгоритмы вычисления оценок (АВО) определяются как последовательное выполнение шести этапов, для каждого из которых имеются различные пути реализации. Ниже будут приведены лишь некоторые основные способы их выполнения. Подробно данные вопросы рассмотрены в [] и в прилагаемой библиографии.

2.2.1. Основные определения и этапы алгоритмов вычисления оценок

1. **Задание системы опорных множеств алгоритма.** Первым шагом определения АВО является задание множества подсистем признаков, по

которым осуществляется сравнение объектов. Пусть Ω_A - некоторая система подмножеств множества *{1,2,…,n}*, называемая системой опорных множеств алгоритма *A*. Элементы $\Omega=\{i_1, i_2,..., i_k\} \in \Omega_A$ называются опорными множествами алгоритма. Они определяют номера признаков, по которым сравниваются части эталонных и распознаваемых объектов. Каждому подмножеству $\Omega=\{i_1, i_2,..., i_k\}$ можно поставить во взаимно однозначное соответствие булевский вектор $\omega = \{\omega_1, \omega_2,..., \omega_n\}$, в котором $\omega_j = 1, j = i_1, i_2,..., i_n$, а остальные компоненты равны нулю.

Геометрически, множество всех 2^n n-мерных булевских векторов определяет дискретный единичный куб $E^n = \{\omega : \omega = (\omega_1, \omega_2,..., \omega_n)\}$, $\omega_i \in \{0,1\}, i = 1,2,..., n$.

Теоретические исследования свойств тупиковых тестов для случайных бинарных таблиц показали, что характеристические векторы «почти всех тупиковых тестов» имеют асимптотически (при неограниченном возрастании размерности таблицы обучения) приблизительно одну и ту же длину. Это явилось одним из обоснований выбора в качестве множества Ω_A всевозможных подмножеств *{1,2,…,n}* длины *k*. Значение *k* находится из решения задачи обучения (оптимизации модели) или задается экспертом. В итоге, широко распространенными подходами к выбору Ω_A являются (наряду с тупиковыми тестами) следующие два:

a) $\Omega_A = \{\Omega : |\Omega| = k\}$;

b) $\Omega_A = \{\Omega\}, \Omega \subseteq \{1,2,..., n\}, \Omega \neq \varnothing$.

Второй способ выбора системы опорных множеств, как всевозможных подсистем *{1,2,…,n},* является «усреднением» первого и не требует нахождения неизвестных параметров.

2. Задание функции близости. Пусть фиксировано некоторое опорное множество Ω и соответствующий ему характеристический вектор ω.

Фрагмент $x_{i_1}(S), x_{i_2}(S), \ldots x_{i_k}(S)$ объекта $S = (x_1(S), x_2(S), \ldots x_n(S)$, соответствующий всем единичным компонентам вектора ω, называется ω-частью объекта, и обозначается ω*S*. Под функцией близости $B_\Omega(S_i, S_j)$ будет пониматься функция от соответствующих ω-частей сравниваемых объектов, принимающая значение 1 («объекты близки») или 0 («объекты далеки»). Приведем примеры подобных функций.

a) $B_\Omega(S_\nu, S_\mu) = \begin{cases} 1, & |x_i(S_\nu) - x_i(S_\mu)| \le \varepsilon_i, \forall i : \omega_i = 1, \omega \leftrightarrow \Omega, \\ 0, & \text{иначе.} \end{cases}$

Здесь $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$ - неотрицательные параметры, именуемые «точности измерения признаков».

b) $B_\Omega(S_\nu, S_\mu) = \begin{cases} 1, & \sum_{i=1}^{n} |x_i(S_\nu) - x_i(S_\mu)| \le \varepsilon, \\ 0, & \text{иначе.} \end{cases}$

Здесь ε также некоторый неотрицательный параметр алгоритма.

3. Оценка близости объекта *S* к эталонному объекту S_i для заданной ω-части. Данная числовая величина формируется на основе функции близости и, возможно, дополнительных параметров.

a) $\Gamma_\Omega(S_i, S) = B_\Omega(S_i, S)$.

b) $\Gamma_\Omega(S_i, S) = w_\Omega B_\Omega(S_i, S)$, где w_Ω - «вес» опорного множества.

c) $\Gamma_\Omega(S_i, S) = \gamma_i (\sum_{i:\omega_i=1} p_i) B_\Omega(S_i, S)$. Здесь γ_i - параметры, характеризующие степень важности объекта S_i (информативность объекта), а $p_1, p_2, \ldots p_n$ - веса (информативность) признаков.

4. Оценка объекта *S* за класс K_j для заданной ω-части.

a) $\Gamma_j^\Omega(S) = \frac{1}{(m_j - m_{j-1})} \sum_{S_i \in K_j} \Gamma_\Omega(S_i, S)$.

5. Оценка объекта S за класс K_j .

a) $\Gamma_j(S) = \sum_{\Omega \in \Omega_A} \Gamma_j^{\Omega}(S)$.

b) $\Gamma_j(S) = v_j \sum_{\Omega \in \Omega_A} \Gamma_j^{\Omega}(S)$, где v_j - «вес» класса K_j. Например, в статистической теории распознавания аналогами параметров v_j являются априорные вероятности классов, которые характеризуют, насколько часто встречаются объекты различных классов .

6. Решающее правило.

Решающее правило – правило (алгоритм, оператор), относящее объект по вектору оценок за классы в один из классов, или вырабатывающее для объекта «отказ от распознавания». Отказ возникает обычно в случаях, когда оценки объекта малы за все классы (объект является принципиально новым, аналоги которого отсутствуют в обучающей выборке), или он имеет две или более близкие максимальные оценки за различные классы (объект лежит на границе классов). Таким образом, решающее правило ***r*** вычисляет для распознаваемого объекта S булевский вектор $r(S) = (\alpha_1(S), \alpha_2(S), ..., \alpha_l(S)), \alpha_i(S) \in \{0,1,\Delta\}$, где $\alpha_i(S) = 1$ означает отнесение объекта в класс K_i, $\alpha_i(S) = 0$ - принятие решения: «объект S не принадлежит классу K_i», $\alpha_i(S) = \Delta$ соответствует отказу от распознавания объекта относительно класса K_i. Приведем примеры решающих правил.

а) Простейшее решающее правило – отнесение объекта в класс, за который он имеет максимальную оценку, и отказ от распознавания в случае двух и более максимальных оценок.

$$\alpha_j(S) = \begin{cases} 1, & \Gamma_j(S) > \Gamma_i(S), i = 1,2,...,l, i \neq j, \\ 0, & \Gamma_j(S) < \max\{\Gamma_i(S), i = 1,2,...,l, i \neq j\}, \\ \Delta, & \text{иначе} \end{cases}$$

b) $$\alpha_j(S) = \begin{cases} 1, & \Gamma_j(S) > \Gamma_i(S) + \delta_1, i = 1,2,...,l, i \neq j, \\ & \Gamma_j(S) > \delta_2 \sum_{i=1}^{l} \Gamma_i(S), \\ 0, & \text{иначе.} \end{cases}$$, где δ_1, δ_2 - неотрицательные параметры решающего правила.

c) $$\alpha_i(S) = \begin{cases} 1, & \sum_{j=1}^{l} \delta_j^i \Gamma_j(S) \geq \delta_{l+1}^i, \\ 0, & \sum_{j=1}^{l} \delta_j^i \Gamma_j(S) \leq \delta_{l+2}^i, \\ \Delta, & \text{иначе.} \end{cases}$$, где $\delta_{l+1}^i > \delta_{l+2}^i$. (5)

Здесь $\delta_1, \delta_2, \delta_j^i$ - параметры алгоритма. В последнем случае наличие двух или более единиц интерпретируется как «объект вероятно принадлежит нескольким классам». Когда бинарный вектор состоит из одних нулей говорят, что данный объект – выброс, он не похож ни на один из классов, близких его аналогов ранее не наблюдалось.

Использование решающего правила означает фактически переход из признакового пространства в пространство оценок, в котором в качестве разделяющих классы функций используются гиперплоскости, проходящие через начало координат симметрично относительно новых координатных осей (случай **a**), пары гиперплоскостей (случай **b**), и наборы из *l-1* гиперплоскостей.

ЛЕКЦИЯ 4

2.2.2. Эффективные формулы вычисления оценок

После последовательной подстановки выражений на этапах 2-5 могут быть получены различные общие формулы для вычисления оценок $\Gamma_j(S)$. Например, выбирая первые примеры реализации различных этапов

будет получена следующая общая формула для вычисления оценок объекта S за классы K_j , $j=1,2,...,l$.

$$\Gamma_j(S) = \frac{1}{(m_j - m_{j-1})} \sum_{\nu=m_{j-1}+1}^{m_j} \sum_{\Omega \in \Omega_A} B_\Omega(S_\nu, S) . \qquad (6)$$

При выборе системы опорных множеств согласно вариантам а) или b) прямое вычисление оценок (6) представляется весьма трудоемким.

Действительно, при вычислении оценок (6) согласно а) требуется mC_n^k вычислений значений функции близости. В действительности необходимость выполнения всех данных вычислений отсутствует, поскольку существуют эффективные комбинаторные формулы вычисления оценок при многих вариантах реализации этапов 2-5 и различных системах опорных множеств.

Теорема 1. Пусть в модели вычисления оценок используются варианты а) выполнения этапов, тогда справедлива формула

$$\Gamma_j(S) = \frac{1}{(m_j - m_{j-1})} \sum_{S_i \in K_j} C^k_{d(S,S_i)} , \text{ где}$$

$$d(S,S_i) = \left| \left\{ \nu : |x_\nu(S_i) - x_\nu(S)| \le \varepsilon_\nu, \nu = 1,2,...,n \right\} \right| .$$

Доказательство.

Для доказательства достаточно показать, что

$$\sum_{\Omega \in \Omega_A} B_\Omega(S_\nu, S) = C^k_{d(S,S_i)} .$$

Действительно, данная сумма является общим числом «совпадений» фрагментов S_ν и S длины *k*. Но данное число и равно $C^k_{d(S,S_i)}$.

Следствие. Пусть в модели вычисления оценок используются варианты а) выполнения этапов 2-5 и b) - первого, тогда справедлива формула

$$\Gamma_j(S) = \frac{1}{(m_j - m_{j-1})} \sum_{S_i \in K_j} (2^{d(S,S_i)} - 1)$$, где

$$d(S,S_i) = \left| \left\{ \nu : \left| x_\nu(S_i) - x_\nu(S) \right| \le \varepsilon_\nu, \nu = 1,2,...,n \right\} \right|.$$

Действительно, $\Gamma_j(S) = \frac{1}{(m_j - m_{j-1})} \sum_{\nu=m_{j-1}+1}^{m_j} \sum_{\Omega \in \Omega_A} B_\Omega(S_\nu, S) =$

$$\frac{1}{(m_j - m_{j-1})} \sum_{S_i \in K_j} \sum_{k=1}^{n} C^k_{d(S,S_i)} = \frac{1}{(m_j - m_{j-1})} \sum_{S_i \in K_j} \sum_{k=1}^{d(S,S_i)} C^k_{d(S,S_i)} =$$

$$\frac{1}{(m_j - m_{j-1})} \sum_{S_i \in K_j} (2^{d(S,S_i)} - 1).$$

Теорема 2. Пусть в модели вычисления оценок используются варианты а) выполнения этапов 1,2,4,5, и с) этапа 3 (т.е. $\Gamma_\Omega(S_i, S) = \gamma_i (\sum_{i:\omega_i=1} p_i) B_\Omega(S_i, S)$), тогда справедлива формула

$$\Gamma_j(S) = \frac{1}{(m_j - m_{j-1})} \sum_{S_i \in K_j} \gamma_i (\sum_{t \in J(S,S_i)} p_t) C^{k-1}_{d(S,S_i)-1}$$, где

$J(S,S_i) = \left\{ \nu : \left| x_\nu(S_i) - x_\nu(S) \right| \le \varepsilon_\nu, \nu = 1,2,...,n \right\}$, $d(S,S_i) = \left| J(S,S_i) \right|$.

Доказательство.

По определению, $\Gamma_j(S) = \frac{1}{(m_j - m_{j-1})} \sum_{\nu=m_{j-1}+1}^{m_j} \gamma_\nu \sum_{\Omega \in \Omega_A} (\sum_{i \in \Omega} p_i) B_\Omega(S_\nu, S)$

$= \frac{1}{(m_j - m_{j-1})} \sum_{\nu=m_{j-1}+1}^{m_j} \gamma_\nu (\sum_{i \in J(S,S_\nu)} p_i) \varphi(S, S_\nu)$. Действительно, из определения

функции близости следует, что в сумму могут войти лишь те веса признаков, по которым объекты не различимы с точностью до соответствующего порога. Коэффициенты при данных p_i для фиксированного S_v будут равны в силу симметрии. Но тогда $\varphi(S, S_v)$ будет равно числу способов выбора из оставшихся $d(S, S_v)$-1 признаков по *k-1* признаку, т.е. $C^{k-1}_{d(S,S_v)-1}$. Теорема доказана.

В теории алгоритмов вычисления оценок существуют и другие эффективные формулы вычисления оценок для более сложно определенных способов их вычисления.

2.3. Алгоритмы голосования по представительным наборам

Другими алгоритмами данного вида являются алгоритмы типа “Кора” /16,12/, в которых опорные множества связаны со значениями признаков конкретных объектов.

Пусть $S_v \in K_j$. Набор $u = \{x_{i_1}(S_v), x_{i_1}(S_v), ..., x_{i_1}(S_v),\}$ называется представительным $\varepsilon -$ набором (или просто набором) для класса K_j, если для любого $S_\mu \in T_{nml}, S_\mu \notin K_j$ система неравенств $\left|x_t(S_v) - x_t(S_\mu)\right| \le \varepsilon_t, t = i_1, i_2, ..., i_k$ несовместна. Представительный набор называется тупиковым, если любой его собственный поднабор не является представительным набором, т.е. для любого его поднабора существует равный ему поднабор в каком-либо другом классе.

Пусть V_j – множество тупиковых представительных наборов для класса K_j. Оценку объекта S по заданному тупиковому представительному набору *u* вычисляем по одной из формул:

а) $\Gamma_u(S) = B(u, \omega S)$, где $\omega \leftrightarrow \Omega = \{i_1, i_1, ..., i_1\}$ (оценка совпадает с близостью набора и соответствующей ω - части объекта S);

b) $\Gamma_u(S) = \gamma_u B(u, \omega S)$ (близость умножается на вес набора);

c) $\Gamma_u(S) = (p_{i_1} + p_{i_2} + ... + p_{i_k}) B(u, \omega S)$ (вес близости оценивается как сумма весов соответствующих признаков).

Тогда $\Gamma_j(S) = \frac{1}{|V_j|} \sum_{u \in V_j} \Gamma_u(S)$. В дальнейшем для краткости прилагательное «тупиковый» будем убирать.

Отметим, что для нахождения множества представительных наборов класса необходимо объединить множества представительных объектов, связанных с отдельными эталонами. Для нахождения же представительных наборов, связанных с фиксированным эталоном, достаточно найти тупиковые тесты таблицы, состоящей из выделенного эталона $S_v \in K_j$ и эталонов дополнения CK_j данного класса по обучающей выборке.

Лекция № 5

2.4. Оптимизация моделей распознавания

Описания всех трех приведенных подходов включают неизвестные числовые параметры. Фактически, каждый из подходов представляет параметрическое множество алгоритмов распознавания, где конкретный алгоритм задается фиксацией значений параметров.

Так в описание тестового алгоритма при голосовании по тупиковым тестам могут быть введены параметры, характеризующие веса признаков и эталонов. Пусть выбрано также общее решающее правило с). Тогда модель тестовых алгоритмов $\mathrm{M}_T(\varepsilon, p, \gamma, \delta)$ содержит множество алгоритмов $\{A_T(\varepsilon, p, \gamma, \delta), \varepsilon \geq 0{,}1 \geq p \geq 0{,}1 \geq \gamma \geq 0\}$. Общее число параметров модели равно $2n + m + l \times (l+1)$.

Аналогично, модель $\mathrm{M}_k(k, \varepsilon, p, \gamma, \delta)$ вычисления оценок, в которой систему опорных множеств составляют всевозможные подмножества из *k* признаков,

является множеством алгоритмов

$\{A_k(k,\varepsilon,p,\gamma,\delta),1\geq k\geq 0,k-\text{целое},\varepsilon\geq 0,1\geq p\geq 0,1\geq\gamma\geq 0\}$. Модель голосования по представительным наборам $\mathrm{M}_V(p,\gamma,\delta)$, при вычислении оценок объекта по представительному набору согласно с), является параметрическим множеством алгоритмов $\{A_u(\varepsilon,p,\delta),\varepsilon\geq 0,1\geq p\geq 0\}$.

Стандартная постановка задачи поиска наилучших алгоритмов заданной модели состоит в следующем.

Пусть дано параметрическое множество распознающих алгоритмов $\{A(y\},y\in D\}$ и на нем определен числовой функционал $\varphi(A)$ качества алгоритм. Требуется найти такой алгоритм $A^*\in\{A\}$, который доставляет экстремум функционалу: $\varphi(A^*)=\underset{A\in\{A\}}{extr}\varphi(A)$.

Обычно проблема оптимизации решается следующим образом.

Пусть задана таблица контрольных объектов T'_{nql}, аналогичная таблице обучения, т.е. состоящая из разбитых на *l* классов *m* числовых строк – описаний объектов $S'_i=(x_1(S'_i),x_2(S'_i),...,x_n(S'_i))$ с помощью *n* признаков. Для определенности считаем, что

$S'_i\in K_j,i=q_{j-1}+1,q_{j-1}+2,...,q_j,q_0=0,q_l=0$.

Пусть $\alpha_{ij}=\begin{cases}1, & S'_i\in K_j,\\ 0, & S'_i\notin K_j.\end{cases}$ Обозначим также $\alpha_{ij}^A=\alpha_j(S'_i)$.

Определение. Стандартным функционалом качества распознавания называется функционал $\varphi(A)=\frac{1}{ql}\sum_{i=1}^{q}\sum_{j=1}^{l}\left|\alpha_{ij}-\alpha_{ij}^A\right|$.

Мы будем обычно использовать не данный функционал (доля ошибок), а ему обратную величину – долю правильных ответов.

Пусть используется общее решающее правило с), когда условием правильной классификации является выполнение системы из *l* неравенств. Тогда условием правильного распознавания некоторого объекта $S_v \in K_j$ является выполнение следующей системы неравенств:

$$\begin{cases} \sum_{i=1}^{l} \delta_i^j \Gamma_i(S'_v) \geq \delta_{l+1}^j, \quad S'_v \in K_j, \\ \sum_{i=1}^{l} \delta_i^j \Gamma_i(S'_v) < \delta_{l+1}^j, \quad S'_v \notin K_j. \end{cases} \qquad (7)$$

Пусть *Z* – система, состоящая из *m* подсистем (7) относительно переменных $y \in D$. Тогда задача оптимизации стандартного функционала качества может быть сформулирована в терминах систем неравенств следующим образом.

Найти совместную подсистему системы Z при условии $y \in D$, *содержащую максимальное число систем (7), и некоторое его допустимое решение.*

2.5. Оценка информативности признаков и эталонов

Алгоритмы вычисления оценок являются удобным инструментом для решения задач оценки информативности (важности) признаков и эталонов. Под данными характеристиками понимаются числовые оценки того, насколько высок вклад признака/эталона в процессе распознавания.

Пусть заданы обучающая и контрольная выборки и некоторый алгоритм. Обозначим через $\varphi(A)$ значение функционала качества распознавания алгоритмом, через $\varphi(A^i)$ - значение функционала качества алгоритма, построенного по данным обучения без учета i-го признака, а через $\varphi(A_j)$ - значение функционала качества алгоритма, построенного по данным обучения без j-го эталона.

Определение. Мерой информативности (весом) признака называется величина $p_i = \frac{\varphi(A) - \varphi(A^i)}{\varphi(A)}$. Мерой представительности (весом) объекта называется величина $\gamma_j = \frac{\varphi(A) - \varphi(A_j)}{\varphi(A)}$.

Данный подход естественен, нагляден и пригоден при использовании любого алгоритма распознавания.

При малых выборках данные величины принимают небольшой набор значений и могут быть грубыми оценками. В данном случае можно использовать другие, эвристические величины, характеризующие факт влияния признака (объекта).

Пусть *A* – некоторый алгоритм типа вычисления оценок, $\Gamma_j(S'_i)$ - оценка контрольного объекта S'_i за класс K_j, $\Gamma_j^{fea(\mu)}(S'_i)$ - та же оценка без учета признака x_μ, $\Gamma_j^{obj(v)}(S'_i)$ оценка контрольного объекта S'_i за класс K_j без учета эталона S_v.

Определение. Мерой информативности (весом) признака называется величина $p_\mu = \frac{\sum_{j=1}^{l} \sum_{S'_i \in K_j} (\Gamma_j(S'_i) - \Gamma_j^{fea(\mu)}(S'_i))}{\sum_{j=1}^{l} \sum_{S'_i \in K_j} \Gamma_j(S'_i)}$. Мерой представительности (весом) объекта называется величина $\gamma_v = \frac{\sum_{j=1}^{l} \sum_{S'_i \in K_j} (\Gamma_j(S'_i) - \Gamma_j^{obj(v)}(S'_i))}{\sum_{j=1}^{l} \sum_{S'_i \in K_j} \Gamma_j(S'_i)}$.

Данное определение подразумевает, что при удалении признака (эталона) теряется часть сходства объектов к своим классам, и эти потери тем больше, чем более важен данный признак (соответственно объект).

Лекция № 6

3. Логические закономерности классов.

В настоящем разделе будут рассмотрены методы поиска закономерностей вида *«если* $A_1(S)\,\&\,A_2(S)\,\&\ldots\&\,A_k(S)$ *то* $S \in K_i$*»*. Здесь A_1, A_2, …, A_k – одноместные предикаты «простейшего вида», зависящие от одного какого-либо признака. Предполагается, что условия данного вида выполняются если не на всех объектах обучающей выборки из некоторого класса, то по крайней мере на многих эталонах класса. Отметим, что к данной постановке сводятся задачи выявления логических связей между набором некоторых независимых величин (признаков) и зависимой. Если зависимая величина является дискретной, то разбиение на классы задают ее значения, если непрерывной (или с высоким уровнем значности) – интервалами значений.

Представительные наборы классов (и их ε - окрестности) также могут рассматриваться как закономерности классов. Здесь представляет интерес к расширению определению понятия представительных наборов путем введения критериев из качества и создания процедур нахождения оптимальных наборов. Второй принципиальный вопрос состоит в выборе ε - порогов, причем, естественно, данные пороги должны быть индивидуальными для каждого набора. В настоящем разделе будет рассмотрен вопрос обобщения представительных наборов – поиск оптимальных признаковых подпространств и окрестностей значений признаков будут осуществляться в рамках решения различных оптимизационных задач.

Рассмотрим следующее множество элементарных предикатов, параметрически зависящих от числовых неизвестных $c_j^1, c_j^2, j = 1,2,\ldots,n$:

$$P_j(c_j^1, c_j^2, x_j) = \begin{cases} 1, & c_j^1 \le x_j \le c_j^2, \\ 0, & \text{иначе.} \end{cases}$$

Пусть $\Omega \subseteq \{1,2,\ldots,n\}$.

Определение. Предикат $$P^{\Omega}(c^1, c^2, x) = \mathop{\&}_{j\in\Omega} P_j(c_j^1, c_j^2, x_j) \qquad (1)$$

называется логической закономерностью класса K_λ, если

1. $\exists\ S_t \in K_\lambda :\ P^{\Omega}(c^1, c^2, S_t) = 1,$
2. $\forall\ S_t \notin K_\lambda,\ \ P^{\Omega}(c^1, c^2, S_t) = 0,$
3. $\varphi(P^{\Omega}(c^1, c^2, x)) = \underset{\{P^{O}(d^1, d^2, x)\}}{extr} \varphi(P^{O}(d^1, d^2, x))$, где

φ - критерий качества предиката.

В 3. поиск экстремума проводится по множеству всевозможных предикатов вида (1).

Предикат (1), удовлетворяющий только первым двум ограничениям, называется допустимым предикатом рассматриваемого класса.

Предикат (1), удовлетворяющий только первому и третьему ограничениям, называется частичной логической закономерностью класса K_λ.

Стандартным критерием качества предиката класса K_λ будем называть следующий критерий: $\varphi(P^{\Omega}(c^1, c^2, S)) = \left|\{S_i : S_i \in K_\lambda, P^{\Omega}(c^1, c^2, S_i) = 1\}\right|$.

Логические закономерности со стандартным критерием качества имеют простую геометрическую интерпретацию. По данным обучающей выборки требуется найти прямоугольный гиперпараллелипипед, лежащий в некотором признаковом подпространстве, содержащий максимальное число эталонов из класса K_λ и только класса K_λ.

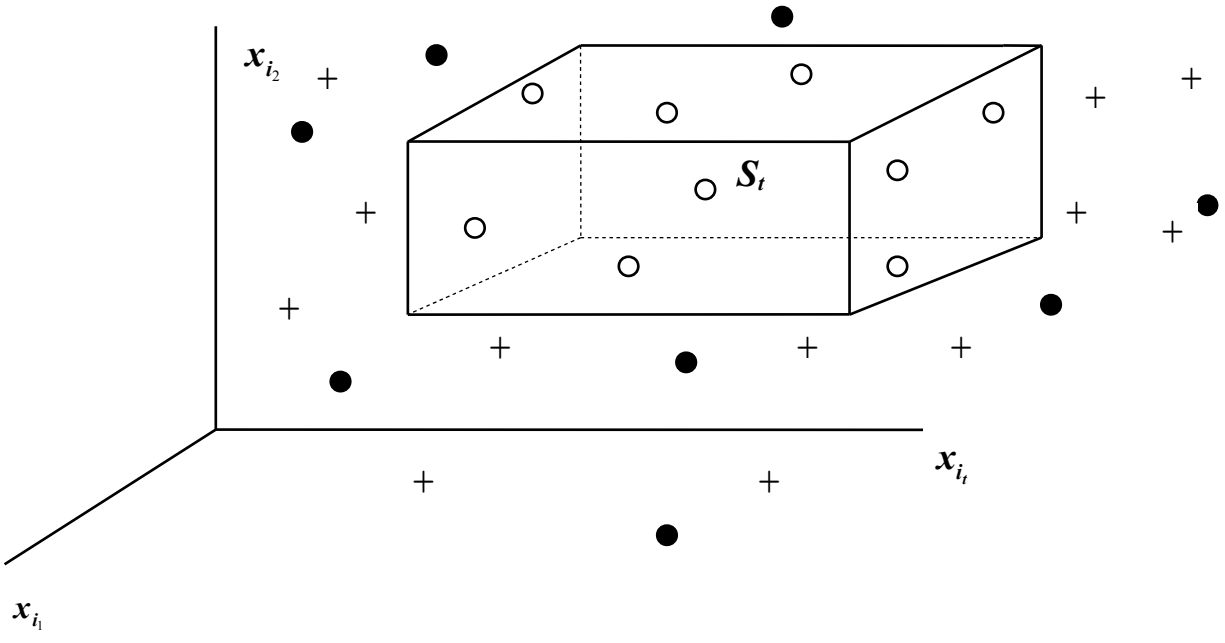

Рис.1 *Геометрическое представление логической закономерности класса K_λ с опорным эталоном S_t*

2.6.1. Нахождение оптимальных логических закономерностей (поиск оптимальных окрестностей представительных наборов).

Задача поиска логических закономерностей классов для стандартного критерия качества предикатов является сложной задачей нелинейной дискретной оптимизации. Создание методов поиска

Для создания эффективных методов их поиска рассмотрим "более слабый" вариант, основанный на следующих двух ограничениях.

1. Множество логических закономерностей "базируется" на эталонах. На геометрическом языке, в центре каждого искомого гиперпараллелипипеда находится один из эталонов класса.
2. Вместо стандартного функционала качества предикатов рассматривается некоторый линейный по признакам эвристический функционал.

Пусть фиксирован произвольный эталон $S_t \in K_\lambda$.

Будем искать предикаты вида

$$P_t^{\Omega,\varepsilon}(x) = \&_{j\in\Omega} P_{tj}(\varepsilon_j, x_j), \quad \text{где} \quad \Omega \subseteq \{1,2,...,n\},$$

$$P_{tj}(\varepsilon_j, x_j) = \begin{cases} 1, & a_{tj} - \varepsilon_j \le x_j \le a_{tj} + \varepsilon_j, \\ 0, & \text{иначе.} \end{cases} \qquad (2)$$

Отметим, что $P_t(S_t)=1$, т.е. первое условие определения логических закономерностей класса K_λ выполнено. Объект S_t будем называть опорным для предикатов (2).

В качестве критерия оптимальности предиката возьмем функционал

$$f(P_t^{\Omega,\varepsilon}) = \sum_{j\in\omega} \varphi_{tj}(\varepsilon_j),$$

$$\varphi_{tj}(\varepsilon_j) = \frac{1}{|K_\lambda|}\sum_{S_i\in K_\lambda} \rho_j(\varepsilon_j, S_t, S_i) + \frac{1}{|CK_\lambda|}\sum_{S_i\in CK_\lambda}(1-\rho_j(\varepsilon_j, S_t, S_i)) \qquad (3)$$

где антиблизость $\rho_j(\varepsilon_j, S_t, S_i) = \rho_j(\varepsilon_j, x_j(S_t), x_j(S_i)) = \begin{cases} 0, & |a_{tj} - a_{ij}| \le \varepsilon_j, \\ 1, & \text{иначе.} \end{cases}$

Здесь $|K_\lambda|, |CK_\lambda|$ означают число эталонных объектов принадлежащих и не принадлежащих отмеченному классу соответственно.

Лекция № 7

Вычислим значения $|a_{tj} - a_{ij}|$, $i=1,2,...,m,$ и упорядочим все различные их значения по возрастанию в виде следующих последовательностей

$$r^j : r_{j1}, r_{j2}, ..., r_{ju_j}, \; r_{jv} < r_{jw} \text{ при } v<w. \qquad (4)$$

Очевидно, $r_{j0} = 0.$

1. Подпоследовательность $r_{ju}, u = v, v+1, \ldots v+w,$ назовем подпоследовательностью первого типа последовательности (4), если

а) для любого ее элемента $r_{ju}, v \le u \le v+w,$ не существует строк

$S_i \in CK_\lambda: \left|a_{tj} - a_{ij}\right| = r_{ju}$,

б) любое возможное расширение подпоследовательности дополнением элементов слева или справа нарушает свойство а).

2. Подпоследовательность $r_{ju}, u = v, v+1, \ldots v+w,$ назовем подпоследовательностью второго типа последовательности (4), если

а) для любого ее элемента $r_{ju}, v \le u \le v+w,$ не существует строк

$S_i \in K_\lambda:, \left|a_{tj} - a_{ij}\right| = r_{ju}$,

б) любое возможное расширение подпоследовательности дополнением элементов слева или справа нарушает свойство а).

Докажем, что область $D=\{\varepsilon_j \ge 0,\ j=1,2,\ldots n\}$ допустимых значений параметров ε можно ограничить некоторым конечным множеством $D^* \subseteq D$ таким, что $\min\limits_{\varepsilon \in D} f(P_t^{\Omega,\varepsilon}) = \min\limits_{\varepsilon \in D^*} f(P_t^{\Omega,\varepsilon})$. (5)

Очевидно, что условие (5) будет выполнено, если в качестве D^* взять множество $D'=\{\varepsilon=(\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n):\ \varepsilon_i \in r^i\ ,\ i=1,2,\ldots n\}.$

Покажем, что множество D' может быть далее существенно сокращено с помощью трех алгоритмов сокращения последовательностей (4) (или их подпоследовательностей, которые будем обозначать так же).

1. *Первый алгоритм сокращения*.

В последовательности r^j выделяется подпоследовательность первого типа $r_{ju}, u = v, v+1, \ldots, v+w.$ Из r^j вычеркиваются все элементы данной подпоследовательности кроме $r_{ju}, u = v+w.$ Данная процедура повторяется для всех подпоследовательностей первого типа последовательностей r^j.

2. *Второй алгоритм сокращения*.

В последовательности r^j выделяется подпоследовательность второго типа $r_{ju}, u = v, v+1, \ldots, v+w.$ Из r^j вычеркиваются все элементы данной подпоследовательности. Данная процедура повторяется для всех подпоследовательностей второго типа последовательностей r^j.

3. Пусть дана произвольная числовая последовательность C: $c_1, c_2, \ldots, c_k$ и функция действительного аргумента $\psi(x) \geq 0$. Последовательности C соответствует числовая последовательность $\psi(c_1), \psi(c_2), \ldots, \psi(c_k)$, которую будем называть функциональной.

Третий алгоритм сокращения произвольной конечной числовой последовательности C состоит в выделении такой ее подпоследовательности

$$c_{i_1}, c_{i_2}, \ldots, c_{i_h}, \qquad (6)$$

которой соответствует специальная строго убывающая подпоследовательность функциональной последовательности

$$\psi(c_1), \psi(c_2), \ldots, \psi(c_k).$$

Третий алгоритм сокращения.

1. Полагаем $v=w=1, i_1=1$.

2. Шаг $w \geq 1$. Пусть из $c_1, c_2, \ldots, c_w$ выделена подпоследовательность $c_{i_1}, c_{i_2}, \ldots, c_{i_v}$, $v \leq w$. Если $w=k$ подпоследовательность (6) считается построенной и переход на этап 3.

В противном случае w увеличивается на единицу.

Если $\psi(c_{i_v}) > \psi(c_w)$, тогда v увеличивается на единицу, $i_v=w$ и осуществляется переход на начало этапа 2.

3. Конец алгоритма.

Пусть $P_t^{\Omega,\varepsilon}(x)$ - произвольный допустимый предикат такой, что $\varepsilon_j \in r^j$, $\varepsilon_j \in \{r_{ju}, u = v, v+1, \ldots, v+w\} \subseteq r^j$, где r^j - последовательность (4) или ее подпоследовательность, а $r_{ju}, u = v, v+1, \ldots, v+w$ - подпоследовательность первого типа последовательности r^j.

Рассмотрим предикат $P_t^{\Omega,e}(x)$, для которого $e_i = \begin{cases} \varepsilon_i, & i \neq j, \\ r_{j,v+w}, & i = j. \end{cases}$

Ясно, что если предикат $P_t^{\Omega,\varepsilon}(x)$ - допустимый, то будет допустимым также и предикат $P_t^{\Omega,e}(x)$.

Из условий $\rho_j(e_j, S_t, S_i) \le \rho_j(\varepsilon_j, S_t, S_i)$ для $S_i \in K_\lambda$ и $\rho_j(e_j, S_t, S_i) = \rho_j(\varepsilon_j, S_t, S_i)$ для $S_i \notin K_\lambda$ следует, что $f(P_t^{\Omega,e}) \le f(P_t^{\Omega,\varepsilon})$.

Рассмотрим теперь случай, когда $r_{ju}, u = v, v+1, \ldots, v+w$ - подпоследовательность второго типа последовательности r^j.

Рассмотрим предикат $P_t^{\Omega,e}(x)$, для которого $e_i = \begin{cases} \varepsilon_i, & i \ne j, \\ r_{j,v-1}, & i = j. \end{cases}$

Тогда, если предикат $P_t^{\Omega,\varepsilon}(x)$ - допустимый, то будет допустимым также и предикат $P_t^{\Omega,e}(x)$ в силу неравенства $e_j < \varepsilon_j$. Кроме того выполнено $f(P_t^{\Omega,e}) \le f(P_t^{\Omega,\varepsilon})$.

Таким образом, после последовательного применения первого и второго алгоритмов сокращения к последовательностям r^j, $j=1,2,\ldots n$, вычисляется такое подмножество D'' множества D', минимум на котором основного функционала совпадает с минимумом на D'. Обозначим сокращенные последовательности снова как r^j, $D''=r^1 \times r^2 \times \ldots \times r^n$. Применим к последовательностям r^j, $j=1,2,\ldots n$, третий алгоритм сокращения, рассматривая в качестве соответствующих функциональных последовательностей последовательности $\varphi_{tj}(r_{ji}), i = 1,2,\ldots,u_j$.

В итоге получим множество $D^* = \varepsilon^1 \times \varepsilon^2 \times \ldots \times \varepsilon^n$,

где $\varepsilon^j = \{\varepsilon_{ji}, i=1,2,\ldots,k_j\}, \varepsilon_{ju} < \varepsilon_{jv}$, при $u<v$.

В силу свойств третьего алгоритма сокращения для любого допустимого предиката $P_t^{\Omega,\varepsilon}(x)$, $\varepsilon_j \in r^j$, существует $e=(e_1, e_2, \ldots, e_n)$, $e_j = \varepsilon_{jv(j)} \le \varepsilon_j$, для которого $\varphi_{tj}(e_j) \le \varphi_{tj}(\varepsilon_j)$, т.е. $\exists$ допустимый предикат $P_t^{\Omega,e}(x)$, для которого $f(P_t^{\Omega,e}) \le f(P_t^{\Omega,\varepsilon})$.

После явного описания процедуры нахождения множества D^* покажем, что задача поиска логических закономерностей может быть сформулирована в виде специальной задачи целочисленного линейного программирования.

Введем следующие обозначения:

$$c_{ij} = \varphi_{ti}(\varepsilon_{ij}),$$

$$y = (y_{11}, y_{12}, \ldots, y_{1k_1}, y_{21}, y_{22}, \ldots, y_{2k_2} \ldots, y_{n1}, y_{n2}, \ldots, y_{nk_n}),$$

$$y_{ij} \in \{0,1\}, i = 1,2,\ldots,n,\ j = 1,2,\ldots,k_i,$$

$$b_{ij}^u = \rho_i(\varepsilon_{ij}, x_i(S_t), x_i(S_u)),$$

$$\sum_{i=1}^{n} \sum_{j=1}^{k_i} c_{ij} y_{ij} \to \min, \qquad (7)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{k_i} b_{ij}^u y_{ij} \geq 1, u = 1,2,\ldots,m, \quad S_u \notin K_\lambda. \qquad (8)$$

По определению коэффициентов и в силу свойств третьего алгоритма сокращения для задачи (7-8) имеют место следующие свойства коэффициентов:

$$c_{iu} > c_{iv} \geq 0, \text{ при } u < v,\ b_{ij}^u \geq b_{ij}^v \geq 0,\ b_{ij}^u \in \{0,1\}. \qquad (9)$$

В силу свойств коэффициентов (9) задачу (7-8) будем называть задачей целочисленного линейного программирования с блочно-монотонными столбцами (задача БМС-ЦЛП).

Пример задачи (7-8).

i= *1*				*i*= *2*				*i*= *n*					
0.5	0.4	0.3	0.1	0.8	0.6	0.3	…	0.9	0.8	0.6	0.4	0.2	$\leftarrow c_{ij}$
1	1	1	0	1	1	1	…	1	1	1	1	0	
1	0	0	0	1	1	1	…	1	1	1	1	1	
1	1	0	0	1	0	0	…	0	0	0	0	0	
1	1	0	0	1	1	0	…	1	0	0	0	0	$\leftarrow b_{ij}^u$
1	0	0	0	0	0	0	…	0	0	0	0	0	
1	0	0	0	1	0	0	…	1	1	1	0	0	
1	1	1	1	0	0	0	…	0	0	0	0	0	
.	.	.	.	.	.	.	…	.	.	.	.	.	
1	1	1	0	1	1	0	…	1	1	0	0	0	

В силу свойств коэффициентов функции (7) оптимальное решение $y^* = (y^*_{11}, y^*_{12}, \ldots y^*_{1k_1}, y^*_{21}, y^*_{22}, \ldots y^*_{2k_2}, \ldots, y^*_{n1}, y^*_{n2}, \ldots y^*_{nk_n})$ содержит не более одной единичной компоненты для каждой из *n* групп параметров $y^*_{i1}, y^*_{i2}, \ldots y^*_{ik_i}$.

Примечание. В случае $c_{ik_i} > 0$ это проверяется просто. При $c_{ik_i} = 0$, по построению, $k_i = 1$ и данное свойство также выполняется.

Множество единичных компонент однозначно определяет предикат $P_t^{\Omega,\varepsilon}(x)$. Действительно, множество Ω признаков предиката определяется первыми индексами единичных значений компонент y^*_{ij}, а соответствующие значения ε-порогов задаются вторыми индексами $(\varepsilon_i = \varepsilon_{ij})$. Найденные предикаты естественно называть оптимальными окрестностями представительных наборов, поскольку алгоритм находит фрагменты эталонных описаний и их окрестности.

Лекция № 8

3. Распознавание на основе логических закономерностей.

Пусть имеется некоторый алгоритм поиска локально-оптимальных решений задачи (7-8). Найденному множеству локально-оптимальных решений задачи (7-8) соответствует некоторое множество логических закономерностей $\{ P_t^{\Omega,\varepsilon}(x) \}$, связанных с объектом S_t класса K_j и функционалом f. Обозначим через $\mathrm{P}_j = \bigcup_t P_t^{\Omega,\varepsilon}(x)$ множество логических закономерностей класса K_j как объединение множеств предикатов $\{ P_t^{\Omega,\varepsilon}(x) \}$, найденных для всех эталонов класса K_j.

Общая схема, которая используется в алгоритмах распознавания, основанных на голосовании по системам логических закономерностей, включает три последовательных этапа и является частным случаем общей последовательности выполнения этапов в моделях вычисления оценок.

1. Для каждого класса K_j по обучающей информации I_0 согласно описанному ранее алгоритму находится множество логических закономерностей $\mathrm{P_j}$, из которого вычеркиваются предикаты с малыми значениями стандартного функционала качества.

2. Для произвольного распознаваемого объекта S вычисляется "мера близости" $G_j = \sum \beta_t \; P_t^{\Omega,\varepsilon}(S)$ объекта S к классу K_j, где суммирование проводится по всем $P_t^{\Omega,\varepsilon}(x) \in \mathrm{P_j}$. Здесь G_j является "взвешенной суммой голосов" за класс K_j, или, следуя терминологии [2], оценкой S за класс K_j. Нормировочные коэффициенты β_t могут вычисляться различными способами и задают тип процедуры голосования.

3. По значениям G_j вычисляется информационный вектор $(\alpha^A_1(S), \alpha^A_2(S), \ldots, \alpha^A_l(S))$.

Например, $\alpha^A_r(I(S))=1$, если $G_r=\max\{G_j,\ j=1,2,\ldots,l\}$. В противном случае $\alpha^A_r(I(S))=0$. Строка $\alpha^A(S) = (\alpha^A_1(S), \alpha^A_2(S),\ldots, \alpha^A_l(S))$, содержащая ровно одну единицу, означает однозначное решение задачи распознавания - отнесение алгоритмом распознавания A объекта S в один из l классов. Строка $\alpha^A(S)$, содержащая несколько единиц, означает многозначное решение задачи распознавание. В данной ситуации алгоритм распознавания указывает несколько классов, которым может принадлежать объект *S*. Строка $\alpha^A(S)$, содержащая одни нули, интерпретируется как отсутствие классов, на которые похож распознаваемый объект.

4. Нахождение логических закономерностей при обучающих выборках большой длины.

Изложенный в п.2 подход для построения множества логических закономерностей предполагает нахождение предикатов вида (2) для каждого эталонного объекта, т.е. решение задач БМС-ЦЛП (7-8) для каждого эталона. Решение большого числа задач БМС-ЦЛП приводит к существенным затратам процессорного времени для поиска предикатов и памяти для хранения предикатов, времени распознавания новых объектов. Ясно, что найденные множества логических закономерностей будут содержать много похожих предикатов (2), а само их число будет «избыточным». Таким образом представляет интерес вычисление множеств логических закономерностей различной мощности.

Естественным подходом для решения данной задачи является реализация следующей общей схемы.

Пусть задан критерий $\gamma(S_j)$ оценки важности (представительности) произвольного объекта S_j обучающей выборки, $I=\{S_1, S_2,\ldots, S_m\}$, $\eta\geq 1$ – некоторое фиксированное число, $\pi = \{\pi_1,\pi_2,\ldots,\pi_m\}, \pi_i = 0, i = 1,2,\ldots,m.$

Рассмотрим следующий алгоритм вычисления множеств $\mathrm{P_j}$, *j=1,2,…l.*

1. Полагаем $\mathrm{P_j} = \varnothing$, *j=1,2,…,l, v=1.*
2. Пусть $S_{i_v} \in I$ - эталонный объект, $S_{i_v} \in K_\lambda$, для которого выполнено $\gamma(S_{i_v}) = \max\limits_{S_j \in I} \gamma(S_j)$. Решается задача БМС-ЦЛП (7-8) относительно S_{i_v} и находится множество логических

закономерностей $P^{i_v} = \{P_{i_v}^{\Omega,\varepsilon}(x)\}$, для которых S_{i_v} является опорным. Множество P_λ дополняется предикатами из P^{i_v}.

3. Для каждого $i \in I$ величина π_i увеличивается на число предикатов из P^{i_v}, принимающих единичные значения на объекте S_i.
4. Из I исключается S_{i_v} и все S_i, для которых $\pi_i \geq \eta$.

Если $I=\varnothing$, множества P_j, $j=1,2,...l$, считаются построенными и алгоритм заканчивает свою работу.

В противном случае значение v увеличивается на единицу и осуществляется переход на второй этап.

При выборе $\eta=1$ множество P_j содержит множество предикатов (2), причем для каждого эталонного объекта из K_j имеется хотя бы один предикат, принимающий на данном объекте значение 1. При выборе $\eta=m$ множество P_j является, как правило, объединением всех предикатов (2), вычисленных для всех задач БМС-ЦЛП с использованием всех эталонов K_j в качестве опорных.

Рассмотрим теперь вопрос выбора функции $\gamma(S_p)$, оценивающей «представительность» объекта S_p.

Здесь можно использовать различные подходы, например, следующий. Выбирается некоторая метрика (полуметрика) $\rho(S_i, S_j)$ в пространстве признаковых описаний и при некотором фиксированном значении k представительность $\gamma(S_j)$ объекта $S_j \in K_\lambda$ определяется как число объектов данного класса из ближайших k соседей , т.е. если $J(S_j)$ - множество ближайших k соседей к S_j, то $\gamma(S_j) = \left|\{S_i \in K_\lambda \cap J(S_j)\}\right|$.

5. Оптимизация логических закономерностей.

Пусть по данным обучающей выборки с использованием эвристического функционала (3) вычислены некоторые множества логических закономерностей $\mathrm{P}_j = \bigcup_t P_t^{\Omega,\varepsilon}(x)$ по всем классам K_j, $j=1,2,...,l$.

Полученные множества "элементарных знаний" содержат большое число элементов, которые используются как в процедурах голосования при классификации новых объектов, так имеют и самостоятельную ценность для понимания и описания классов. Независимо от метода их поиска наиболее естественным критерием их оценки является (понятный и наглядный) стандартный функционал качества предикатов.

В самом начале сложная задача поиска логических закономерностей (1) со стандартным критерием их оценки была заменена более простой. Множества «элементарных знаний» $\mathrm{P}_j = \bigcup_t P_t^{\Omega,\varepsilon}(x)$ каждого класса вычислялись при двух существенных ограничениях: поиск логических закономерностей осуществляется лишь на множестве предикатов с опорными эталонами и с использованием эвристического функционала качества. При этом, конкретный используемый метод решения задачи БМС-ЦЛП не гарантирует вычисление всех локально-оптимальных экстремумов (тем более, на длинных выборках) или нахождение глобального экстремума. В итоге, найденные множества логических закономерностей содержат близкие или даже равные элементы. С другой стороны, существуют логические закономерности, имеющие более высокие значения стандартного критерия качества, но данные «элементарные знания» найти не удалось в силу ограниченности процедур поиска, использования более простых функционалов и множества предикатов частного вида. Таким образом, возникает вопрос о поиске множеств P_j^* логических закономерностей с более высокими значениями стандартного функционала качества чем у найденных из P_j, используя множества P_j в качестве начальных приближений и естественные гипотезы о связи "близких" знаний. Если данная задача решается положительно, то «элементарные знания» из P_j^* могут рассматриваться как обобщения множеств «элементарных знаний» P_j.

Пусть $P_t(x) \in \mathrm{P}_j$.

Обозначим через $\psi(P_t) = (\psi_l(P_t), \psi_l(P_t), ..., \psi_l(P_t))$ - оценочный вектор предиката $P_t(x)$, где $\psi_j(P_t) = \left|\{S_i : \cdot S_i \in K_j, P_t(S_i) = 1\}\right| / \left|K_j\right|$.

Предикат $P_t(x)$ из множества P_j назовем λ - допустимым, если $\psi_i(P_t) \le \lambda\psi_j(P_t), i \ne j,,\ \lambda \ge 0.$ Далее предполагаем, что множества P_j содержат лишь λ-допустимые предикаты, а λ для краткости будем опускать.

Пусть $P_1(x)$, $P_2(x)$ - пара допустимых предикатов из P_j,

$$P_1(x) = \mathop{\&}_{j\in\Omega_1}(a_j^1 \le x_j \le b_j^1) \mathop{\&}_{j\in\Omega}(c_j^1 \le x_j \le d_j^1),$$

$$P_2(x) = \mathop{\&}_{j\in\Omega_2}(a_j^2 \le x_j \le b_j^2) \mathop{\&}_{j\in\Omega}(c_j^2 \le x_j \le d_j^2),$$ где

$$\Omega_1, \Omega_2, \Omega \in \{1,2,...,n\}, \cdot\ \Omega_1 \cap \Omega_2 = \varnothing,\ \Omega_1 \cap \Omega = \varnothing,\ \Omega_2 \cap \Omega = \varnothing.$$

Допустимый предикат $P(x) = \underset{j\in\Omega_1}{\&}(a_j^1 \le x_j \le b_j^1) \underset{j\in\Omega}{\&}(\sigma_j^1 \le x_j \le \delta_j^1)$ назовем расширением предиката $P_1(x)$ по предикату $P_2(x)$, если $\sigma_j^1 \in \{-\infty, \cdot c_j^1, \cdot \min\{c_j^1, c_j^2\}\}$, $\delta_j^1 \in \{d_j^1, \cdot \max\{d_j^1, d_j^2\}, \cdot +\infty\}$. В случаях, когда $\sigma_j^1 = -\infty$, $\delta_j^1 = +\infty$, переменная x_j становится фиктивной. Множество предикатов P_j^2 назовем расширением множества предикатов P_j^1, если оно состоит из всех расширений предикатов множества P_j^1 по предикатам этого же множества.

Допустимый предикат $P(x)$ назовем максимальным по множеству P_j, если не существует его расширений ни по одному предикату из P_j.

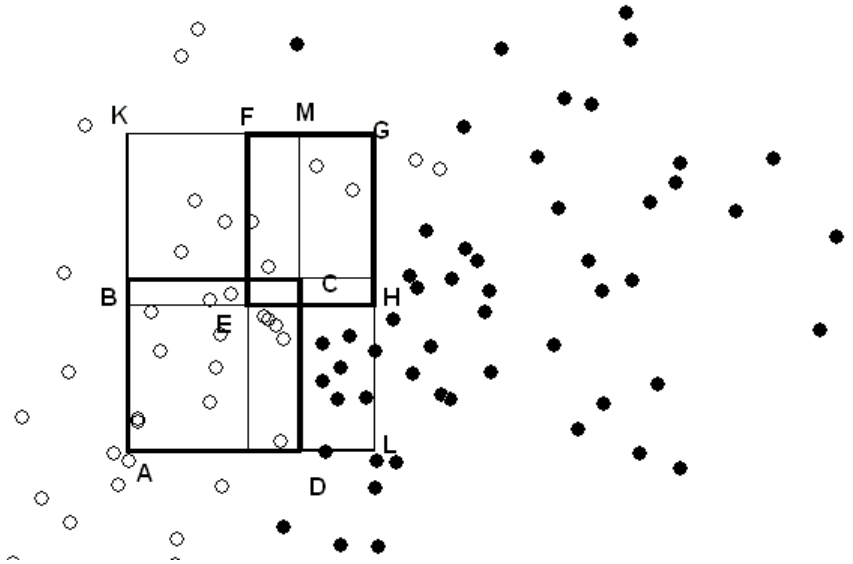

Рис.2. ABCD и **EFGH** соответствуют элементарным предикатам. **AKMD** и **GHBK** их максимальные допустимые обобщения при λ=0. Обобщение **AKGL** не является допустимым.

В результате построения последовательных расширений $\mathrm{P}_j^1, \mathrm{P}_j^2, \ldots, \mathrm{P}_j^i, \ldots$, будут построены множества P_j^*, состоящие только из максимальных расширений. Конечность настоящего процесса очевидна. В случае $\lambda=0$ по исходным множествам логических закономерностей классов будут построены множества логических закономерностей классов, обладающие максимально возможными значениями стандартного функционала качества среди всевозможных допустимых расширений начальных P_j.

Построенные множества максимальных предикатов P_j^* обладают рядом благоприятных свойств, которые делают их более предпочтительными относительно исходных P_j. Сам процесс построения по множествам P_j (исходным "элементарным знаниям") множеств P_j^* естественно называть оптимизацией исходных элементарных знаний. Максимальные предикаты из P_j^* имеют более высокие показатели для $\varphi_j(P_t)$ чем их "предки", причем значения $\varphi_i(P_t)$, $i\#j$, не превосходят установленных ограничений сверху.

ЛЕКЦИЯ № 9

6. Нахождение логических закономерностей, оптимальных по стандартному критерию качества (дискретный подход).

Вернемся к общей задаче поиска логических закономерностей. Пусть задано множество элементарных предикатов, параметрически зависящих от числовых неизвестных $c_j^1, c_j^2, j = 1,2,...,n$:

$$P_j(c_j^1, c_j^2, x_j) = \begin{cases} 1, & c_j^1 \le x_j \le c_j^2, \\ 0, & \text{иначе.} \end{cases}$$

и $\Omega \subseteq \{1,2,...,n\}$. Будем рассматривать более общее определение логической закономерности, чем ранее введенное. Пусть $\zeta \ge 0$ некоторый параметр.

Определение. Предикат $P^{\Omega}(c^1, c^2, x) = \underset{j\in\Omega}{\&} P_j(c_j^1, c_j^2, x_j)$ (1)

называется логической закономерностью класса K_λ, если

4. $\exists\, S_t \in K_\lambda :\ P^{\Omega}(c^1, c^2, S_t) = 1,$
5. $\dfrac{\left|\{S_i \notin K_\lambda \mid P(S_i) = 1\}\right|}{\left|\{P(S_i) = 1\}\right|} \le \varsigma$ (доля объектов $P(S_i) = 1$ чужих классов не превышает заданный порог ζ).
6. $\varphi(P^{\Omega}(c^1, c^2, x)) = \underset{\{P^{O}(d^1, d^2, x)\}}{extr} \varphi(P^{O}(d^1, d^2, x))$, где

$\varphi(P^{\Omega}(c^1, c^2, S)) = \left|\{S_i : S_i \in K_\lambda, P^{\Omega}(c^1, c^2, S_i) = 1\}\right|$.

В настоящем разделе будет рассмотрен метод прямого поиска логических закономерностей классов при ζ=0. Рассмотрим данную задачу

при ограничении $P^{\Omega}(c^1, c^2, S_t) = 1,$ где $S_t \in K_\lambda$ - произвольный фиксированный объект.

Пусть $D_i^1 = \{d_{i1}^1, d_{i2}^1, ..., d_{iu}^1\}$, $D_i^2 = \{d_{i1}^2, d_{i2}^2, ..., d_{iv}^2\}$ - некоторые варианты выбора возможных значений левых и правых границ предикатов (1), соответственно, $-\infty = d_{iu}^1 < ... < d_{i2}^1 < d_{i1}^1 < a_{ti} < d_{i1}^2 < d_{i2}^2 < ... < d_{iv}^2 = \infty$. Для простоты записи, мы рассмотрим случай, когда значения u и v равны для всех признаков i, а также будем считать, что первые по порядку объекты обучающей выборки и принадлежат рассматриваемому классу $K_\lambda = \{S_1, S_2, ...S_{m_\lambda}\}$.

Построим числовую матрицу $\mathbf{M} = \begin{pmatrix} \mathbf{B}_1^1\mathbf{B}_1^2\mathbf{B}_2^1\mathbf{B}_2^2...\mathbf{B}_n^1\mathbf{B}_n^2 \\ \mathbf{C}_1^1\mathbf{C}_1^2\mathbf{C}_2^1\mathbf{C}_2^2...\mathbf{C}_n^1\mathbf{C}_n^2 \end{pmatrix}_{m \times N}$,

$N = n \times (u + v)$,

$\mathbf{B}_i^1 = (b_{ij}^{1q})_{m_\lambda \times u}$, $q = 1,2,...,m_\lambda$, $i = 1,2,...,n$, $j = 1,2,...,u$,

$\mathbf{B}_i^2 = (b_{ij}^{2q})_{m_\lambda \times v}$, $q = 1,2,...,m_\lambda$, $i = 1,2,...,n$, $j = 1,2,...,v$,

$\mathbf{C}_i^1 = (c_{ij}^{1q})_{(m-m_\lambda) \times u}$, $q = 1,2,...,m - m_\lambda$, $i = 1,2,...,n$, $j = 1,2,...,u$,

$\mathbf{C}_i^2 = (c_{ij}^{2q})_{(m-m_\lambda) \times v}$, $q = 1,2,...,m - m_\lambda$, $i = 1,2,...,n$, $j = 1,2,...,v$, where

$$b_{ij}^{1q} = \begin{cases} 1, & x_i(S_q) \geq d_{ij}^1, \\ 0, & \text{иначе}, \end{cases} \quad b_{ij}^{2q} = \begin{cases} 1, & x_i(S_q) \leq d_{ij}^2, \\ 0, & \text{иначе}, \end{cases} \tag{2}$$

$$c_{ij}^{1q} = \begin{cases} 1, & x_i(S_{m_\lambda + q}) < d_{ij}^1, \\ 0, & \text{иначе}, \end{cases} \quad c_{ij}^{2q} = \begin{cases} 1, & x_i(S_{m_\lambda + q}) > d_{ij}^2, \\ 0, & \text{иначе}. \end{cases} \tag{3}$$

Как следует из (2), (3), выполняются следующие ограничения для элементов $\mathbf{M}$:

$$b_{ij}^{1q} \leq b_{i,j+1}^{1q}, \; b_{ij}^{2q} \leq b_{i,j+1}^{2q}, \tag{4}$$

$$c_{ij}^{1q} \geq c_{i,j+1}^{1q}, \; c_{ij}^{2q} \geq c_{i,j+1}^{2q}. \tag{5}$$

Таким образом, матрица $\mathbf{M}$ имеет структуру, как показано в Таблице 1.

Таблица 1. Иллюстрация структуры матрицы коэффициентов

i=1	i=2	i=3	i=4	i=5	i=6	…	i=n-1	i=n
0000011	1111111	0111111	1111111	0000001	…	0000001	1111111
0011111	1111111	1111111	0001111	0000001	1111111	…	0011111	1111111
1111111	0001111	0000001	1111111	0001111	1111111	…	1111111	0000111
1111111	0000011	1111111	0000011	0011111	1111111	…	1111111	0011111
.	.	.	.	.	.	.	.	.
1110000	0000000	0000000	1111100	1100000	0000000	…	0000000	1000000
1000000	0000000	1110000	0000000	1110000	0000000	…	1000000	0000000
1111000	0000000	1000000	0000000	1111000	0000000	…	1100000	0000000
0000000	1100000	0000000	1100000	0000000	1100000	…	0000000	1111000
.	.	.	.	.	.	.	.	.

Множество всех предикатов (1) с возможными границами D_i^1, D_i^2 находится во взаимнооднозначном соответствии с множеством векторов $\{ < x_{ij}^1, x_{ij}^2 > \}$,

$$< x_{ij}^1, x_{ij}^2 > =$$
$$< x_{11}^1, x_{12}^1, ..., x_{1u}^1, x_{11}^2, x_{12}^2, ..., x_{1v}^2, x_{21}^1, x_{22}^1, ..., x_{2u}^1, x_{21}^2, x_{22}^2, ..., x_{2v}^2, ..., x_{n1}^1, x_{n2}^1, ..., x_{nu}^1, x_{n1}^2, x_{n2}^2, ..., x_{nv}^2 >$$

при ограничениях $x_{ij}^1, x_{ij}^2 \in \{0,1\}, \sum_{j=1}^{u} x_{ij}^1 = 1, \sum_{j=1}^{v} x_{ij}^2 = 1,\ i=1,2,...,n.$

В виду данного соответствия, мы будем использовать также запись $\varphi(< x_{ij}^1, x_{ij}^2 >)$ для стандартного критерия оптимальности.

Единицы в $\{ < x_{ij}^1, x_{ij}^2 > \}$ соответствуют выбору значений параметров $c_j^1, c_j^2, j = 1,2,...,n$.

Для выполнения условия 2) должны выполняться неравенства (6).

$$f_q^c(< x_{ij}^1, x_{ij}^2 >) = \sum_{i=1}^{n} (\sum_{j=1}^{u} c_{ij}^{1q} x_{ij}^1 + \sum_{j=1}^{v} c_{ij}^{2q} x_{ij}^2) \geq 1, q = 1,2,..., m - m_\lambda \qquad (6)$$

Наконец, стандартный критерий оптимальности предиката $\varphi(< x_{ij}^1, x_{ij}^2 >)$ равен числу выполненных равенств в системе (7) (при соответствующих значениях параметров $c_j^1, c_j^2, j = 1,2,...,n$).

$$f_q^b(< x_{ij}^1, x_{ij}^2 >) = \sum_{i=1}^{n} (\sum_{j=1}^{u} (b_{ij}^{1q} - 1) x_{ij}^1 + \sum_{j=1}^{v} (b_{ij}^{2q} - 1) x_{ij}^2) = 0, q = 1,2,..., m. \qquad (7)$$

Таким образом, проблема поиска оптимального предиката (1) может быть сформулирована как специальная дискретная оптимизационная задача.:

Задача Z:

$\varphi(< x_{ij}^1, x_{ij}^2 >)$=<число выполненных уравнений в (7)> $\rightarrow \max$,

при ограничениях (8-9)

$$\sum_{i=1}^{n} (\sum_{j=1}^{u} c_{ij}^{1q} x_{ij}^1 + \sum_{j=1}^{v} c_{ij}^{2q} x_{ij}^2) \geq 1, q = 1,2,..., m - m_\lambda . \qquad (8)$$

$$x_{ij}^1, x_{ij}^2 \in \{0,1\}, \sum_{j=1}^{u} x_{ij}^1 = 1, \sum_{j=1}^{v} x_{ij}^2 = 1,\ i=1,2,...,n, \qquad (9)$$

Поставим в соответствие задаче (7-9) аналогичную задачу (10-13) относительно вещественных переменных.

Задача ZC:

<число выполненных неравенств в (10)> → max,

при ограничениях (11-13)

$$\sum_{i=1}^{n}(\sum_{j=1}^{u}(b_{ij}^{1q}-1)x_{ij}^{1}+\sum_{j=1}^{v}(b_{ij}^{2q}-1)x_{ij}^{2})\geq 0, q=1,2,...,m,. \quad (10)$$

$$\sum_{i=1}^{n}(\sum_{j=1}^{u}c_{ij}^{1q}x_{ij}^{1}+\sum_{j=1}^{v}c_{ij}^{2q}x_{ij}^{2})\geq 1, q=1,2,...,m-m_{\lambda}. \quad (11)$$

$x_{ij}^{1}\geq 0$, $i=1,2,...,n,\ j=1,2,...,u,$ (12)

$x_{ij}^{2}\geq 0$, $i=1,2,...,n, j=1,2,...,v.$ (13)

Пусть $Q=\{q:\sum_{i=1}^{n}(\sum_{j=1}^{u}(b_{ij}^{1q}-1)x_{ij}^{\circ 1}+\sum_{j=1}^{v}(b_{ij}^{2q}-1)x_{ij}^{\circ 2})=0, q=1,2,...,m\}$

, где $< x_{ij}^{\circ 1}, x_{ij}^{\circ 2} >$ является некоторым решением задачи (10-13).

Пусть номер признака $i=1,2,...,n$ фиксирован, и $p(i)=\min\{j: b_{ij}^{1q}=1, \forall q\in Q\}$, $r(i)=\min\{j: b_{ij}^{2q}=1, \forall q\in Q\}$. Пусть $x_{ip}^{*1}=1, x_{ij}^{*1}=0, j\neq p,$ $x_{ir}^{*2}=1, x_{ij}^{*2}=0, j\neq r.$ После выполнения аналогичных операций для $i=1,2,...,n,$ будет определен вектор $< x_{ij}^{*1}, x_{ij}^{*2} >$.

ЛЕКЦИЯ № 10

Теорема. Вектор $< x_{ij}^{*1}, x_{ij}^{*2} >$ является решением Задачи **Z.**

Доказательство.

$\varphi(< x_{ij}^{o1}, x_{ij}^{o2} >)=\varphi(< x_{ij}^{*1}, x_{ij}^{*2} >)$, а вектор $< x_{ij}^{*1}, x_{ij}^{*2} >$ удовлетворяет ограничениям (9) по построению. Покажем справедливость (11) для $< x_{ij}^{*1}, x_{ij}^{*2} >$.

При любом i, $x_{ij}^{\circ 1}=0$ при $j<p(i)$. Действительно, если $\exists j<p(i)$: $x_{ij}^{\circ 1}>0$, тогда $\exists q\in Q$: $b_{ij}^{1q}=0$ и $f_q^b(< x_{ij}^{o1}, x_{ij}^{o2} >)<0$, что противоречит условию $q\in Q$. Аналогично, $x_{ij}^{\circ 2}=0$ при $j<r(i)$.

Тогда $\sum_{j=1}^{u}c_{ij}^{1q}x_{ij}^{o1}+\sum_{j=1}^{v}c_{ij}^{2q}x_{ij}^{o2}=\sum_{j=p(i)}^{u}c_{ij}^{1q}x_{ij}^{o1}+\sum_{j=r(i)}^{v}c_{ij}^{2q}x_{ij}^{o2}$ (p,r зависят от i).

В силу (5), $\forall q = 1,2,...,m - m_\lambda$, $\exists i$, т.ч. $c_{ip(i)}^{1q} \vee c_{ir(i)}^{2q} = 1$.

Окончательно,

$$f_q^c(< x_{ij}^{*1}, x_{ij}^{*2} >) = \sum_{i=1}^{n}(\sum_{j=1}^{u} c_{ij}^{1q} x_{ij}^{*1} + \sum_{j=1}^{v} c_{ij}^{2q} x_{ij}^{*2}) \geq c_{ip(i)}^{1q} x_{ip(i)}^{*1} + c_{ir(i)}^{2q} x_{ir(i)}^{*2} \geq 1, q = 1,2,...,m - m_\lambda$$

, т.е. (11) выполнено.

Данная теорема дает основу для создания алгоритма поиска предикатов (1) для каждого класса:

1. Вычисление опорного объекта.
2. Вычисление множеств D^1, D^2 .
3. Решение Проблемы **ZC,** и нахождение решения $\{Q, < x_{ij}^{*1}, x_{ij}^{*2} >\}$ проблемы **Z**.

Замечания.

1. Для нахождения множества логических закономерностей некоторого класса выбирается произвольный эталон класса в качестве опорного эталонов. Находятся логические закономерности, соответствующие локально-оптимальным решениям Проблемы **Z,** связанной с конкретным опорным элементом. Объединение всех полученных логических закономерностей образует конечное множество логических закономерностей класса. Далее выбирается новый опорный элемент класса из эталонов, оставшихся «непокрытыми» найденными ранее предикатами класса. Для нового опорного элемента аналогично находится множество связанных с ним логических закономерностей, которые пополняют множество найденных ранее логических закономерностей класса. Процесс продолжается до «покрытия» найденными закономерностями всех эталонов класса.
2. Возможно вычисление “полных множеств” D^1, D^2, которые включают значения параметров оптимальных предикатов (1). Доказательство аналогично рассмотренному в [5] для некоторого эвристического критерия оптимальности логических закономерностей. В случае практически больших размерностей множеств D^1, D^2 , некоторые приближения могут быть использованы для сокращения размерностей.
3. Для нахождения максимальной совместной подсистемы системы линейных неравенств практически успешно использовался итеративный практический алгоритм [6].

Рассмотренный алгоритм является вполне универсальным но трудоемким в случаях «плохой отделимости» класса. Под плохой отделимостью класса здесь будем понимать наличие значительного числа объектов данного класса, находящихся в окружении объектов других классов (хотя доля данных объектов класса относительно общего числа объектов класса может быть мала). В данном случае приходится решать большое число задач **Z**, но результаты, полученные при

использовании в качестве опорных элементов объектов-выбросов, будут отрицательны. Полученные закономерности будут иметь низкое значение критерия качества. Сложность основной задачи существенно возрастает с ростом длины обучающей выборки. Эффективное решение для выборок большой длины возможно при следующей модификации задачи **Z** .

Поставим задачу поиска логической закономерности $P(S)$ класса, для которой $\varphi(P) \geq h$, где h >0 – некоторый параметр (качество предиката будем оценивать долей объектов своего класса, на которых он равен 1). Т.е. априори предполагается, что логические закономерности данного качества существуют. Пусть g – параметр (0<g<1).

Опишем алгоритм поиска логических закономерностей класса, результат которого формулируется следующим образом: «В вычисленном множестве логических закономерностей $\mathbf{P_j}=\{P(S)\}$ класса K_j с вероятностью не менее g имеется логическая закономерность $P(S)$, для которой $\varphi(P) \geq h$».

Пусть задача поиска логических закономерностей класса решается относительно k случайно выбранных «опорных» эталонов класса K_j , а все найденные логические закономерности объединяются в одно множество $\mathbf{P_j}$. Тогда вероятность того, что искомый предикат не будет найден оценивается сверху как $(1-h)^k$. Тогда значение параметра k определяется из соотношения

$$(1-h)^k \leq (1-g). \qquad (14)$$

Значение параметра k является важным фактором эффективности алгоритма. Например, при g=0.9 и h=0.1 из (14) следует $k \geq 22$. Значение $k=22$ вполне приемлемо для задач большой размерности.

В заключение, рассмотрим вопрос выбора значения параметра *h*. Как его оценить для заданной конкретной задачи, какое его значение является обоснованным с каких-либо позиций?

Найденные логические закономерности класса K_j могут быть «статистически значимыми» или нет. Для этого используется подход, именуемый «перестановочный тест». Выполняется серия из следующих *t* однотипных расчетов (*t* – параметр «количество случайных перестановок»). Осуществляется случайная перестановка строк таблицы обучения, при этом , как и ранее, первые m_1 строк новой таблицы $\widetilde{T}_{nml}$ считаются эталонами первого класса, следующие по порядку $(m_2 - m_1)$ строк - эталонами второго класса, и т.д. (т.е. проводится случайное изменение номеров классов эталонных объектов с сохранением общего числа эталонов класса). Для класса $\widetilde{K}_j$ таблицы $\widetilde{T}_{nml}$ находится наилучшая закономерность $\widetilde{P}_j$ с качеством $\varphi(\widetilde{P}_j)$. Тогда логическая закономерность P_q из множества $\mathbf{P_j} = \{ P(S) \}$ считается статистически значимой, если неравенства $\varphi(P_q) \geq \varphi(\widetilde{P}_j)$ выполнены не менее чем 100**q*%, (0<q<1* – уровень значимости) в серии из *t* испытаний. Таким образом, значения величин $\varphi(P_j)$, полученных в результате перестановочных тестов, являются естественным вариантом выбора нижней границы для параметра *h.*

6. Нахождение логических закономерностей, оптимальных по стандартному критерию качества (генетический алгоритм).

В настоящем разделе рассматривается следующий подход. Определяется некоторое множество предикатов вида (1), на котором существует наилучшая логическая закономерность рассматриваемого класса. Каждому предикату взаимнооднозначно соответствует бинарный вектор, длина которого равна числу эталонов класса. Поскольку допустимость

предикатов и значения их критерия качества вычисляются просто, задача нахождения наилучшей закономерности эффективно решается с помощью генетического (эволюционного) подхода для заданного критерия качества закономерностей относительно введенного множества бинарных кодов.

6.1. Основные элементы генетических алгоритмов – эволюционных методов поиска.

Основной задачей является нахождение оптимального по заданному критерию объекта в некотором, вообще говоря бесконечном, множестве объектов. Сама процедура поиска основана на аналогии и моделировании процессов селекции в природе, когда происходит выживание и появление объектов с более высокими показателями, за счет отбора, скрещивания и мутации объектов. Ниже в круглых скобках приводятся наименования терминов, принятые в генетике.

Имеется множество объектов $\widetilde{O} = \{O\}$, на котором задана **функция цели** $f : O \Rightarrow R$.

Определяется конечное множество $\widetilde{\Sigma} = \{\Sigma\}$ (**пространство представлений**) и **функция кодирования** $e : O \Rightarrow \Sigma$. Элементы пространства представлений называются **представлениями (генотипами, хромосомами).** Функция, обратная к функции кодирования, называется **функцией декодирования** и обозначается как $e^{-1} : \Sigma \Rightarrow O$. Декодирование может быть неоднозначным и одному представлению может соответствовать множество объектов, имеющих одно и то же представление. Если для некоторого представления не существует элемента $e^{-1}(\Sigma)$, данное представление считается недопустимым.

Считается заданной также функция оценки представлений (приспособленность) $\mu : \Sigma \Rightarrow R$. Обычно полагают $\mu(\Sigma) = f(e^{-1}(\Sigma))$. Набор $\{O, \Sigma, \mu\}$ называется особью. Совокупность особей называется популяцией.

Основная задача состоит в нахождении $S^* = \arg\max_{\Sigma \in \tilde{\Sigma}} \mu(\Sigma)$.

Таким образом, предварительным этапом решения основной задачи является создание символьной модели, элементами которой являются:

- пространство решений $\tilde{O} = \{O\}$;

- пространство представлений $\tilde{\Sigma} = \{\Sigma\}$;

- функции кодирования и декодирования;

- функция цели;

- функция оценки представлений.

Опишем общую схему генетического алгоритма.

Пусть задана символьная модель задачи, где пространство представлений есть дискретный единичный куб размерности N, а представления являются строками длины N.

Генетический алгоритм начинается со случайной генерации совокупности особей – популяции. Далее организуется итеративный процесс, который продолжается до выполнения какого-либо критерия остановки (достижение максимального числа итераций, требуемого значения функции цели, и т.п.). На каждой итерации (поколении) осуществляются операторы отбора особей (случайно, но пропорционально значениям функции приспособленности), и операторы порождения новых особей – кроссовер и мутация. Рассмотрим их подробнее.

Пусть имеется некоторая популяция из m особей (на первой итерации заданная случайно). Считаем, что число особей четное. Популяция случайно разбивается на $m/2$ пар особей (родителей).

1. Оператор **кроссовер**.

Выбирается некоторая пара особей и к ней с вероятностью p_c применяется следующий оператор. Пусть $\Sigma_1 = (\sigma_{11}, \sigma_{12}, ..., \sigma_{1N})$ и $\Sigma_2 = (\sigma_{21}, \sigma_{22}, ..., \sigma_{2N})$ - коды родителей.

Случайно выбирается натуральное $l < N$ и создаются два новых кода (определяющие новые особи - потомки) :

$\Sigma_1^* = (\sigma_{11}, \sigma_{12}, ..., \sigma_{1l}, \sigma_{2,l+1}, \sigma_{2,l+2}, ..., \sigma_{2N})$,

$\Sigma_2^* = (\sigma_{21}, \sigma_{22}, ..., \sigma_{2l}, \sigma_{1,l+1}, \sigma_{1,l+2}, ..., \sigma_{1N})$.

Пример.

Родители	Потомки
110110|0000110	110110|1110000
101010|1110000	101010|0000110

Данный способ «разрыва и склеивания» называется одноточечным кроссовером. Имеются и другие подходы. Например, в двухточечном кроссовере происходит между родителями «обмен» некоторых центральных фрагментов, имеющих одинаковые номера генов. Полученные потомки включаются в имеющуюся популяцию.

2. Оператор **мутации.**

Во всех генотипах популяции с вероятностью p_m осуществляется замена значений всех генов на противоположные (1 на 0, 0 на 1).

3. Оператор **отбора.**

Пусть в результате применения операторов кроссовера и мутации получено m потомков, а общее число особей данного поколения составило $2m$. Следующее поколение из m особей формируется в результате случайного выбора m особей из имеющихся $2m$ особей, причем вероятность попадания в новую популяцию некоторого Σ_i из имеющегося множества $\Sigma_1, \Sigma_2, ..., \Sigma_{2m}$ равна $\mu(\Sigma_i) / \sum_{j=1}^{2m} \mu(\Sigma_j)$.

6.2. Генетический алгоритм поиска логических закономерностей.

Построим символьную модель генетического алгоритма поиска логических закономерностей некоторого класса K_λ. Для простоты, обозначим эталоны данного класса как $S_1, S_2, ..., S_N$.

Пусть $\widetilde{S} = \{S_{i_1}, S_{i_2}, ..., S_{i_k}\}$ - подмножество эталонов класса K_λ.

Поставим ему в соответствие предикат $P(c^1, c^2, x) = \& P_j(c_j^1, c_j^2, x_j)$,

где $c_j^1 = \min\limits_{S_i \in \widetilde{S}} \{x_j(S_i)\}$, $c_j^2 = \max\limits_{S_i \in \widetilde{S}} \{x_j(S_i)\}$. (15)

Если некоторый предикат (1) является логической закономерностью, то существует эквивалентный ему предикат (т.е. условия 1-3 предикатов полностью совпадают), для которого константы $c_j^1, c_j^2, j = 1,2,...,n$ определяются согласно (15). Следовательно, **пространство решений** можно ограничить множеством предикатов (1), которые определяются всевозможными подмножествами $\widetilde{S}$ с вычислением констант $c_j^1, c_j^2, j = 1,2,...,n$, согласно (15).

Пространство представлений определим как множество бинарных векторов $\Sigma_i = (\sigma_{i1}, \sigma_{i2}, ..., \sigma_{iN})$, где N – число эталонов класса K_λ, а

$$\sigma_{ij} = \begin{cases} 1, & S_j \in \widetilde{S}, \\ 0, & S_j \notin \widetilde{S}. \end{cases}$$

Функции кодирования и декодирования описаны выше. Очевидно, имеется взаимнооднозначное соответствие между пространством решений и пространством представлений.

Функцией цели для нас является стандартный функционал качества предикатов.

Функция оценки представлений определяется следующим образом. Пусть представление $\Sigma_i = (\sigma_{i1}, \sigma_{i2}, ..., \sigma_{iN})$ соответствует подмножеству $\widetilde{S} = \{S_{i_1}, S_{i_2}, ..., S_{i_k}\}$, которому соответствует также предикат $P_i(c^1, c^2, x)$.

Положим

$$\mu(\Sigma_i) = \begin{cases} \varphi(P_i(c^1, c^2, S_j)), & P_i(c^1, c^2, S_j) = 0, \forall S_j \notin K_\lambda, \\ \varphi(P_i(c^1, c^2, S_j)/(1+\psi)), & \text{иначе.} \end{cases}$$

Здесь $\psi = \left|\{S_j \notin K_\lambda, P_i(c^1, c^2, S_j) = 1\}\right|$.

Генетический алгоритм строится согласно общей схеме с двухточечным кроссовером.

Строится начальная популяция хромосом, «покрывающих» все эталоны рассматриваемого класса. Размер популяции m может быть различным в зависимости от типа решаемой задачи. В данной задаче принималось $m = N/2$. Начальный вид хромосом также может быть различным. Выбирается фиксированный номер хромосомы, которая будет являться «лидером» популяции. Лидер популяции – это хромосома, имеющая наибольшее значение целевой функции. Лидер популяции не может подвергаться никаким изменениям и операциям, кроме замены на хромосому с большим значением целевой функции.

При решении данной задачи оптимальное значение вероятности мутации находится в пределах $p_m \in \left[\frac{1}{N}; \frac{3}{N}\right]$, где N- длина хромосомы.

Экспериментальные вычисления на таблицах обучения различных задач распознавания показали, что при длине хромосомы L порядка нескольких сотен алгоритм работает надежно и находит оптимальный предикат за 50-100 итераций.

ЛЕКЦИЯ № 11

Ранее были рассмотрены алгоритмы поиска логических закономерностей класса K_λ, т.е. предикатов $P^{\Omega}(c^1, c^2, x) = \underset{j\in\Omega}{\&} P_j(c_j^1, c_j^2, x_j)$, удовлетворяющих специальным ограничениям, причем по построению $\Omega = \{1,2,...,n\}$. Рассмотрим вопрос поиска логических закономерностей с минимальными по мощности множествами Ω.

Рассмотрим следующую задачу целочисленного линейного программирования.

$$\sum_{j=1}^{n} y_j \to \min,$$

$$\sum_{j=1}^{n} \beta_{ij} y_j \geq 1, \text{ для всех } S_i \notin K_\lambda, \qquad (1)$$

$$y_j \in \{0,1\}.$$

Здесь $\beta_{ij} = 1 - P_j(c_j^1, c_j^2, a_{ij})$. Множество всех единичных компонент решения $(y_1, y_2, ..., y_n)$ определяет соответствующее подмножество признаков Ω. Действительно, полученный предикат $P^{\Omega}(c^1, c^2, x) = \underset{j\in\Omega}{\&} P_j(c_j^1, c_j^2, x_j)$ удовлетворяет условиям 1, 3 определения логической закономерности, а выполнение линейных ограничений в (1) соответствует выполнению второго условия основного определения. Множество всех единиц построенного предиката является аналогом максимального интервала для булевского случая.

6. Обобщение знаний по выборкам прецедентов

В настоящем разделе будет рассмотрена задача обобщения знаний (множества логических закономерностей), под которой понимается нахождение и вычисление обобщенных характеристик признаков и классов на основе ранее полученных множеств логических закономерностей классов.

Пусть для класса K_λ вычислено множество логических закономерностей $P_t^{\Omega_t}(x), t \in T$.

Определение. Логическим описанием класса K_λ назовем логическую сумму $D_\lambda(x) = \underset{t\in T}{\vee} P_t^{\Omega_t}(x)$.

Ясно, что $D_\lambda(S_t) = 1$ для всех обучающих объектов из класса K_λ (если обучающая выборка не противоречива), и $D_\lambda(S_t) = 0$ для всех эталонных объектов, не принадлежащих классу K_λ. Таким образом, $D_\lambda(x)$ совпадает на множестве описаний эталонных объектов с характеристической функцией класса K_λ.

Очевидно, $D_\lambda(x)$ является прямым аналогом сокращенных ДНФ для булевых функций. Тогда естественно рассмотреть и аналоги минимальных и кратчайших ДНФ.

Определение. Минимальным логическим описанием класса K_λ назовем логическую сумму $D_\lambda^m(x) = \bigvee_{t \in T' \subseteq T} P_t^{\Omega_t}(x)$, в которой $\sum_{t \in T'} |\Omega_t| \to \min$, а функция $D_\lambda^m(x)$ совпадает с $D_\lambda(x)$ на обучающей выборке.

По аналогии с булевым случаем величину $|\Omega_t|$ можно назвать рангом соответствующего предиката.

Определение. Кратчайшим логическим описанием класса K_λ назовем логическую сумму $D_\lambda^s(x) = \bigvee_{t \in T'' \subseteq T} P_t^{\Omega_t}(x)$, в которой $|T''| \to \min$, а функция $D_\lambda^s(x)$ совпадает с $D_\lambda(x)$ на обучающей выборке.

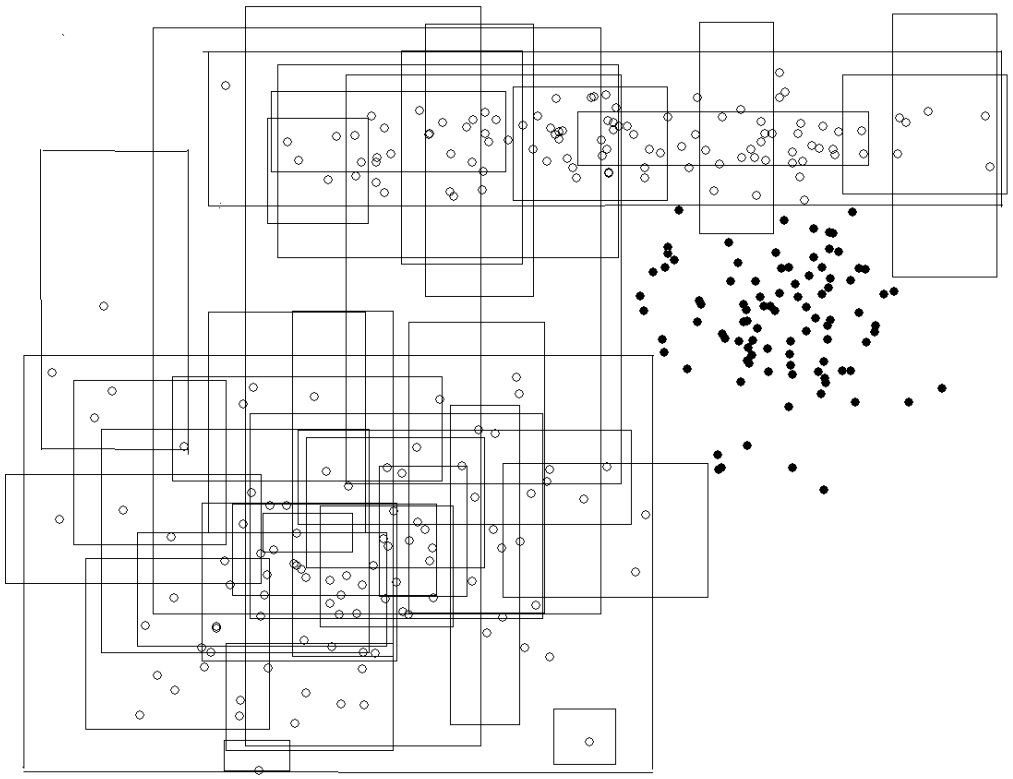

Рис.3. Покрытие эталонных объектов класса «белый кружок» множеством его логических закономерностей.

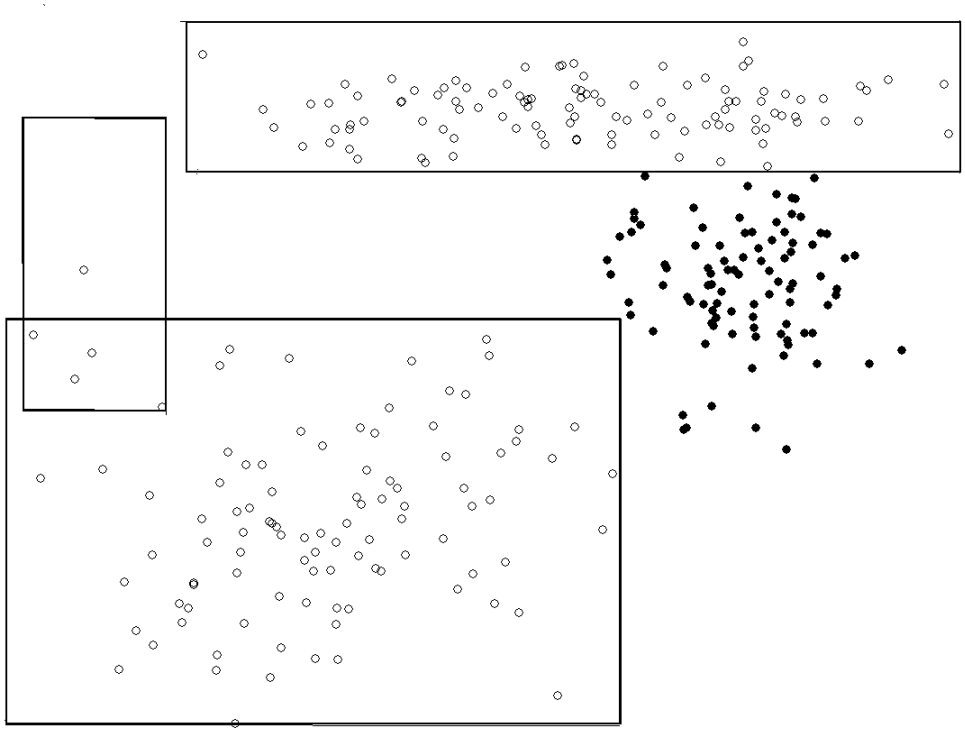

Рис.4. Кратчайшее покрытие эталонных объектов класса «белый кружок» множеством его логических закономерностей

Логические (кратчайшие, минимальные) описания классов являются аналогами представлений частичных булевых функций в виде сокращенных дизъюнктивных нормальных форм (кратчайших, минимальных), а геометрические образы логических закономерностей классов - аналогами максимальных интервалов [6,7].

Задачи поиска минимальных и кратчайших логических описаний формулируются как задачи на покрытия:

$$\sum_{t\in T} a_t y_t \to \cdot \min, \qquad (10)$$

$$\sum_{t\in T} P_t^{\Omega_t}(S_t) y_t \geq 1, \cdots \forall S_t \in K_\lambda, \cdot y_t \in \{0,1\}. \qquad (11)$$

Тогда при $a_t = 1$ единичные компоненты решения задачи (10-11) определяют кратчайшее логическое описание $D_\lambda^s(x)$ класса K_λ, а при $a_t = |\Omega_t|$, - минимальное логическое описание.

Отметим, что исходные множества $P_t^{\Omega_t}(x), t \in T$ могут содержать равные или близкие элементы, «вырожденные» решения, соответствующие локальным максимумам с малыми абсолютными значениями, мощность данных множеств может быть весьма велика (что однако является благоприятным в процедурах распознавания). Данные «свойства» существенно зависят от длины обучающей выборки и самого алгоритма их поиска. В то же время, кратчайшие и минимальные логические описания классов образуют уже неизбыточные подмножества, выражающие как основные свойства данных множеств, так и свойства самих классов. Поэтому

вычисление $D_j^s(S)$ и $D_j^m(S)$ может рассматриваться как один из подходов к проблеме сортировки логических закономерностей классов. Входящие в $D_j^s(S)$ и $D_j^m(S)$ предикаты могут рассматриваться как наиболее компактные представления о классах, включающие как наиболее представительные знания (предикаты, покрывающие большое число эталонов), так и уникальные или редкие (предикаты, покрывающие малое число эталонов или отдельные из них).

Определение. Логической сложностью (компактностью) классов называются величины:

1. $\psi_1(K_j)$=<*число конъюнкций в* $D_j^s(S)$>;
2. $\psi_2(K_j)$=<*число переменных в* $D_j^m(S)$>.

Величина $\Phi(l_0)=\sum_{i=1}^{l}\psi(K_i)$, где ψ - некоторый критерий логической сложности класса, называется логической сложностью задачи.

Естественно ожидать, что если некоторый класс является компактным, простым множеством объектов, хорошо логически отделимым от других классов, то он имеет малое число переменных в минимальном логическом описании класса и/или малое число конъюнкций в кратчайшем.

7. Минимизация признакового пространства в задачах распознавания

Стандартная постановка задачи распознавания предполагает, что начальная информация о классах (обучающая информация) задается выборкой векторных признаковых описаний объектов, представляющих все классы. Во многих случаях система признаков формируется "стихийно". В ее состав включаются все показатели, влияющие на классификацию (хотя бы чисто гипотетически), и которые можно вычислить или измерить. Независимо от числа имеющихся признаков, исходная система признаков, как правило, избыточна и включает признаки, не влияющие на классификацию или дублирующие друг друга. В некоторых практических задачах распознавания затраты на вычисление части признаков могут быть значительными и конкурировать со стоимостью потерь при распознавании. Решение задач обучения при меньшем числе признаков также является более точным, а полученные решения более устойчивыми. Таким образом, решение задач минимизации признаковых пространств является важным во многих отношениях.

Рассмотрим задачу минимизации признакового пространства в следующей постановке. Пусть имеется модель M алгоритмов распознавания, признаковое пространство $X_1, X_2, \ldots, X_N$ и критерий качества $f(A)$ алгоритма A. Требуется найти такое признаковое подпространство $X_{i_1}, X_{i_2}, \ldots, X_{i_n}$, с минимальным n, для которого $f(A) \geq f_0$, где f_0 - некоторый минимально

допустимый порог точности алгоритма распознавания *A,* построенного по данным обучения для данного подпространства.

В силу своего комбинаторного характера, методы перебора значительного числа различных признаковых подпространств практически нереализуемы, поэтому обычно используются процедуры последовательного выбора из системы *k* признаков подсистемы из *k-1* признака. Здесь используются различные общие подходы: последовательный отброс наименее информативных признаков, с использованием кластеризации признаков, и другие. Специальные подходы отбора и преобразования признаков имеются в статистической теории распознавания. Многие модели распознавания включают и свои специфические способы оценки и отбора признаков.

В настоящем разделе рассматривается метод минимизации признакового пространства, ориентированный на модели частичной прецедентности [1,2] и основанный на кластеризации признаков с учетом их информативности и взаимосвязи.

7.1. Информативность признаков и логические корреляции

Задача минимизации признакового пространства рассматривалась для моделей распознавания, основанных на голосовании по системам логических закономерностей [1-3].

Пусть P - некоторое множество предикатов, найденное по данным обучения.

Определение. Величина $wei(i) = N(i)/N$ называется мерой информативности признака X_i , если *N(i)* - число предикатов множества P, содержащих признак X_i .

Пусть *N(i,j)* -число одновременных вхождений признаков X_i , X_j в одну закономерность по множеству P. Величину $Lcorr(i,j) = 1 - \dfrac{N(i,j)}{\min(N(i),N(j))}$ назовем логической корреляцией признаков X_i и X_j . Данная величина равна нулю, когда во всех закономерностях, куда входит признак X_i, присутствует X_j (и наоборот), т.е. признаки "дополняют друг друга". Корреляция равна единице, если ни в одну закономерность с признаком X_i не входит X_j. Отметим, что при выборе критериев $\phi(P_t)$ согласно [2,3] равным признакам будет соответствовать единичная корреляция. В случаях равных или пропорциональных признаков (столбцов таблицы обучения), в силу свойств логических закономерностей N_{ij}=0 (что непосредственно следует из алгоритма их поиска) и, следовательно, r_{ij}=1.

Если ***min***(N_i ,N_j)*=0,* полагаем r_{ij}=0 (данный случай возникает , например, если x_i(S)≡ *const*).

7.2. Кластеризация признаков и выбор подсистем признаков

Рассмотрим задачу нахождения таких кластеров признаков, для которых входящие в них признаки обладают близкими корреляционными

свойствами. В качестве меры корреляционной близости признаков рассмотрим более "тонкий" критерий чем $1 - Lcorr(i, j)$, а именно, основанный на полуметрике $r(i, j) = \sum_{l=1, l \neq i, j}^{k} \left| Lcorr(i,l) - Lcorr(j,l) \right| + k \times (1 - Lcorr(i, j))$.

Первое слагаемое показывает насколько близки признаки по отношению к другим признакам, а второе – насколько они «схожи» между собой. Множитель k добавлен для того, чтобы слагаемые были соразмерны и вносили примерно одинаковый вклад в определение близости между признаками.

В качестве алгоритма кластеризации для заданной полуметрики $r(i, j)$ и фиксированного числа классов использовалась иерархическая группировка [4], в которой расстояние между кластерами определялось согласно функции $r(K_p, K_q) = \max_{i \in K_p, j \in K_q}(r(i, j))$.

После нахождения *n* кластеров в сокращенную подсистему признаков включаются наиболее информативные признаки (по одному из каждого кластера).

Таким образом задача минимизации признакового пространства решается следующим образом. Предполагается, что для исходного признакового пространства выполнено $f(A) \geq f_0$. Вычисляется матрица значений $r(i, j)$.

Пусть на некотором шаге *k=0,1,2,...* получено с помощью кластеризации признаковое подпространство из *N-k* признаков $X^{N-k} = \{X_{i_1}, X_{i_2}, ..., X_{i_{N-k}}\}$, и A_{N-k} - построенный в данном подпространстве алгоритм распознавания. В качестве решения задачи минимизации признакового пространства принимается $X^{N-k} = \{X_{i_1}, X_{i_2}, ..., X_{i_{N-k}}\}$, соответствующее максимальному *k* при ограничении $f(A_{N-k}) \geq f_0$.

7.3. Примеры

На Рис.1 приведены графики изменения точности распознавания в модели распознавания [3] при двух подходах к минимизации признакового пространства на примере задачи распознавания состояния ионосферы [5]. Исходное признаковое пространство включало 34 признака, задача распознавания решалась относительно двух классов, обучающая выборка имела длину 181 объектов, контрольная - 170. Черная линия соответствует последовательному отсеву менее информативных признаков, серая - минимизации признакового пространства согласно предложенному в настоящей работе алгоритму. Видно, что серая линия лежит, как правило, ниже черной. "Волнистость" графиков $f(A)$ является естественным следствием набора факторов (не идеальность процедур вычисления

предикатов $P_l(S)$, малая длина выборок, "частичная несогласованность" выборок, когда информативность признака на обучающей таблице и контрольной имеет некоторое различие). Из рисунка следуют естественные качественные выводы о данной задаче распознавания. Удаление первой трети малоинформативных признаков мало влияет на точность распознавания и не зависит от используемого метода их сокращения. Оставшиеся 20 признаков вполне компенсируют отсутствие остальных 14. При удаление последующих 10 потери при кластеризационной минимизации растут меньше, чем при частотной.

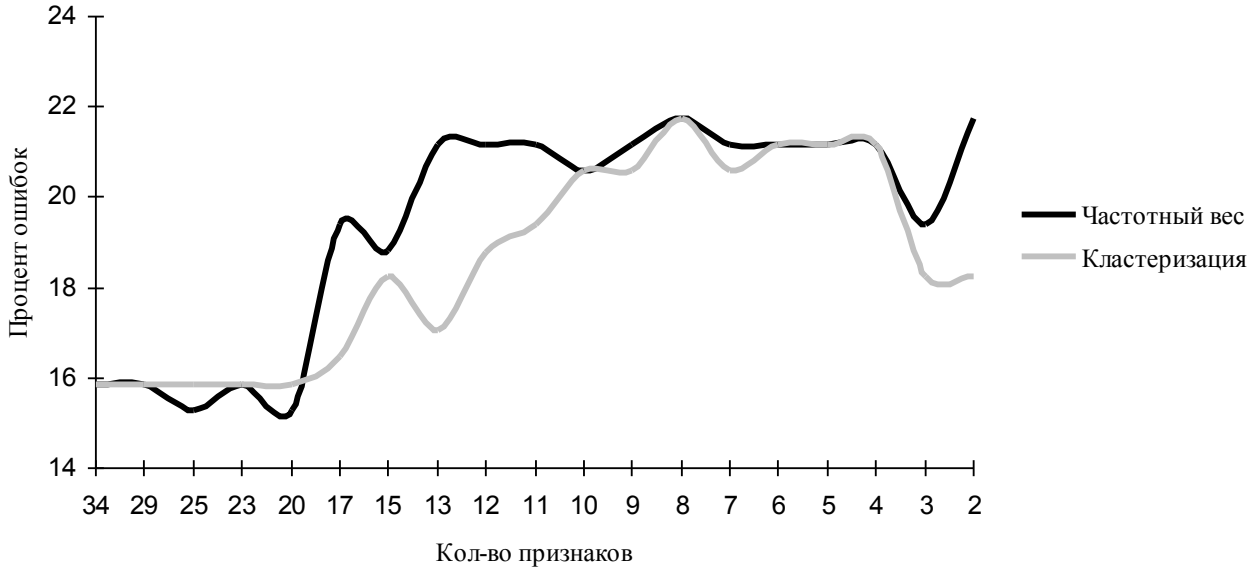

Рис.1 Минимизация признакового пространства на примере задачи распознавания состояния ионосферы

Другой пример практической иллюстрации выполнен для задачи оценки тяжести заболевания пневмонией с параметрами $N=41, l=4$ (рис.2).

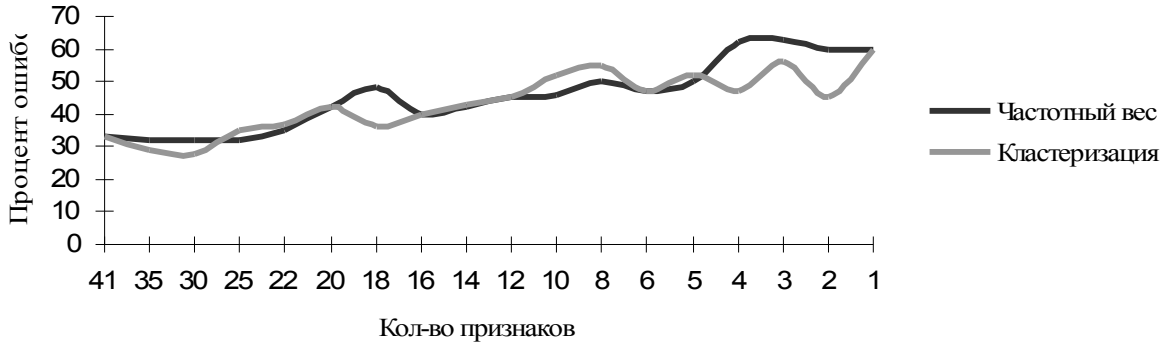

Рис.2. Минимизация признакового пространства на примере задачи оценки тяжести заболевания пневмонией

При всей условности результатов, обусловленных малыми выборками обучения (116 объектов) и контроля (57 объектов) выбор малого числа

информативных признаков с использованием кластеризации явно предпочтительнее.

Литература.
1. Журавлев Ю.И. Об алгебраическом подходе для решения задач распознавания или классификации, Проблемы кибернетики, Наука, Москва, 1978, выпуск 33, стр.5-68.
2. Ryazanov V.V. Recognition Algorithms Based on Local Optimality Criteria // Pattern Recognition and Image Analysis. 1994. Vol.4. no.2. P.98-109.
3. Богомолов В.П., Виноградов А.П., Ворончихин В.А., Журавлев Ю.И., Катериночкина Н.Н., Ларин С.Б., Рязанов В.В., Сенько О.В. Программная система ЛОРЕГ - алгоритмы распознавания, основанные на голосовании по множествам логических закономерностей. Москва, ВЦ РАН, 1998, 63 с.
4. Р.Дуда, П.Харт, Распознавание образов и анализ сцен. Издательство "Мир", Москва, 1976, 511 с.
5. Sigillito, V. G., Wing, S. P., Hutton, L. V., \& Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest, 10, 262-266.

1.2.5. Решающие деревья.

Методы распознавания, основанные на построении решающих деревьев, относятся к типу логических методов.

В данном классе алгоритмов распознавание некоторого объекта осуществляется как прохождение по бинарному дереву из корня в некоторую висячую вершину. В каждой вершине вычисляется определенная логическая функция. В зависимости от полученного значения функции происходит переход далее по дереву в левую или правую вершину следующего уровня. Каждая висячая вершина связана с одним из классов, в который и относится распознаваемый объект, если путь по дереву заканчивается в данной вершине.

Бинарным корневым деревом (БД) называется дерево, имеющее следующие свойства:

а) каждая вершина (кроме корневой) имеет одну входящую дугу;

б) каждая вершина имеет имеет либо две, либо ни одной выходящей дуги.

Вершины, имеющие две выходящие дуги, называются внутренними, а остальные – терминальными или листьями.

Пусть задано N предикатов $\mathrm{P} = \{y_1 = P_1(S), y_2 = P_2(S), ..., y_N = P_N(S)\}$, определенных на множестве допустимых признаковых описаний *{S}*, именуемые признаковыми предикатами. Каждый предикат отвечает на вопрос, выполняется ли некоторое свойство

на объекте *S* или нет. Примерами признаковых предикатов могут быть

$$P_1(S) = \begin{cases} 1, & \text{если } (1.5 \le x_2(S) \le 3.2) \,\&\, (x_5(S) = 0) \\ 0, & \text{в противном случае,} \end{cases}$$

$$P_2(S) = \begin{cases} 1, & \text{если } 0.5x_2(S) + 1.2x_7(S) < 1 \\ 0, & \text{в противном случае.} \end{cases}$$

Каждой строке $S_i = (a_{i1}, a_{i2}, ...a_{in})$ таблицы обучения $T_{nml} = \left(a_{ij}\right)_{m \times n}$ поставим в соответствие бинарную строку значений предикатов на описании $S_i \Rightarrow (y_{i1}, y_{i2}, ..., y_{iN}), y_{ij} = P_j(S_i)$. В результате, таблице $T_{nml} = \left(a_{ij}\right)_{m \times n}$ будет соответствовать бинарная таблица $T^0_{Nml} = \left(y_{ij}\right)_{m \times N}$ бинарных «вторичных описаний» объектов обучения.

Бинарное дерево называется решающим, если выполнены следующие условия:

1. каждая внутренняя вершина помечена признаковым предикатом из P;
2. выходящие из вершин дуги помечены значениями, принимаемыми предикатами в вершине;
3. концевые вершины помечены метками классов;
4. ни в одной ветви дерева нет двух одинаковых вершин.

Пример подобного дерева для конфигурации из трех классов (рис.7) приведен на рис. 8.

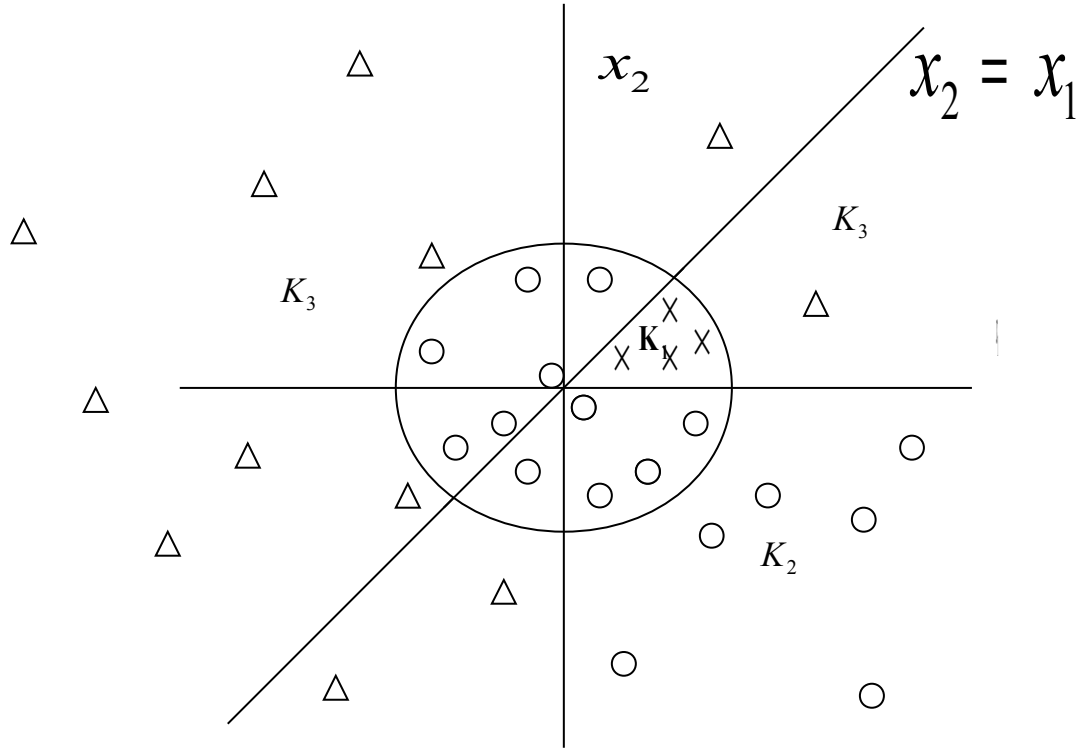

Рис. 7. Конфигурация объектов трех классов. Объекты классов K_1, K_2, K_3 обозначены, соответственно, символами ×, О, Δ.

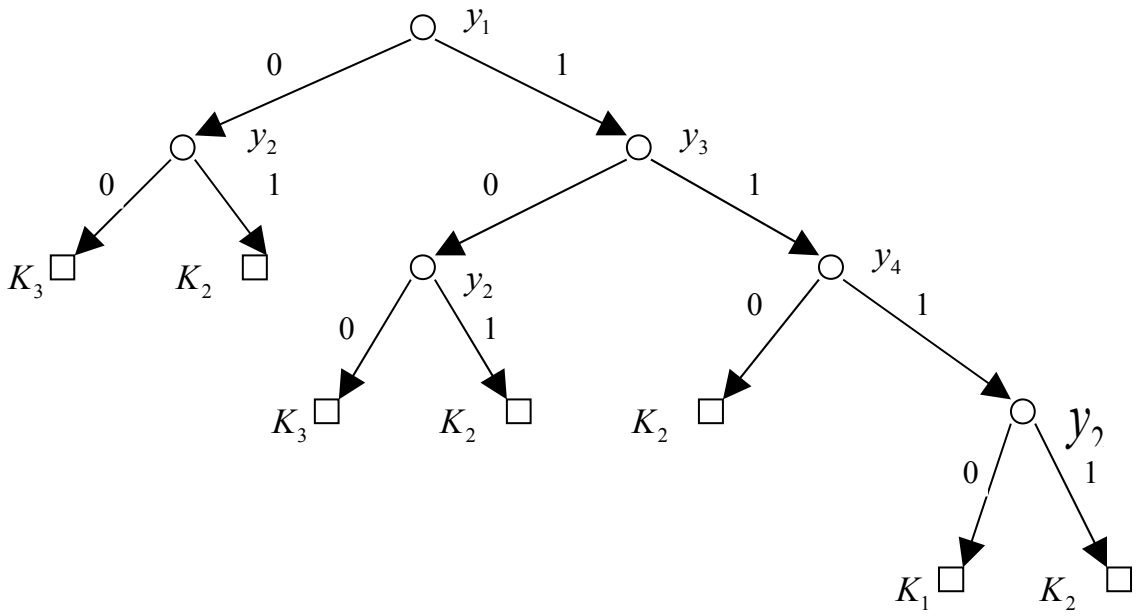

Рис.8. Решающее дерево для классов рис. 7.

На рис.8 приведено некоторое решающее дерево, позволяющее правильно распознавать объекты трех классов, изображенных на рис.7. Вершины дерева помечены следующими предикатами: $y_1 = "x_1 > 0", y_2 = "x_1^2 + x_2^2 < 1", y_3 = "x_1 > x_2", y_4 = "x_2 > 0".$ Данному решающему дереву соответствуют характеристические функции классов $f_1(S) = y_1\bar{y}_2y_3y_4$, $f_2(S) = \bar{y}_1y_2 \vee y_1y_2\bar{y}_3 \vee y_1y_3\bar{y}_4 \vee y_1y_2y_3y_4$, $f_3(S) = \bar{y}_1\bar{y}_2 \vee y_1\bar{y}_2\bar{y}_3$, принимающие значение 1 на объектах «своего» класса и 0 на объектах остальных классов.

Приведем еще один пример решающего дерева (рис.9), построенного непосредственно по таблице обучения:

$$T_{5.4,2} = \begin{pmatrix} 00111 \\ \underline{11101} \\ 10111 \\ \underline{01010} \end{pmatrix} \begin{matrix} \in K_1 \\ \\ \in K_2 \end{matrix}. \qquad (9)$$

В качестве признаковых предикатов используются $y_1 = \bar{x}_1$, $y_2 = x_3$.

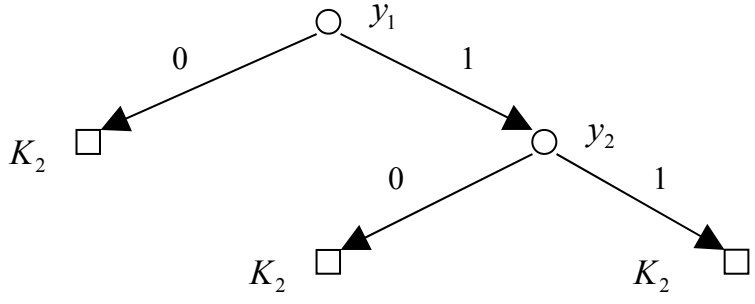

Рис.9. Решающее дерево бинарной таблицы обучения (9)

Здесь в качестве приближений для характеристических функций классов выбраны функции $f_1(S) = y_1 y_2 = \bar{x}_1 x_3$, $f_2(S) = \bar{y}_1 \vee y_1 \bar{y}_2 = x_1 \vee \bar{x}_1 \bar{x}_3$. В данном примере, для построения решающего дерева использованы только два признака.

Задача построения решающего дерева по обучающим данным решается неоднозначно, методам построения решающих деревьев посвящена обширная литература /……../.

1.3. Алгоритмы распознавания, основанные на принципе частичной прецедентности

Настоящий класс алгоритмов выделен в отдельный раздел по нескольким причинам. Представляя исторически логические подходы в теории распознавания, в рамках данного подхода разработана также теория алгоритмов вычисления оценок, объединяющая все существующие методы распознавания и положенная в основу алгебраического подхода в теории распознавания. Теоретические основы алгоритмов частичной прецедентности (вычисления оценок, голосования, или комбинаторно-логических алгоритмов) описаны в многочисленных научных публикациях /…./. В настоящем разделе описана часть алгоритмов, широко используемая в практическом распознавании.

Принципиальная идея данных алгоритмов основана на отнесении распознаваемого объекта *S* в тот класс, в котором имеется большее число «информативных» фрагментов эталонных объектов («частичных прецедентов»), приблизительно равных соответствующим фрагментам объекта *S* [1, 2]. Вычисляются близости – «голоса» (равные 1 или 0) распознаваемого объекта к эталонам некоторого класса по различным информативным фрагментам объектов класса. Данные близости («голоса») суммируются и нормируются на число эталонов класса. В результате вычисляется нормированное число голосов, или оценка объекта *S* за класс $\Gamma_j(S)$ – эвристическая степень близости объекта *S* к классу K_j. После вычисления оценок объекта за каждый из классов, осуществляется отнесение объекта к одному из классов (т.е. распознавание класса объекта) с помощью порогового решающего правила.

1.3.1. Тестовый алгоритм.

Первоначально тестовый алгоритм распознавания был предназначен для бинарных и к-значных признаков. Впервые был опубликован в /…/ и базируется на понятии теста, введенного в 1956 году С.В.Яблонским. Приведем одну из основных модификаций алгоритма.

Определение. Тестом таблицы T_{nml} называется совокупность столбцов $\{i_1, i_2, ..., i_k\}$ таких, что после удаления из T_{nml} всех столбцов, за исключением имеющих номера $\{i_1, i_2, ..., i_k\}$, в полученной таблице $T_{n-k,m,l}$ все пары строк, принадлежащих разным классам, различны. Тест $\{i_1, i_2, ..., i_k\}$ называется тупиковым, если никакая его истинная часть не является тестом /71/.

Пусть найдено множество $\{T\}$ тупиковых тестов таблицы T_{nml} и $T = \{i_1, i_2, ..., i_k\}$ $\in \{T\}$. Выделим в описании распознаваемого объекта S фрагмент описания $(a_{i_1}, a_{i_2}, ..., a_{i_k})$, соответствующий признакам с номерами $i_1, i_2, ..., i_k$. Сравним $(a_{i_1}, a_{i_2}, ..., a_{i_k})$ со всеми фрагментами $(a_{ji_1}, a_{ji_2}, ..., a_{ji_k})$ объектов $S_j, j = 1,2,...,m,$ таблицы T_{nml}.

Число совпадений $(a_{i_1}, a_{i_2}, ..., a_{i_k})$ с $(a_{ji_1}, a_{ji_2}, ..., a_{ji_k})$, $j = m_{k-1}+1, m_{k-1}+2, ..., m_k$, $k = 1,2,...,l$, $m_0 = 0, m_l = m$, обозначим через $\Gamma_j(T)$.

Величина $\Gamma_j(S) = \dfrac{1}{m_j - m_{j-1}} \sum_{T \in \{T\}} \Gamma_j(T)$ (78)

называется оценкой S по классу K_j. Имея оценки $\Gamma_1(T), \Gamma_2(T), ..., \Gamma_l(T)$ объект S легко классифицируется с помощью решающих правил. Например, он относится в тот класс, за который получена его максимальная оценка. В случае наличия нескольких классов с максимальными оценками, происходит отказ от его классификации.

Если отдельное совпадение в (78) назвать «голосом» в пользу класса K_j, то выражение (78) будет нормированным числом голосов за данный класс. В связи с данной интерпретацией, алгоритмы с подобной схемой вычисления оценок называют также алгоритмами голосования.

Тестовый алгоритм является удобным базисом для оценки информативности (важности) признаков. Пусть N_i - число тупиковых тестов таблицы T_{nml}, содержащих *i*-й столбец , а $N = |\{T\}|$ - общее число тупиковых тестов таблицы. Тогда вес *i*-го признака определяется

как $p_i = \frac{N_i}{N}$. Чем больше вес, тем важнее признак для описания объектов. Это утверждение основывается на следующем рассуждении. Таблица T_{nml} является исходным описанием классов и признаков. Тупиковый тест есть не сжимаемое далее по признакам описание классов, сохраняющее разделение классов. Чем в большее число таких неизбыточных описаний входит *i*-й признак, тем он важнее.

Задача нахождения множества тупиковых тестов таблицы T_{nml} сводится к вычислению всех тупиковых покрытий строк некоторой бинарной матрицы ее столбцами.

Рассмотрим постановку данной задачи. Пусть $T_{nml} = \|a_{ij}\|_{m\times n}$, где

$a_{ij} = x_j(S_i) \in \{0,1,...,k-1\}$. Таблице T_{nml} ставится в соответствие матрица сравнения по признакам всевозможных пар объектов из различных классов $C = \|c_{ij}\|_{N\times n}$, где

$$c_{ij} = \begin{cases} 1, & a_{vj} \neq a_{\mu j}, \\ 0, & a_{vj} = a_{\mu j}, \end{cases} \quad i = i(v,\mu),\ S_v \in K_u, S_\mu \in K_v, u \neq v,$$

$$N = \sum_{\substack{i>j \\ i,j=1,2,...,l}} (m_i - m_{i-1})(m_j - m_{j-1}).$$

Столбцы $(i_1, i_2, \ldots, i_k)$ образуют покрытие строк матрицы $C = \|c_{ij}\|_{N\times n}$, если $\forall i = 1,2,...,N, \quad \exists j \in \{i_1, i_2, ..., i_k\}: \ c_{ij} = 1$. Покрытие называется тупиковым, если произвольное его собственное подмножество не является покрытием. Каждому тупиковому тесту соответствует тупиковое покрытие строк матрицы сравнения и наоборот. Нахождение всех тупиковых тестов является сложной вычислительной задачей. Тем не менее здесь существуют подходы, эффективные для таблиц средней размерности / …/. При решении практических задач достаточно нахождение лишь части тупиковых тестов для вычисления оценок согласно (78). В случае наличия признаков вещественнозначных используют обычно один из следующих двух подходов. Область определения каждого вещественнозначного признака разбивается на *k* интервалов. Значение признака из *i*- го интервала полагается равным *i-1*. Далее задача распознавания решается относительно полученной *k*- значной таблицы обучения и *k*- значных описаний распознаваемых объектов. Другой подход связан с введением числовых пороговых

параметров для каждого признака. Для веществennозначных таблиц вводится аналог тупиковых тестов, в котором требование несовпадения значений признаков заменяется их различием не менее чем на соответствующий порог. В обоих подходах при выборе значности новой таблицы или значений пороговых параметров необходимо учитывать, что соответствующие матрицы сравнения не должны содержать нулевых строк, поэтому значность (или пороговые параметры) следует выбирать из требования отделимости *k*-значных «образов» эталонных объектов различных классов (или отделимости с точностью до пороговых значений).

1.3.2. Алгоритмы распознавания с представительными наборами.

Другими алгоритмами данного вида являются алгоритмы типа “Кора” /16,12/, в которых опорные множества связаны со значениями признаков конкретных объектов.

Пусть $S_v \in K_j$, а признаки принимают значения 0 или 1. Набор $u = \{x_{i_1}(S_v), x_{i_2}(S_v),..., x_{i_k}(S_v),\}$ назовем представительным набором для класса K_j, если для любого $S_\mu \in T_{nm}, S_\mu \notin K_j$ хотя бы одно из равенств $x_{i_t}(S_v) = x_{i_t}(S_\mu), t = 1,2,...,k,$ невыполнено. Представительный набор называется тупиковым, если любой его собственный поднабор не является представительным набором, т.е. для любого его поднабора существует равный ему поднабор в каком-либо другом классе. Таким образом, тупиковый представительный набор некоторого класса K_j есть несократимый фрагмент некоторого эталона данного класса, для которого нет равных ему фрагментов эталонов других классов.

Пусть вычислено V_j – множество тупиковых представительных наборов для класса K_j. Тогда степень близости распознаваемого объекта S к классу K_j по заданному множеству тупиковых представительных наборов V_j можно оценить по формуле

$$\Gamma_j(S) = \frac{1}{|V_j|} \sum_{u \in V_j} B(S,u) \qquad (181),$$

где $B(S,u)$ равно 1, если в признаковом описании объекта S имеется фрагмент *u*. Далее классификация объекта осуществляется по максимальной из вычисленных оценок, как и в тестовом алгоритме. Таким образом, в данном алгоритме с представительными наборами распознаваемый объект относится в тот класс, для которого он имеет максимальную часть

его тупиковых представительных наборов в своем признаковом описании. Отметим, что в модели с представительными наборами (как и для тестового алгоритма) возможны и другие способы вычисления оценок (181). Например, другие нормировки или весовые коэффициенты – множители перед $B(S,u)$, равные числу эталонов класса K_j, в которых встречается u.

Для нахождения тупиковых представительных наборов, содержащихся в некотором эталоне S_j, формируются матрицы сравнения S_j со всеми эталонами других классов. Тупиковые покрытия данных матриц сравнения и определяют тупиковые представительные наборы. Однако, если для поиска тупиковых тестов таблицы T_{nml} требуется решать задачу на покрытия для матрицы из $N = \sum_{\substack{i>j \\ i,j=1,2,...,l}} (m_i - m_{i-1})(m_j - m_{j-1})$ строк и n столбцов, то для поиска тупиковых представительных наборов объекта $S_i \in K_j$ задача на покрытия решается для матрицы из $(m - m_j)$ строк и n столбцов. Таким образом, нахождение множества всех тупиковых представительных наборов таблицы T_{nml} требует решения m задач на покрытия но «малой » размерности /Дюкова/.

Вопрос обобщения алгоритмов распознавания с представительными наборами на случаи k- значной и вещественнозначной информации информации решается аналогично тестовому алгоритму. Множество допустимых значений некоторого признака делится на конечное число интервалов, каждому из которых приписывается целое число 0, 1, 2,…, или k. Таблице T_{nml} и распознаваемым объектам ставятся в соответствие строки новых целочисленных значений признаков. Далее процесс определения тупиковых представительных наборов и распознавания полностью идентичен бинарному случаю. Другое распространение на вещественнозначные признаки связано с введением пороговых параметров $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$ и следующей модификацией понятия представительного набора: набор $u = \{x_{i_1}(S_v), x_{i_2}(S_v), ..., x_{i_k}(S_v),\}$ называется представительным набором для класса K_j, если для любого $S_\mu \in T_{nml}, S_\mu \notin K_j$ хотя бы одно из неравенств $\left|x_t(S_v) - x_t(S_\mu)\right| \le \varepsilon_t, t = i_1, i_2, ..., i_k$ будет невыполненным.

Заметим, что дискретизация признаков или введение пороговых параметров $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$ здесь могут быть индивидуальны для каждого эталона. Главным требованием их выбора является отделимость рассматриваемого объекта $S_i \in K_j$ относительно эталонов из CK_j. Отметим, что , в отличие от тестового алгоритма, описанная модель с представительными наборами допускает пересечение классов. В последнем случае оказываются «бесполезными» объекты, по которым классы пересекаются, поскольку данные объекты не содержат представительных наборов. Впрочем, существуют модификации понятия представительного набора, допускающие и пересечение классов.

Можно провести аналогию тестов и представительных наборов с линейными и кусочно-линейными разделяющими поверхностями. Тестовый алгоритм как и модели с разделяющими гиперплоскостями требуют отделимость классов, и в них особое значение имеют «граничные» эталоны классов. Модель с представительными наборами и кусочно-линейные модели являются «менее требовательными» к обучающей информации. Они не требуют, чтобы эталоны разных классов были одновременно отделимы рассматриваемыми функциями (линейными или булевыми). Для их настройки на обучающую информацию оказывается достаточным отсутствие равных эталонов в разных классах.

1.3.3. Алгоритмы распознавания, основанные на вычислении оценок.

Идеи распознавания по частичным прецедентам, первоначально заложенные в тестовом алгоритме распознавания, были обобщены в моделях распознавания, основанных на вычислении оценок. Алгоритмы данных моделей определяются заданием шести последовательных этапов, для которых могут быть использованы различные конкретные способы определения или выполнения. Тестовый алгоритм и алгоритмы с представительными наборами могут быть представлены как частные случаи более общей конструкции. Ниже будут приведены лишь некоторые основные способы их выполнения. Подробно данные вопросы рассмотрены в /…/.

1. **Задание системы опорных множеств алгоритма.** Первым шагом определения АВО является задание множества подсистем признаков, по которым осуществляется сравнение объектов. Пусть Ω_A - некоторая система подмножеств множества *{1,2,...,n}*, называемая системой опорных множеств алгоритма *A*. Элементы $\Omega = \{i_1, i_2, ..., i_k\} \in \Omega_A$ называются опорными множествами алгоритма. Они определяют номера признаков, по которым сравниваются части эталонных и распознаваемых объектов. Примером выбора системы

опорных множеств Ω_A является множество тупиковых тестов. Каждому подмножеству Ω= $\{i_1, i_2, ..., i_k\}$ можно поставить во взаимно однозначное соответствие характеристический булевский вектор $\omega = \{\omega_1, \omega_2, ..., \omega_n\}$, в котором $\omega_j = 1, j = i_1, i_2, ..., i_k$, а остальные компоненты равны нулю. В силу данного соответствия $\Omega \leftrightarrow \omega$, использование данных величин будет для нас равнозначным.

Множество всех n-мерных булевских векторов определяет дискретный единичный куб $E^n = \{\omega : \omega = (\omega_1, \omega_2, ..., \omega_n)\}$, $\omega_i \in \{0,1\}, i = 1,2,...,n$. Число элементов куба равно 2^n.

Теоретические исследования свойств тупиковых тестов для случайных бинарных таблиц показали, что характеристические векторы «почти всех тупиковых тестов» имеют асимптотически (при неограниченном возрастании размерности таблицы обучения) приблизительно одну и ту же длину. Это явилось одним из обоснований выбора в качестве множества Ω_A всевозможных подмножеств *{1,2,...,n}* длины *k*. Значение *k* находится из решения задачи обучения (оптимизации модели) или задается экспертом. В итоге, широко распространенными подходами к выбору Ω_A являются (наряду с тупиковыми тестами) следующие два:

a) $\Omega_A = \{\Omega : |\Omega| = k\}$;

b) $\Omega_A = \{\Omega\}, \Omega \subseteq \{1,2,...,n\}, \Omega \neq \varnothing$.

Второй способ выбора системы опорных множеств, как всевозможных подсистем *{1,2, ...,n},* является «усреднением» первого и не требует нахождения подходящего значения параметра *k.*

2. Задание функции близости. Пусть фиксировано некоторое опорное множество Ω и соответствующий ему характеристический вектор ω. Фрагмент $x_{i_1}(S), x_{i_2}(S), ...x_{i_k}(S)$ объекта $S = (x_1(S), x_2(S), ...x_n(S)$, соответствующий всем единичным компонентам вектора ω, называется ω-частью объекта, и обозначается ω*S*. Под функцией близости $B_\Omega(S_i, S_j)$ будет пониматься функция от соответствующих ω-частей сравниваемых объектов, принимающая значение 1 («объекты близки») или 0 («объекты далеки»). Приведем примеры подобных функций.

a) $B_\Omega(S_\nu, S_\mu) = \begin{cases} 1, & |x_i(S_\nu) - x_i(S_\mu)| \le \varepsilon_i, \forall i : \omega_i = 1, \omega \leftrightarrow \Omega, \\ 0, & \text{иначе.} \end{cases}$

Здесь $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$ - неотрицательные параметры, именуемые «точности измерения признаков».

b) $B_\Omega(S_\nu, S_\mu) = \begin{cases} 1, & \sum_{i=1}^{n} |x_i(S_\nu) - x_i(S_\mu)| \le \varepsilon, \\ 0, & \text{иначе.} \end{cases}$

Здесь ε также некоторый неотрицательный параметр алгоритма.

3. Оценка близости объекта S к эталонному объекту S_i для заданной ω-части. Данная числовая величина формируется на основе функции близости и, возможно, дополнительных параметров.

a) $\Gamma_\Omega(S_i, S) = B_\Omega(S_i, S)$.

b) $\Gamma_\Omega(S_i, S) = w_\Omega B_\Omega(S_i, S)$, где w_Ω - «вес» опорного множества.

c) $\Gamma_\Omega(S_i, S) = \gamma_i (\sum_{i:\omega_i=1} p_i) B_\Omega(S_i, S)$. Здесь γ_i - параметры, характеризующие степень важности объекта S_i (информативность объекта), а $p_1, p_2, \ldots p_n$ - веса (информативность) признаков.

4. Оценка объекта S за класс K_j для заданной ω-части.

a) Функция $\Gamma_j^\Omega(S) = \frac{1}{(m_j - m_{j-1})} \sum_{S_i \in K_j} \Gamma_\Omega(S_i, S)$ является примером естественной оценки близости объекта к классу для заданного подмножества признаков.

5. Оценка объекта S за класс K_j.

Данная функция задает суммарную степень близости распознаваемого объекта S к классу K_j. Приведем обычно используемые выражения для ее вычисления.

a) $\Gamma_j(S) = \sum_{\Omega \in \Omega_A} \Gamma_j^\Omega(S)$.

b) $\Gamma_j(S) = \nu_j \sum_{\Omega \in \Omega_A} \Gamma_j^\Omega(S)$, где ν_j - «вес» класса K_j. Например, в статистической теории распознавания аналогами параметров ν_j являются априорные вероятности классов, которые характеризуют, насколько часто встречаются объекты различных классов .

6. Решающее правило.

Решающее правило есть правило (алгоритм, оператор), относящее объект по вектору оценок за классы в один из классов, или вырабатывающее для объекта «отказ от распознавания». Отказ является более предпочтительным вариантом решения в случаях, когда оценки объекта малы за все классы (объект является принципиально новым, аналоги которого отсутствуют в обучающей выборке), или он имеет две или более близкие максимальные оценки за различные классы (объект лежит на границе классов).

В формальной постановке, решающее правило ***r*** вычисляет для распознаваемого объекта S по вектору оценок $\overline{\Gamma}(S) = (\Gamma_1(S), \Gamma_2(S), ..., \Gamma_l(S))$ булевский вектор $r(\overline{\Gamma}(S)) = (\alpha_1^A(S), \alpha_2^A(S), ..., \alpha_l^A(S)), \alpha_i(S) \in \{0,1,\Delta\}, i = 1,2,...,l.$ Интерпретация обозначений приведена ранее в (898).

a) Пример простейшего решающего правила – отнесение объекта в единственный класс, за который он имеет максимальную оценку.

$$\alpha_j^A(S) = \begin{cases} 1, & \Gamma_j(S) > \Gamma_i(S), i = 1,2,...,l, i \neq j, \\ 0, & \text{иначе.} \end{cases}$$

b) Для «осторожного» принятия решения относительно принципиально новых объектов, или находящихся на границе двух и более классов, обычно достаточно введение в решающее правило двух пороговых параметров

$$\alpha_i^A(S) = \begin{cases} & \Gamma_j(S) > \Gamma_i(S) + \delta_1, i = 1,2,...,l, i \neq j, \\ 1, & \Gamma_j(S) > \delta_2 \sum_{i=1}^{l} \Gamma_i(S), \\ 0, & \text{иначе.} \end{cases}$$

c)
$$\alpha_i^A(S) = \begin{cases} 1, & \sum_{j=1}^{l} \delta_j^i \Gamma_j(S) \geq c_1^i, \\ \Delta, & c_1^i > \sum_{j=1}^{l} \delta_j^i \Gamma_j(S) > c_2^i, \\ 0, & \sum_{j=1}^{l} \delta_j^i \Gamma_j(S) \leq c_2^i, \end{cases} \qquad (5)$$

Здесь $\delta_1, \delta_2, \delta_j^i, c_1^i, c_2^i$ - параметры алгоритма. В последнем случае (5) наличие двух или более единиц интерпретируется как «объект принадлежит нескольким классам». Когда бинарный вектор состоит из одних нулей говорят, что данный объект – выброс, он не похож ни на один из классов, близких его аналогов ранее не наблюдалось.

Использование решающего правила означает фактически переход из признакового пространства в пространство оценок, в котором в качестве разделяющих классы функций используются гиперплоскости, проходящие через начало координат симметрично относительно новых координатных осей (случай **a**), пары гиперплоскостей (случай **b**), и наборы из *l-1* гиперплоскостей.

Варьируя в моделях вычисления оценок правила определения этапов 1-6 (часть примеров их определения приведена выше), можно получить различные модели распознающих алгоритмов типа вычисления оценок.

Если конкретные правила этапов (1-5) определены, то после последовательной подстановки выражений на этапах 2-5 могут быть получены различные общие формулы для вычисления оценок $\Gamma_j(S)$. Например, выбирая первые примеры реализации различных этапов будет получена следующая общая формула для вычисления оценок объекта *S* за классы K_j , *j=1,2,…,l.*

$$\Gamma_j(S) = \frac{1}{(m_j - m_{j-1})} \sum_{\nu=m_{j-1}+1}^{m_j} \sum_{\Omega \in \Omega_A} B_\Omega(S_\nu, S). \qquad (6)$$

При выборе системы опорных множеств согласно вариантам a) или b) прямое вычисление оценок (6) представляется весьма трудоемким. Действительно, при вычислении оценок (6) согласно a) требуется mC_n^k вычислений значений функции близости. В действительности нет необходимости выполнения всех данных вычислений, поскольку при многих вариантах реализации этапов 2-5 и различных системах опорных множеств существуют эффективные комбинаторные формулы вычисления оценок. Например, при использовании в качестве системы опорных множеств $\Omega_A = \{\Omega : |\Omega| = k\}$ и вариантов a) выполнения этапов (2-5), справедлива формула

$$\Gamma_j(S) = \frac{1}{(m_j - m_{j-1})} \sum_{S_i \in K_j} C_{d(S,S_i)}^k , \qquad (354)$$

где $d(S, S_i) = \left| \left\{ \nu : |x_\nu(S_i) - x_\nu(S)| \le \varepsilon_\nu, \nu = 1,2,...,n \right\} \right|$.

При использовании вариантов a) выполнения этапов (2-5) и b) для первого этапа, справедлива формула

$$\Gamma_j(S) = \frac{1}{(m_j - m_{j-1})} \sum_{S_i \in K_j} (2^{d(S,S_i)} - 1), \qquad (355)$$

где $d(S,S_i) = \left| \left\{ \nu : \left| x_\nu(S_i) - x_\nu(S) \right| \le \varepsilon_\nu, \nu = 1,2,...,n \right\} \right|$.

Другие, более сложные и общие способы определения этапов (1-5), а также соответствующие им эффективные формулы вычисления оценок, приведены в /…/ .

1.3.4. Оптимизация многопараметрических моделей распознавания.

Процесс распознавания во многих моделях вычисления оценок предполагает знание числовых параметров модели (веса признаков, веса эталонов, пороговые параметры, и т.п.). Их значения могут быть выбраны непосредственно пользователем исходя из содержательных или эвристических соображений, поскольку многие параметры имеют естественную интерпретацию. Основным же подходом к их вычислению является процесс обучения или оптимизации модели. Желаемым результатом в обоих случаях является нахождение таких значений параметров, при которых будет обеспечена высокая точность распознавания.

Поиск значений параметров, как процесс «обучения с учителем» используется в нейросетевых подходах, методе потенциальных функций, построении линейных разделяющих гиперплоскостей. Применяется следующая общая схема обучения. Задаются начальные значения параметров (например, случайные из некоторого интервала). Алгоритму предъявляется один из обучающих объектов, класс которого известен. Если объект распознается правильно, предъявляется для распознавания следующий объект. Если объект классифицируется неправильно, происходит коррекция параметров «в нужном направлении». Процесс продолжается до достижения стабилизации работы алгоритма, когда последующее обучение не уменьшает общее число ошибок на обучающей выборке.

Более общая постановка процесса «настройки» алгоритмов связана с задачей оптимизации модели, когда каждый конкретный алгоритм модели полностью задается набором значений параметров модели.

Пусть дано параметрическое множество распознающих алгоритмов

$\{A(y\}, y \in D\}$ и на нем определен числовой функционал $\varphi(A)$ качества алгоритма.

Требуется найти такой алгоритм $A^* \in \{A\}$, который доставляет экстремум функционалу:

$$\varphi(A^*) = \underset{A \in \{A\}}{extr}\, \varphi(A).$$

Так, например, модель вычислении оценок со способами выполнения этапов (a,a,c,a,a,c) является следующим параметрическим семейством алгоритмов:

$$\{A_k(k, \varepsilon, p, \gamma, \delta, c_1, c_2), 1 \geq k \geq 0, k - \text{целое}, \varepsilon \geq 0, 1 \geq p \geq 0, 1 \geq \gamma \geq 0, c_1 > c_2\}.$$

Стандартная постановка проблема оптимизации состоит в следующем.

Пусть задана таблица контрольных объектов T'_{nql}, аналогичная таблице обучения, т.е. состоящая из разбитых на *l* классов *m* числовых строк – признаковых описаний объектов $S'_i = (x_1(S'_i), x_2(S'_i), ..., x_n(S'_i))$. Для определенности считаем, что

$$S'_i \in K_j, i = q_{j-1} + 1, q_{j-1} + 2, ..., q_j, q_0 = 0, q_l = q.$$

Пусть $\alpha_{ij} = \begin{cases} 1, & S'_i \in K_j, \\ 0, & S'_i \notin K_j. \end{cases}$ Обозначим $\alpha_{ij}^A = \alpha_j(S'_i)$.

Определение. Стандартным функционалом качества распознавания называется функционал $\varphi(A) = \frac{1}{ql} \sum_{i=1}^{q} \sum_{j=1}^{l} \left| \alpha_{ij} - \alpha_{ij}^A \right|$.

В статистической теории распознавания данный критерий называют эмпирическим риском. Очевидными эквивалентными его вариантами являются «доля правильных ответов» или «число правильных ответов».

Постановка задачи оптимизации моделей вычисления оценок (и многих других моделей) может быть записана в терминах систем неравенств. Для простоты ограничимся случаем двух классов и моделью (a,a,c,a,a,a) .

Условием правильного распознавания некоторого контрольного объекта $S'_i \in K_j$ является выполнение неравенства $\Gamma_1(S'_i) > \Gamma_2(S'_i)$, если объект из первого класса, и $\Gamma_2(S'_i) > \Gamma_1(S'_i)$, если объект из второго класса. Тогда число правильно распознанных объектов при некотором варианте выбора параметров модели будет равно числу выполненных неравенств системы (*).

$$\Gamma_1(S'_i) > \Gamma_2(S'_i),\ i = 1, 2, ..., m_1,$$

$$\Gamma_2(S'_i) > \Gamma_1(S'_i),\ i = m_1 + 1, m_1 + 2, ..., m \qquad (*)$$

Учитывая, что оценки являются билинейными формами от параметров $p_1, p_2, \ldots p_n$ и $\gamma_1, \gamma_2, \ldots \gamma_m$, задача оптимизации модели может быть сформулирована следующим образом: «Найти максимальную совместную подсистему системы (**) и некоторое ее решение».

$$\sum_{j=1}^{m}\sum_{i=1}^{n} b_{ij}^{k}(\varepsilon) p_i \gamma_j > 0, \; k = 1,2,\ldots,q, \qquad (**)$$

Данная задача является сложной оптимизационной задачей даже для частного случая линейной системы, когда в (**) фиксированы параметры $\varepsilon_1, \varepsilon_2, \ldots \varepsilon_n, p_1, p_2, \ldots p_n$ или параметры $p_1, p_2, \ldots p_n, \gamma_1, \gamma_2, \ldots \gamma_m$.

Примечание. *Фундаментальные теоретические результаты, связанные с исследованием задачи поиска максимальных совместных подсистем, получены в Уральском Университете (Мазуров, Хачай). Комбинаторные алгоритмы для задач малой размерности созданы в ВЦ РАН Катериночкиной Н.Н. В системе «РАСПОЗНАВАНИЕ» используется эвристический алгоритм, основанный на релаксационном спуске. Оптимизация стандартного функционала качества как последовательность вспомогательных оптимизационных задач в пространстве параметров ε при фиксированных p, γ и пространстве p, γ, при фиксированных ε, рассматривалась в /.../.*