Московский государственный университет
имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Математические основы теории прогнозирования

(курс лекций)

лектор — академик РАН Ю. И. Журавлев

2008

Оглавление

Курс лекций, прочитанный для 3 потока IV курса факультета ВМиК, набран в системе LaTeX студентами:
1 лекция — П. Клеменков
2 лекция — А. Гудков
3 лекция — К. Симонян
4 и 5 лекции — А. Фокин

Лекция 1

1.1 Стандартная задача распознавания

Пусть дано множество M, являющееся суммой подмножеств $K_1, \dots, K_l$, называемых обычно классами.

$$M = \bigcup_{j=1}^{l} K_j$$

Различают случаи а) $K_u \cap K_v = \varnothing$, б) $K_u \cap K_v$, вообще говоря, не пусто. В случае а) говорят о задаче с непересекающимися, в случае б) — пересекающимися классами (множества $K_j, j = 1 \dots l$ принято называть классами).

В дальнейшем рассматриваются только M специального вида: элементы M являются наборами длины n: $\tilde{a} \in M, \tilde{a} = (a_1, a_2, \dots, a_i, \dots, a_n)$. При этом для каждого номера $i, i = 1 \dots n$, определено множество допустимых значений M_i, являющееся метрическим пространством с метрикой ρ_i, т.е. выполнены аксиомы: $\rho_i(c, d) \geq 0, \rho_i(c, c) = 0, \rho_i(c, d) = \rho_i(d, c), \rho_i(c, e) + \rho_i(e, d) \geq \rho_i(c, d), c, d, e \in M_i$. В некоторых случаях выполнения последней аксиомы (аксиомы треугольника) не требуют. Тогда говорят, что в M_i введена полуметрика.

В качестве исходной информации задаются некоторые сведения о множестве M и классах $K_1, \dots, K_l$.

В дальнейшем в качестве исходной информации рассматривается так называемая стандартная обучающая информация I: выделяется конечное множество $S_1, \dots, S_m$ элементов из $M : S_i = (a_{i1}, a_{i2}, \dots, a_{it}, \dots, a_{in}), i = 1 \dots m, a_{it} \in M_t$, для которых известно, в какие из $K_1, \dots, K_l$ они входят. Последнее оформляется заданием информационного вектора $\tilde{\alpha}(S_i) = (\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{ij}, \dots, \alpha_{in}), (\alpha_{ij} = 1) \to S_i \in K_j, (\alpha_{ij} = 0) \to S_i \notin K_j,\ i = 1 \dots m$, $j = 1 \dots l$.

Для удобства данные об элементах S_i и их информационных векторах представляют в виде таблиц:

$$\underbrace{\begin{array}{c|cccccc}
 & 1 & 2 & \dots & t & \dots & n \\
\hline
S_1 & a_{11} & a_{12} & \dots & a_{1t} & \dots & a_{1n} \\
S_2 & a_{21} & a_{22} & \dots & a_{2t} & \dots & a_{2n} \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots \\
S_i & a_{i1} & a_{i2} & \dots & a_{it} & \dots & a_{in} \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots \\
S_m & a_{m1} & a_{m2} & \dots & a_{mt} & \dots & a_{mn}
\end{array}}_{T_1}$$

$$\underbrace{\begin{array}{cccccc|c}
K_1 & K_2 & \dots & K_j & \dots & K_l & \\
\hline
\alpha_{11} & \alpha_{12} & \dots & \alpha_{1j} & \dots & \alpha_{1l} & \tilde{\alpha}(S_1) \\
\alpha_{21} & \alpha_{22} & \dots & \alpha_{2j} & \dots & \alpha_{2l} & \tilde{\alpha}(S_2) \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\alpha_{i1} & \alpha_{i2} & \dots & \alpha_{ij} & \dots & \alpha_{il} & \tilde{\alpha}(S_i) \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\alpha_{m1} & \alpha_{m2} & \dots & \alpha_{mj} & \dots & \alpha_{ml} & \tilde{\alpha}(S_m)
\end{array}}_{T_2}$$

Совокупность таблиц T_1, T_2 называется стандартной обучающей информацией I, таблица T_2 называется информационной матрицей.

Стандартная задача распознавания: пусть задан элемент $S \in M, S \notin \{S_1, S_2, \dots, S_i, \dots, S_m\}$. Найти алгоритм A, который, используя только I и представление S строит информационный вектор $\tilde{\alpha}(S) = (\tilde{\alpha_1}(S), \dots, \tilde{\alpha_j}(S), \dots, \tilde{\alpha_l}(S))$.

$$A(I, S) = \tilde{\alpha}(S).$$

В задачах распознавания часто обучающая информация I оказывается недостаточной для построения "правильного" вектора $\tilde{\alpha}(S)$. Поэтому допускаются и широко используются эвристические алгоритмы, допускающие ошибки и отказы при вычислении координат информационных векторов.

Такие алгоритмы $\tilde{A}(I, S)$ строят квази-информационные векторы $\tilde{\beta}(S) = (\beta_1(S), \dots, \beta_j(S), \dots, \beta_l(S))$. При этом возможно, что: $\beta_j(S) \neq \tilde{\alpha_j}(S)$ (ошибка в распознавании), $\beta_j(S) = \Delta$ — так кодируется отказ от вычисления j-й координаты информационного вектора.

В литературе описано значительное число таких эвристических алгоритмов, допускающих небольшое (допустимое при практическом применении) число ошибок и отказов при решении достаточно узких классов реальных прикладных задач. Мы опишем два таких алгоритма, получивших большое распространение при прогнозировании в геологии, медицине, технике и т.п.

В дальнейшем координаты $1, 2, \dots, n$, задающие n-мерные объекты в M, будем называть признаками.

1.2 Алгоритм "Кора" (Вайнцвайг, Бонгарт)

Применяется для M, элементами которых являются бинарные признаки: $M_i = \{0, 1\}$, $i = i \dots n$, в основном для задач с двумя непересекающимися классами: $M = K_1 \cup K_2$, $K_1 \cap K_2 = \varnothing$.

В таблице $\|a_{ij}\|_{m \times n}$, задающей объекты с известной классовой принадлежностью, пусть $S_1, \dots, S_q$ принадлежат K_1, $S_{q+1}, \dots, S_m$ принадлежат K_2. Просматриваем все тройки признаков r, u, v (число таких троек, очевидно, равно $\binom{n}{3}$) и анализируем часть таблицы T_1, составленной только из столбцов r, u, v:

$$\begin{array}{ccc}
a_{1r} & a_{1u} & a_{1v} \\
a_{2r} & a_{2u} & a_{2v} \\
\cdots & \cdots & \cdots \\
a_{ir} & a_{iu} & a_{iv} \\
\cdots & \cdots & \cdots \\
a_{qr} & a_{qu} & a_{qv} \\
\hline
a_{q+1r} & a_{q+1u} & a_{q+1v} \\
a_{q+2r} & a_{q+2u} & a_{q+2v} \\
\cdots & \cdots & \cdots \\
a_{jr} & a_{ju} & a_{jv} \\
\cdots & \cdots & \cdots \\
a_{mr} & a_{mu} & a_{mv}
\end{array}$$

Среди первых q строк выделяем и фиксируем все тройки, не совпадающие ни с одной из троек в строках $q+1, \ldots, m$. Формируем множество таких троек $\{(a_{ir}, a_{iu}, a_{iv})\}$. Аналогично выделяем все тройки (a_{jr}, a_{ju}, a_{jv}), не совпадающие ни с одной из первых q троек. Множества $\{(a_{ir}, a_{iu}, a_{iv})\}$, $\{(a_{jr}, a_{ju}, a_{jv})\}$ назовем, соответственно, характеристиками классов K_1, K_2. Такие характеристики формируем для всех троек (r, u, v).

Пусть задан для распознавания объект $S = (b_1 \ldots b_r \ldots b_u \ldots b_v \ldots b_n)$. Сравниваем все характеристики всех троек для K_1 с соответствующими тройками в распознаваемом объекте S. Число совпадений $(a_{ir}, a_{iu}, a_{iv}) = (b_r, b_u, b_v)$ обозначаем $\Gamma(S, K_1)$ — число голосов, поданных для S за класс K_1. Аналогично формируем величину $\Gamma(S, K_2)$: число совпадений $(a_{jr}, a_{ju}, a_{jv}) = (b_r, b_u, b_v)$. Вводим пороговый параметр ν.

Если $\Gamma(S, K_1) - \nu > \Gamma(S, K_2)$, относим S классу K_1, при $\Gamma(S, K_2) - \nu > \Gamma(S, K_1)$ — в класс K_2. В остальных случаях алгоритм отказывается от классификации. На практике часто полагают $\nu = 0$.

Пример 1

Дана таблица T_1

	1	2	3	4	5
S_1	1	0	1	0	0
S_2	0	1	0	1	0
S_3	0	0	1	0	1
S_4	1	0	0	1	0
S_5	1	0	0	0	1
S_6	0	1	0	0	1

Имеем $\binom{5}{3} = 10$ троек признаков. Перечислим характеристики для K_1 и K_2.

Характеристики для K_1:

$$\begin{array}{ll}
1.(1,2,3) : (101), (001) & 2.(1,2,4) : (011), (000) \\
3.(1,2,5) : (010), (001) & 4.(1,3,4) : (110), (001), (010) \\
5.(1,3,5) : (100), (000), (011) & 6.(1,4,5) : (100), (010) \\
7.(2,3,4) : (010), (101) & 8.(2,3,5) : (010), (100), (011) \\
9.(2,4,5) : (000), (110) & 10.(3,4,5) : (100), (101)
\end{array}$$

Характеристики для K_2

$1.(1,2,3) : (100)$	$2.(1,2,4) : (101),(010)$
$3.(1,2,5) : (101),(011)$	$4.(1,3,4) : (101),(100),(000)$
$5.(1,3,5) : (100),(101),(001)$	$6.(1,4,5) : (110),(101)$
$7.(2,3,4) : (001),(000),(100)$	$8.(2,3,5) : (000),(001),(101)$
$9.(2,4,5) : (010),(101)$	$10.(3,4,5) : (011)$

Очевидно, что с увеличением n (числа признаков) число троек в характеристиках растет весьма быстро. Поэтому при реальных решениях обязательно использование компьютеров.

Заметим, что для объекта $S = (00000) : \Gamma(S, K_1) = \Gamma(S, K_2) = 3$. Поэтому алгоритм не классифицирует этот объект.

При распознавании объекта $S = (10101)$ имеют место совпадения с элементами характеристики K_1: $(1,2,3),(101);(1,3,4),(110);(2,3,4),(010);(2,3,5),(011);(3,4,5),(101)$. Следовательно, $\Gamma(S, K_1) = 5$. Легко проверить, что $\Gamma(S, K_2) = 2$. Алгоритм "Кора" заносит S в класс K_1.

1.3 Тестовый алгоритм (Ю. И. Журавлев)

Пусть задана бинарная таблица $\|a_{ij}\|_{m\times n}$, строки которой $S_1, \ldots, S_m$ разделены на два класса, причем $S_1, \ldots, S_q$ — строки первого класса K_1, $S_{q+1}, \ldots, S_m$ — строки второго класса K_2.

Набор столбцов с номерами $k_1, \ldots, k_l$ образует тест, если после удаления из таблицы всех столбцов за исключением вышеозначенных, ни одна из строк из K_1 не совпадет ни с одной из строк класса K_2. Тест называется тупиковым, если при удалении из него хотя бы одного столбца, хотя бы одна из строк K_1 совпадет хотя бы с одной из строк K_2.

Предположим, что построены все тупиковые тесты $T_1, \ldots, T_r$ бинарной таблицы, $T_i = \{n_{i1}, \ldots, n_{ip(i)}\}$, $i = 1 \ldots r$, n_{uv} — номера столбцов, входящих в тест.

Распознаваемый объект $S = (a_1 \ldots a_n)$ последовательно совмещается с тупиковыми тестами. При работе с тестом T_i набор $a_{n_{i1}}, \ldots, a_{n_{ip(i)}}$ сравнивается по столбцам теста со всеми строками таблицы $\|a_{ij}\|_{m\times n}$. При этом возможно совпадение со строкой не более чем в одном из классов K_1, K_2 (это следует из определения теста). Число совпадений суммируется отдельно для классов K_1, K_2. Полученные суммы $\Gamma(S, K_1)$, $\Gamma(S, K_2)$ используются для классификации объекта S так же, как в алгоритме "Кора".

Понятия "тест" и "тупиковый тест" нетрудно распространить для таблиц, составленных из элементов произвольной природы. Необходимо только, чтобы элементы каждого столбца содержались в метрическом пространстве. Обозначим метрику этого пространства через ρ_t. Два элемента a_{it}, a_{jt} назовем различимыми, если $\rho_t(a_{it}, a_{jt}) > \epsilon_t$; в противном случае a_{it}, a_{jt} неразличимы. В определении теста достаточно заменить слова "равны" и "не равны" на "неразличимы" и "различимы". Значение ϵ_t задается из "содержательных" соображений или определяется при решении другой задачи.

Построение совокупности тупиковых тестов связано с решением системы базовых уравнений.

Пусть дана система

$$f_i(x_1 \ldots x_n) = 1, i = 1 \ldots k. \tag{1.1}$$

Система (1.1) эквивалентна одному уравнению

$$\prod_{i=1}^{k} f_i(x_1 \ldots x_n) = 1. \tag{1.2}$$

Представим f_i в виде дизъюнктивной нормальной формы (д.н.ф.)

$$f_i = \mathfrak{D}_i = \bigvee_i \mathcal{K}_{it(i)},$$

где $\mathcal{K}_{it(i)}$ — элементарные конъюнкции, т.е. произведения вида $x_{j_1}^{\sigma_1} \cdot \ldots \cdot x_{j_p}^{\sigma_p}, x^\sigma = x$ при $\sigma = 1, \bar{x}$ при $\sigma = 0$.

Выполним в (1.2) операции логического умножения и получим финальную д.н.ф.

$$\bigvee_{i=1}^{r} Q_i = 1. \qquad Q_i = x_{i_1}^{\delta_1} \cdot \ldots \cdot x_{i_p}^{\delta_p}.$$

Последовательно решаем уравнения $Q_i = 1;\ x_{i_1} = \delta_1, \ldots, x_{i_p} = \delta_p$, остальные $x_j = 0, 1$. Совокупность всех решений и есть совокупность всех решений системы (1.1).

Выведем систему уравнений для построения всех тупиковых тестов таблицы $\|a_{ij}\|_{m \times n}$, в которой строки $S_1, \ldots, S_q$ принадлежат K_1, а строки $S_{q+1}, \ldots, S_m$ — классу K_2.

Сопоставим столбцам $1, 2, \ldots, n$ булевские переменные $x_1, \ldots, x_n$. Напишем систему из $q \cdot (m - q)$ булевых уравнений. Паре $S_i = (a_{i1}, \ldots, a_{in}) \in K_1,\ S_j = (a_{j1}, \ldots, a_{jn}) \in K_2$ сопоставим уравнение

$$f_{ij} = \bigvee_{a_{it} \neq a_{jt}} x_t, \qquad i = 1, \ldots, q;\ j = q+1, \ldots, m$$

$$\prod_{\substack{i=1,\ldots,q \\ j=q+1,\ldots,m}} f_{ij} = 1.$$

Выполняем умножение и приходим к д.н.ф. $\bigvee x_{u_1} \cdot \ldots \cdot x_{u_p}$, в которой проводим все упрощения $\mathcal{K} \vee \mathcal{K} \cdot \tilde{\mathcal{K}} = \mathcal{K}$.

Финальное уравнение

$$\bigvee x_{l_1} \cdot \ldots \cdot x_{l_v}$$

определяет все тупиковые тесты $(l_1, \ldots, l_v)$.

Обоснование алгоритма см. в лекции 4.

Процесс умножения при переходе к финальному уравнению и реализация функций в классе д.н.ф. весьма трудоемки. В тестовом алгоритме последнее отсутствует, т.к. уравнения сразу задаются в виде д.н.ф. Процесс умножения можно существенно упростить, используя специфику булевой алгебры. Укажем несколько упрощающих приемов.

а) из двух уравнений $f = 1,\ f \vee \tilde{f} = 1$ второе можно удалить, т.к. оно является следствием первого.

б) пусть даны уравнения $f_0 \vee f_i = 1,\ i = 1 \ldots k$; тогда $\prod_{i=1}^{k}(f_0 \vee f_i) = f_0 \vee f_1 \cdot \ldots \cdot f_k$. Действительно: $f_0 \cdot f_0 = f_0,\ \ f_0 \vee f_0 \cdot f_i = f_0$. (правило поглощения)

в) $\mathcal{K} \cdot (\mathcal{K} \cdot \mathcal{K}') = K \cdot K', \qquad Q \vee Q = Q.$

Существует большое число других упрощающих правил. Эффективность трех приведенных выше продемонстрируем на примере, рассмотренном в алгоритме "Кора".

	1	2	3	4	5
S_1	1	0	1	0	0
S_2	0	1	0	1	0
S_3	0	0	1	0	1
S_4	1	0	0	1	0
S_5	1	0	0	0	1
S_6	0	1	0	0	1

$S_1, S_2, S_3 \in K_1;\ \ S_4, S_5, S_6 \in K_2$

Имеем 9 уравнений, получаемых при сравнении строк $S_i, i = 1,2,3$ со строками S_j, $j = 4,5,6$.

$$\begin{array}{lll}
(S_1,S_4): x_3 \vee x_4 = 1; & (S_1,S_5): x_3 \vee x_5 = 1; & (S_1,S_6): x_1 \vee x_2 \vee x_3 \vee x_5 = 1 \\
(S_2,S_4): x_1 \vee x_2 = 1; & (S_2,S_6): x_4 \vee x_5 = 1; & (S_2,S_5): x_1 \vee x_2 \vee x_4 \vee x_5 = 1 \\
(S_3,S_5): x_1 \vee x_3 = 1; & (S_3,S_6): x_2 \vee x_3 = 1; & (S_3,S_4): x_1 \vee x_3 \vee x_4 \vee x_5 = 1
\end{array}$$

По правилу а) удаляются уравнения (S_1,S_6), (S_2,S_5), (S_3,S_4). Среди оставшихся выделим уравнения $x_3 \vee x_4 = 1$, $x_3 \vee x_5 = 1$, $x_1 \vee x_3 = 1$, $x_2 \vee x_3 = 1$. По правилу б) произведение левых частей даст $x_3 \vee x_1 \cdot x_2 \cdot x_4 \cdot x_5 = 1$. Перемножаем оставшиеся два уравнения

$$(x_1 \vee x_2) \cdot (x_4 \vee x_5) = x_1x_4 \vee x_1x_5 \vee x_2x_4 \vee x_2x_5$$

Перемножая левые части двух полученных уравнений и, используя в), имеем:

$$x_1x_3x_4 \vee x_1x_3x_5 \vee x_2x_3x_4 \vee x_2x_3x_5 \vee x_1x_2x_4x_5 = 1.$$

К последней д.н.ф. правило поглощения неприменимо, поэтому наборы

$$(1,3,4),(1,3,5),(2,3,4),(2,3,5),(1,2,4,5) \quad \text{образуют все тупиковые тесты.}$$

Классифицируем, как в алгоритме “Кора” набор $S = (10101)$. По тесту $(1,3,4)$ имеем совпадение с $S_1 \in K_1$; по $(1,3,5)$ нет совпадений ни с одной строкой; по $(2,3,4)$ — совпадение с $S_1, S_3 \in K_1$; по $(2,3,5)$ — с $S_3 \in K_1$; по $(1,2,4,5)$ — с $S_5 \in K_2$. Следовательно

$$\Gamma(S,K_1) = 3, \quad \Gamma(S,K_2) = 1.$$

Вывод: при $\nu < 2: S \in K_1$, при $\nu \geq 2$ алгоритм откажется от распознавания.

Лекция 2

2.1 Логические алгоритмы распознавания

Для избежания громоздких выкладок и привлечения теории функций k-значной логики ограничимся задачей распознавания с двумя непересекающимися классами K_1, K_2, причем признаки будут принимать только значения 0,1.

В дальнейшем объекты исходной информации I задаются бинарными наборами $S_1, \dots, S_r, S_{r+1}, \dots, S_m$, где $S_i = (\alpha_{i1} \dots \alpha_{ik} \dots \alpha_{in})$, $i = 1 \dots n$. Объекты S_i, $i = 1 \dots r$ принадлежат K_1, объекты S_i, $i = r+1, \dots, m$ — классу K_2.

Напомним некоторые сведения из теории булевых функций (функций алгебры логики).

Каждая $f(x_1, \dots, x_n)$, вообще говоря, неоднозначно представима дизъюнктивной нормальной формой д.н.ф. $\bigvee\limits_i \mathcal{K}_i$, где $\mathcal{K}_i$ — элементарные конъюнкции. Если $\mathcal{K}_i = x_1^{\sigma_1}, \dots, x_k^{\sigma_k}$, то k — ранг конъюнкции,

$$x^\sigma = \begin{cases} x, & \text{если } \sigma = 1; \\ \overline{x}, & \text{если } \sigma = 0. \end{cases}$$

Если $\mathcal{N}_f$ — множество единиц f, $\mathcal{N}_\mathcal{K}$ — интервал, K - множество единиц конъюнкции $\mathcal{K}$, то $f = \bigvee\limits_{i=1}^{t} \mathcal{K}_i \Leftrightarrow \mathcal{N}_{\mathcal{K}_i} = \bigcup\limits_{i=1}^{t} \mathcal{N}_{\mathcal{K}_i}$. Интервал называется максимальным, а соответствующая ему элементарная конъюнкция — простой импликантой, если не существует $\mathcal{N}_{\mathcal{K}'} : \mathcal{N}_\mathcal{K} \subset \subset \mathcal{N}_{\mathcal{K}'} \subset \mathcal{N}_f$.

Пусть $\mathcal{N}_{\mathcal{K}_i}, \dots, \mathcal{N}_{\mathcal{K}_c}$ — совокупность всех максимальных интервалов функции f. Д.н.ф. $D_c(f)$ называют сокращенной д.н.ф. функции f. Каждая д.н.ф. минимальной сложности получается удалением из $D_c(f)$ некоторых э.к. ($D_c(f) = \bigvee\limits_{i=1}^{l} \mathcal{K}_i$).

Напомним, что сложностью д.н.ф называется сумма рангов входящих в нее э.к.

Построение минимальных д.н.ф подразделяется на следующие этапы:

I. строится произвольная д.н.ф. D_f, реализующая f

II. к D_f применяются преобразования $x_i\mathcal{K}_u \vee \overline{x}_i\mathcal{K}_v \to x_i\mathcal{K}_u \vee \overline{x}_i\mathcal{K}_v \vee \mathcal{K}_u\mathcal{K}_v$; $\mathcal{K} \vee \mathcal{K}\mathcal{K}' \to \mathcal{K}$ до тех пор, пока это возможно. Построенная д.н.ф. называется сокращенной.

III. в д.н.ф. $D_c(f) = \bigvee\limits_{i=1}^{l} \mathcal{K}_i$ выбирается произвольный интервал $\mathcal{N}_{\mathcal{K}_j}, 1 \leq j \leq l$, такой, что $\mathcal{N}_{K_j} \subset \bigcup\limits_{i \neq j} \mathcal{N}_{K_i}$. Э.к. $\mathcal{K}_j$ удаляются из $D_c(f)$, оставшиеся интервалы образуют покрытие $\mathcal{N}_f$; поэтому $D_c(f) \setminus \mathcal{K}_j$ реализует f. Процесс повторяется до тех пор, пока в покрытии не остаются только интервалы, не покрывающиеся суммой остальных $\mathcal{N}_{\mathcal{K}_{U_1}}, \ldots, \mathcal{N}_{\mathcal{K}_{U_p}}$. Соответствующая д.н.ф называется тупиковой для $f : D_1(f) = = \bigvee\limits_{i=1}^{p} K_{U_i}$.

Процесс удаления конъюнкций (их интервалов из покрытия) не однозначен, поэтому число тупиковых д.н.ф. может быть велико. Очевидно, что среди тупиковых содержатся и все минимальные д.н.ф.

Для дальнейшего нам понадобятся несколько утверждений и алгоритмов:

I. Как относительно нетрудоемко построить д.н.ф. приемлемой сложности, реализующую f

II. Найти аналитический критерий, позволяющий легко проверить:

$$\mathcal{N}_{\mathcal{K}} \subseteq \bigcup_{i=1}^{q} \mathcal{N}_{\mathcal{K}_i},$$

что необходимо для построения тупиковых д.н.ф.

III. Как влияют на соотношения $\mathcal{N}_{\mathcal{K}} \subseteq \bigcup \mathcal{N}_{\mathcal{K}_i}$, $\mathcal{N}_{\mathcal{K}} \not\subseteq \bigcup \mathcal{N}_{\mathcal{K}_i}$ преобразования $x_i \to x_j^{\sigma_{ij}}$, $\begin{pmatrix} i \\ j \end{pmatrix}$ — подстановка (преобразование взаимно однозначно), $\sigma_{ij} \in \{0, 1\}$

Проверка соотношения $\mathcal{N}_{\mathcal{K}} \subseteq \bigcup\limits_{i=1}^{l} \mathcal{N}_{\mathcal{K}_i}$. Не ограничивая общности считаем, что в $\mathcal{K}$ и $\mathcal{K}_i,\ i = 1 \ldots l$ нет переменных x_t в различных степенях. Т.е. если $x_t^\sigma \in \mathcal{K}$, то в $\mathcal{K}_i,\ i = 1 \ldots l$ нет сомножителей $x_t^{\overline{\sigma}}$. Действительно, если бы $x_i^{\overline{\sigma}}$ находится в $\mathcal{K}_i$, то $\mathcal{K}\mathcal{K}_i \equiv 0$, $(\mathcal{N}_{\mathcal{K}} \cap \mathcal{N}_{\mathcal{K}_i} = = \oslash)$ и $\mathcal{N}_{\mathcal{K}_i}$ не влиял бы на выполнимость проверяемого соотношения.

Представим $\mathcal{K}_i$ в виде $\mathcal{K}_i^1 \cdot \mathcal{K}_i^2$; в $\mathcal{K}_i^1$ входят все сомножители, общие с $\mathcal{K}$, в $\mathcal{K}_i^2$ — оставшиеся. Если оставшихся нет, полагаем $\mathcal{K}_i^2 = 1$.

Теорема 1 (Критерий поглощения) $\mathcal{N}_{\mathcal{K}} \subseteq \bigcup\limits_{i=1}^{l} \mathcal{N}_{\mathcal{K}_i}$ *тогда и только тогда, когда* $\bigvee\limits_{i=1}^{l} \mathcal{K}_i^2 \equiv \equiv 1$.

Доказательство. *Достаточность.* Пусть $\tilde{\alpha} \in \mathcal{N}_{\mathcal{K}} : \mathcal{K}(\tilde{\alpha}) = 1$. Но тогда очевидно $\mathcal{K}_i^1 = 1,\ i = 1 \ldots l$. Выделим в $\tilde{\alpha}$ поднабор из координат, соответствующих переменным из $\mathcal{K}_i^2,\ i = 1 \ldots l$. Так как $\bigvee\limits_{i=1}^{l} \mathcal{K}_i^2 = 1$, то найдется $\mathcal{K}_u^2,\ 1 \le u \le l$, равное 1 на этом наборе. Но в этом случае $\mathcal{K}_u^1 \cdot \mathcal{K}_u^2(\tilde{\alpha}) = 1$, $\mathcal{K}_u(\tilde{\alpha}) = 1$. Следовательно $\tilde{\alpha} \in \mathcal{N}_{\mathcal{K}_u} \subseteq \bigcup\limits_{i=1}^{l} \mathcal{N}_{\mathcal{K}_i}$. Достаточность доказана.

Необходимость. Пусть $\bigvee\limits_{i=1}^{l} \mathcal{K}_i^2 \not\equiv 1$. Тогда найдется поднабор $\tilde{\beta}$ координат переменных, не входящих в $\mathcal{K}$, такой, что $\mathcal{K}_i^2(\tilde{\beta}) = 0,\ i = 1 \ldots l$.

Пусть $\mathcal{K} = X_{t_1}^{\sigma_1}, \ldots, X_{t_v}^{\sigma_v}$. Сформулируем набор $\tilde{\gamma}$, положив в нем координаты $t_1, \ldots, t_v$, равные, соответственно $\sigma_1, \ldots, \sigma_v$, добавим значения координат из набора $\tilde{\beta}$; остальные координаты зададим произвольно. Тогда $\mathcal{K}(\tilde{\gamma}) = 1, \mathcal{K}_i^1(\tilde{\gamma}) \cdot \mathcal{K}_i^2(\tilde{\gamma}) = 0,\ i = 1 \ldots l$, так $\mathcal{K}_i^2(\tilde{\gamma}) = 0,\ i = 1 \ldots l\ \ (\mathcal{K}_i^2(\tilde{\beta}) = 0$, а $\tilde{\beta}$ — часть набора $\tilde{\gamma})$. Имеем: $\tilde{\gamma} \in \mathcal{N}_{\mathcal{K}}$, $\tilde{\gamma} \overline{\in} \bigcup\limits_{i=1}^{l} \mathcal{N}_{\mathcal{K}_i}$. Необходимость доказана.

Пусть π — преобразование $x_i \to y_j^{\sigma_{ij}}$, $\begin{pmatrix} i \\ j \end{pmatrix}$ — подстановка, $\pi(\mathcal{K})$ — результат преобразования $\mathcal{K}$ с помощью π.

Теорема 2 $\mathcal{N}_{\mathcal{K}} \subseteq \bigcup\limits_{i=1}^{l} \mathcal{N}_{\mathcal{K}_i} \leftrightarrow \mathcal{N}_{\pi(\mathcal{K})} \subseteq \bigcup\limits_{i=1}^{l} \mathcal{N}_{\pi(\mathcal{K}_i)}$

Доказательство. Пусть $\mathcal{N}_{\mathcal{K}} \subseteq \bigcup\limits_{i=1}^{l} \mathcal{N}_{\mathcal{K}_i}$. Тогда $\bigvee\limits_{i=1}^{l} \mathcal{K}_i^2 \equiv 1$ (по т. 1). Но любая подстановка в функцию $f(z_1, \ldots, z_m)$, $z_i = \varphi_i(y_{i_1}, \ldots, y_{i_{k_i}})$ приводит к $f(\varphi_1, \ldots, \varphi_m) \equiv 1$, если $f(z_1, \ldots, z_m) \equiv 1$. Проверка последнего тривиальна. Если $\mathcal{N}_{\mathcal{K}} \not\subseteq \bigcup\limits_{i=1}^{l} \mathcal{N}_{\mathcal{K}_i}$, то $\bigvee\limits_{i=1}^{l} \mathcal{K}_i^2 \not\equiv 1$ (т. 1), и существует набор $\tilde{\beta}$: $\mathcal{K}_i^2(\tilde{\beta}) = 0$, $i = 1 \ldots l$. Это значит, что в каждой $\mathcal{K}_i^2$ есть сомножитель x_r^σ, а r-я координата в $\tilde{\beta}$ равна $\overline{\sigma}$. Если при $\pi : x_r \to y_t$, то в новом наборе t-я координата равна $\overline{\sigma}$, а соответствующий сомножитель: y_t^σ. Очевидно, $\pi(\mathcal{K}_i^2(\tilde{\beta})) = 0$. Случай $x_r \to \overline{y}_t$ разбирается аналогично.

Сказанное выше применимо ко всем $\mathcal{K}_i^2$, имеющим сомножитель x_r^σ. Остальные $\mathcal{K}_u^2$ либо не имеют сомножителя от переменной x_r, либо имеют сомножитель $x_r^{\overline{\sigma}}$. Следовательно, в каждой из этих $\mathcal{K}_u^2$ найдется сомножитель от x_q, $q \neq r$, для которого проходят предыдущие выкладки. Таким образом $\pi(\bigvee\limits_{i=1}^{l} \mathcal{K}_i^2) \not\equiv 1$ и $\mathcal{N}_{\pi(\mathcal{K})} \not\subseteq \bigcup\limits_{i=1}^{l} \mathcal{N}_{\pi(\mathcal{K}_i)}$.

Переходим к реализации I. Докажем сначала:

$$\begin{gathered}(x_1 \vee \ldots \vee x_n) \cdot (\overline{x}_1 \vee \ldots \vee \overline{x}_n) = \\ = x_1 \cdot x_2 \vee x_2 \cdot x_3 \vee \ldots \vee x_i \cdot \overline{x}_{i+1} \vee x_{i+1} \cdot \overline{x}_{i+2} \vee \ldots \vee x_{n-1} \cdot \overline{x} \vee x_n \cdot \overline{x}_1)\end{gathered}$$

Легко видеть, что левая часть реализует функцию, равную на наборах $(0 \ldots 0 \ldots 0)$, $(1 \ldots 1 \ldots 1)$. Действительно, на любом другом наборе либо найдется пара координат i, $i+1$, таких, что $\alpha_i = 1$, $\alpha_{i+1} = 0$ и тогда $x_i \cdot \overline{x}_{i+1} = 1$, либо $\alpha_n = 1$, $\alpha_1 = 0$. И тогда $x_n \cdot \overline{x}_1 = 1$ (рассматривается набор $\tilde{\alpha} = (\alpha_1 \ldots \alpha_n)$.

Заметим, что умножение двух конъюнкций $(x_1^{\sigma_1} \vee \ldots \vee x_n^{\sigma_n}) \cdot (x_1^{\delta_1} \vee \ldots \vee x_n^{\delta_n})$ — произведение реализует функцию, равную 0 только на наборах $(\overline{\sigma}_1 \ldots \overline{\sigma}_n)$, $(\overline{\delta}_1, \ldots, \overline{\delta}_n)$ — с помощью преобразования π можно привести к умножению двух конъюнкций

$$\begin{gathered}(y_1 \vee \ldots \vee y_k \vee \overline{y}_{k+1} \vee \ldots \vee \overline{y}_l \vee y_{l+1} \vee \ldots \vee y_n) \cdot \\ \cdot (y_1 \vee \ldots \vee y_k \vee \overline{y}_{k+1} \vee \ldots \vee \overline{y}_l \vee \overline{y}_{l+1} \vee \ldots \vee \overline{y}_n) = \\ = y_1 \vee \ldots \vee y_k \vee \overline{y}_{k+1} \vee \ldots \vee \overline{y}_l \vee y_{l+1} \cdot \overline{y}_{l+2} \vee \ldots \vee y_{n+1} \cdot \overline{y}_n \vee y_n \cdot \overline{y}_{l+1})\end{gathered}$$

Таким образом, д.н.ф., реализующую функцию с двумя нулями можно построить, использовав только n конъюнкций. Оказывается, что, обобщив приведенные выше построения, можно легко получить д.н.ф. относительно невысокой сложности, если число нулей функции невелико.

Пусть таблица нулей булевой функции имеет вид

$$\begin{gathered}(\alpha_{11} \ldots \alpha_{1i} \ldots \alpha_{1n}) = \tilde{\alpha}_1 \\ (\alpha_{21} \ldots \alpha_{2i} \ldots \alpha_{2n}) = \tilde{\alpha}_2 \\ \ldots \\ (\alpha_{k1} \ldots \alpha_{ki} \ldots \alpha_{kn}) = \tilde{\alpha}_k\end{gathered}$$

Формулы, реализующая функцию, нуля которой суть наборы $\tilde{\alpha}_1, \tilde{\alpha}_2, \cdots \tilde{\alpha}_k$ имеет вид

$$\prod_{i=1}^{k} (x_1^{\alpha_{i1}} \vee x_2^{\alpha_{i2}} \vee \ldots \vee x_n^{\alpha_{in}}) \tag{2.1}$$

Приведем произведение (2.1) к д.н.ф. Сразу исключим из таблицы нулевой и единичный столбцы, так соответствующие им переменные в (2.1) можно сразу вынести за скобки и написать $x_{u_1} \vee \ldots \vee x_{u_p} \vee \overline{x}_{t_1} \vee \ldots \vee \overline{x}_{t_v}$, соответственно, для нулевых и единичных столбцов.

Выполним преобразование $x_i \to x_j^{\sigma_{ij}}$, $\sigma_{ij} \in \{0,1\}$, $\begin{pmatrix} i \\ j \end{pmatrix}$ — подстановка, таким образом, чтобы строка $\tilde{\alpha}_1$ перешла в строку $(0\ 0\ \ldots\ 0\ \ldots\ 0\ 0) = \tilde{0}$, одинаковые столбцы таблицы $\|\alpha_{ij}\|_{k\times n}$ получили последовательные номера, объединившись в блоки, число которых не может превосходить $2^{k-1} - 1$ (столбцы образует $k-1$ строка и нет нулевого столбца).

Пример 2 *Исходная таблица T_1:*

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
0	1	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0	1	0
1	1	1	1	1	1	0	0	0	0

Проведем сначала преобразование $x_1 \to x_1,\ x_2 \to \overline{x_2},\ x_3 \to x_3,\ x_4 \to \overline{x_4},\ x_5 \to x_5,\ x_6 \to \overline{x_6},\ x_7 \to x_7,\ x_8 \to \overline{x_8},\ x_9 \to x_9,\ x_{10} \to \overline{x_{10}}$, а затем номера переменных преобразуем подстановкой

$$\begin{pmatrix} 1 & 3 & 5 & 8 & 10 & 2 & 4 & 6 & 7 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}$$

Получим таблицу:

y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	0	0	0	0

Если рассматривать блок отдельно, то имеем функцию, принимающую значения 0 на нулевом и единичном наборах. Напишем формулу:

$$\varphi = y_1\overline{y}_2 \vee y_2\overline{y}_3 \vee y_3\overline{y}_4 \vee y_4\overline{y}_5 \vee y_5\overline{y}_1 \vee y_6\overline{y}_7 \vee y_7\overline{y}_8 \vee y_8\overline{y}_9 \vee y_9\overline{y}_{10} \vee y_{10}\overline{y}_6$$

Функция φ равно 0 на всех трех строках таблицы, но также имеются и "лишние"нули, например:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$\varphi = 0$ на любом наборе, имеющем одинаковые первые 5 и вторые 5 координат. Чтобы отбросить лишние нули, построим функцию ψ: она зависит от первых переменных каждого из блоков (в нашем случае от y_1 и y_6). Образуем таблицу нулей функции ψ из столбцов значений этих переменных. На остальных наборах $\psi = 1$ (в нашем случае это набор (0 1). Реализуем ψ с помощью д.н.ф.: $\psi = \overline{y}_1 \cdot y_6$.

y_1	y_6	ψ
0	1	0
1	1	0
1	0	0
0	1	1

Формула $\varphi \vee \psi$ реализует функцию, нули которой заданы таблицей T_1: φ равна 0 на любых наборах, которые в каждом из блоков имеют одинаковые значения координат, ψ "отбрасывает"лишние нули, оставляя те, которые реально задаются таблицей.

Теперь достаточно выполнить обратное преобразование, и мы получим формулу для функции, нули которой заданы таблицей T_1. Заметим, что для функции от 10 переменных мы получили д.н.ф. из $10+1$ конъюнкции.

В "наихудшем случае"в преобразованной таблице получится $2^{3-1}-1=3$ блока, и функция ψ примет вид:

y_1	y_{i_1}	y_{i_2}	ψ
0	0	0	1
0	1	1	0
1	0	1	0

На остальных наборах $\psi=1$. Легко написать д.н.ф. для ψ:

$$\psi = \overline{y}_1\overline{y}_{i_1}y_{i_2} \vee y_1\overline{y}_{i_2} \vee \overline{y}_{i_2}y_{i_1} \vee y_1\overline{y}_{i_2}$$

Таким образом для функций с тремя нулями д.н.ф. для ψ не может состоять более, чем из 4 элементарных конъюнкций.

В общем случае, для функций с k нулевыми наборами $\tilde{\alpha}_1,\ldots,\tilde{\alpha}_k$ процесс построения д.н.ф. полностью повторяет действия примера.

Находим преобразование, которое приводит таблицу нулей к виду: 1) первая строка становится нулевой, 2) таблица состоит из блоков одинаковых столбцов (различные блоки состоят из разных столбцов). В каждом блоке присутствуют только строки $\tilde{0}=(0\ \ldots\ 0)$ и $\tilde{1}=(1\ \ldots\ 1)$. Пусть таблица разбита на блоки $B_1,B_2,\ldots,B_i,\ldots,B_q,\ q\le 2^{k-1}-1$. Пусть блок B_i состоит из столбцов значений переменных

$$x_{t_i}, x_{t_i+1}, \ldots, x_{t_i+v_i}$$

Запишем формулу $\mathcal{X}(B_i)=x_{t_i}\cdot\overline{x}_{t_i+1}\vee x_{t_i+1}\cdot\overline{x}_{t_i+2}\vee\ldots\vee x_{t_i+v_i-1}\cdot\overline{x}_{t_i+v_i}\vee x_{t_i+v_i}\cdot\overline{x}_{t_i}$. Д.н.ф. $D=\bigvee\limits_{i=1}^{q}\mathcal{X}(B_i)$ реализует функцию, равную 0 на (и только на) наборах, таких, что значения координат в каждом блоке одинаковы.

Для исключения "лишних"нулей образуем (как в примере) функцию, зависящую от переменных, взятых по одному из каждого блока (например, от первых переменных блока). Соответствующие столбцы образуют множество нулей функции ψ (столбцы значений выбранных переменных). Остальные наборы образуют множество единиц функции ψ, что исключает "лишние"нули и позволяет написать для функции с нулями $\tilde{\alpha}_1,\ldots,\tilde{\alpha}_k$ формулу

$$\bigvee_{i=1}^{q}\mathcal{X}(B_i)\vee\psi$$

Написав д.н.ф. для ψ (ψ зависит не более, чем от $2^{k-1}-1$ переменной), получим искомую д.н.ф. Применив к ней преобразования

$$x_i\mathcal{K}\vee\overline{x}_i\mathcal{K}'\to x_i\mathcal{K}\vee\overline{x}_i\mathcal{K}'\vee\mathcal{K}\mathcal{K}',\quad \mathcal{K}\vee\mathcal{K}\mathcal{K}'\to\mathcal{K}$$

построим сокращенную д.н.ф. с нулями $\tilde{\alpha}^1,\ldots,\tilde{\alpha}^k$.

Замечание. Умножение "тестовых"уравнений приводит к д.н.ф., в которой нет переменных с отрицаниями . Поэтому сокращенная д.н.ф. (совокупность всех простых импликант) получается применением только преобразования

$$\mathcal{K}\vee\mathcal{K}\mathcal{K}'\Rightarrow\mathcal{K}$$

Но $\mathcal{K}\cdot\mathcal{K}'$ не может соответствовать тупиковой тест, так как тест, соответствующих $\mathcal{K}$, получается из него удалением некоторых столбцов. Заметим также, что полученная д.н.ф. является единственной тупиковой (для доказательства применить критерий поглощения и заметить, что д.н.ф., не содержащая отрицаний переменных, не может быть равной 1 на всех наборах значений переменных).

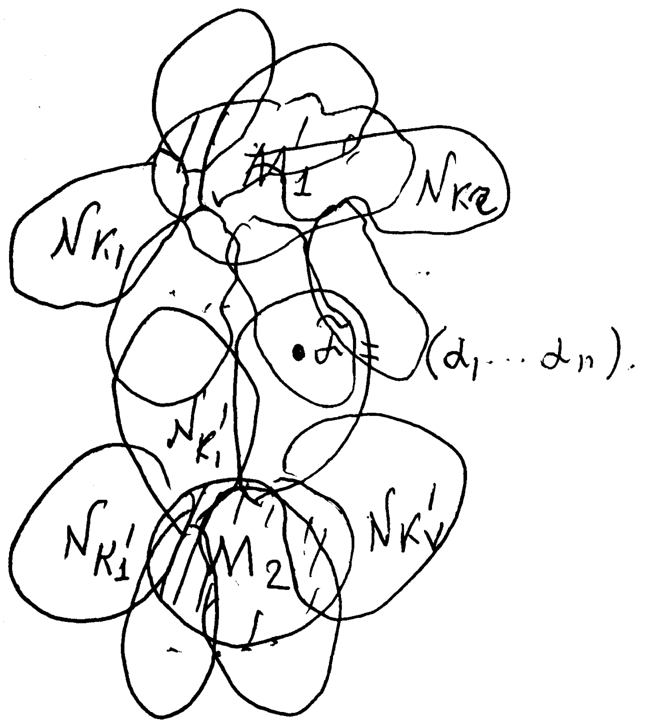
Применение к построению алгоритмов распознавания. Пусть в исходной бинарной таблице наборы $\tilde{\alpha}^i = (\alpha_{i1}, \ldots, \alpha_{in}),\ i = 1 \ldots k$ принадлежат K_1, наборы $\tilde{\beta}^j = (\beta_{j1}, \ldots, \beta_{jn})$, $j = 1 \ldots l$ принадлежат K_2, $K_1 \cap K_2 = \oslash$. Построим семейство логических алгоритмов распознавания. Введем две не всюду определенные булевские функции:

$$F_1(x_1, \ldots, x_n) = \begin{cases} 1 \text{ на наборах } \tilde{\alpha}^i,\ i = 1 \ldots k \\ 0 \text{ на наборах } \tilde{\beta}^j,\ j = 1 \ldots l \end{cases}$$

$$F_2(x_1, \ldots, x_n) = \begin{cases} 1 \text{ на наборах } \tilde{\beta}^j,\ j = 1 \ldots l \\ 0 \text{ на наборах } \tilde{\alpha}^i,\ i = 1 \ldots k \end{cases}$$

С помощью описанной выше процедуры по нулям F_1 и F_2 построим д.н.ф. для функций, равных 0 только на этих наборах, и с помощью преобразований Блейка построим сокращенные д.н.ф этих функций (на практике число добавлеяемых конъюнкций ограничивается). Из сокращенной д.н.ф. удалим все конъюнкции, интервалы которых имеют пустое пересечение с множеством единиц соответствующей функции.

Пусть после этих процедур для F_1 остались интервалы $\mathcal{N}_{\mathcal{K}_1}, \ldots, \mathcal{N}_{\mathcal{K}_r}$, имеющие непустое пересечение с $M_1 = \{\tilde{\alpha}^1 \ldots \tilde{\alpha}^k\}$, и для F_2 — $\mathcal{N}_{\mathcal{K}'_1}, \ldots, \mathcal{N}_{\mathcal{K}'_r}$ с непустым пересечением с $M_2 = \{\tilde{\beta}^1, \ldots, \tilde{\beta}^l\}$. Схематически это можно изобразить так:



Здесь $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ — распознаваемый объект. Построим совокупность тупиковых д.н.ф $D_{11}, \ldots, D_{1p}$ для F_1 и $D_{21}, \ldots, D_{2q}$ — для F_2. При этом можно пользоваться критерием поглощения. Но можно проверить условия

$$M_1 \cap \mathcal{N}_{\mathcal{K}_i} \subseteq \bigcup_{j \neq i} \mathcal{N}_{\mathcal{K}_j} \cap M_1$$

$$M_2 \cap \mathcal{N}'_{\mathcal{K}_i} \subseteq \bigcup_{j \neq i} \mathcal{N}'_{\mathcal{K}_j} \cap M_2$$

и удалить интервалы, пересечение которых с M_1 (M_2) покрывается остающимися интервалами. Для такой проверки существуют более сложный критерий поглощения (см. Ю. И.

Журавлев, "Об отделимости подмножеств вершин единичного n-мерного куба [1]). Пусть построены тупиковые д.н.ф. $T_{11}, \ldots, T_{1K_1}$ для F_1 и $T_{21}, \ldots, T_{2K_2}$ для F_2.

Пусть число выполненных равенств $T_{1i}(\tilde{\alpha}) = 1,\ i = 1 \ldots K_1$ равно Q_1, а выполненных равенств $T_{2i}(\tilde{\alpha}) = 1, i = 1, \ldots K_2$ равно Q_2.

Простейшее решающее правило:

$$Q_1 > Q_2 \to \tilde{\alpha} \in K_1,$$
$$Q_1 < Q_2 \to \tilde{\alpha} \in K_2$$

при $Q_1 = Q_2$ алгоритм отказывается от распознавания.

Возможно усложнение введением порога с

$$Q_1 - c > Q_2 \to \tilde{\alpha} \in \mathcal{K}_1,$$
$$Q_2 - c > Q_1 \to \tilde{\alpha} \in \mathcal{K}_2$$

в остальных случаях алгоритм отказывается от распознавания.

На базе описанного выше алгоритма может быть построено параметрическое семейство — тупиковым д.н.ф. приписаны веса w, и суммируется не число вхождений $\tilde{\alpha}$, а сумма весов тупиковых д.н.ф., интервалы которых содержат точку $\tilde{\alpha}$.

Возможна оптимизация весов по текущему контролю или независимому контрольному материалу (см. подбор значений параметров алгоритмов вычисления оценок).

Лекция 3

3.1 Алгоритмы вычисления оценок

В эвристическом алгоритме "Тест" было проведено сравнение объектов из таблицы обучения и распознаваемого объекта по подмножествам признаков, образующих тупиковые тесты. Очевидно, такие сравнения могут выполняться и по другим подмножествам, например, выбираемым экспертами. При формировании величин $\Gamma(S, K_1), \Gamma(S, K_2)$ добавлялась 1 при выявлении любого совпадения. Однако в реальности признаки и объекты из таблицы обучения не равноценны, и при формировании $\Gamma(S, K_i)$ следует учитывать — с какими объектами по каким признакам произошло совпадение. Это можно делать, задавая различные веса признакам и объектам из таблицы обучения — эталонным объектам. При самом определении близости, если признаки не бинарные, можно также использовать различные возможности, определенные, например, параметрами $\epsilon_1, \dots, \epsilon_n$. Наконец, "поощрения" при формировании $\Gamma(S, K_i)$ возможны также за отсутствие близости к объектам другого класса, а штраф — за наличие близости к объектам класса, для которого не формируется оценка и отсутствие близости к объектам, для которых оценка формируется. Указанные обстоятельства могут реализовываться многими различными способами. Формальное описание таких способов приводит к формированию класса алгоритмов, получившего название "алгоритмы вычисления оценок", или "алгоритмы голосования".

Как и ранее, будем рассматривать исходную (или обучающую) информацию, заданную в виде двух таблиц:
$\|a_{ij}\|_{m\times n}$ — совокупность m объектов, заданных наборами n признаков,
$\|\alpha_{ij}\|_{m\times l}$ — информационная матрица (таблица), где строка $(\alpha_{i1} \dots \alpha_{ij} \dots \alpha_{il}) = \tilde{\alpha}(S_i)$ указывает — каким из классов $K_1, \dots, K_l$ принадлежит или не принадлежит объект S_i, определенный строкой $(a_{i1} \dots a_{in})$ значений признаков $1, 2, \dots, n$ из таблицы обучения. Как и ранее, полагаем, что области определения признаков — это метрические пространства M_t с метриками $\rho_t,\ t = 1, \dots, n$.

Алгоритмы вычисления оценок определяются:

I. Заданием системы опорных множеств признаков.
Это могут быть любые множества, элементами которых являются непустые подмножества множества признаков $1, 2, \dots, n$. Такими могут быть:

a) совокупность всех непустых подмножеств множества $\{1, 2, \dots, n\}$,

b) совокупность всех подмножеств из k элементов и т. д.

В случаях a) и b) имеем, соответственно, $2^n - 1$ и $\binom{n}{k},\ k = 1, 2, \dots, n-1$ подмножеств признаков, по которым происходит сравнение эталонных и распознаваемого объекта.

Вообще говоря, может быть задана любая совокупность $\{\Omega\}_A$ опорных множеств, задающих распознающий алгоритм A. Для удобства, в некоторых случаях, вместо опорного множества $\Omega = \{u_1, \dots u_k\}$ будем рассматривать характеристический вектор $\tilde{\omega} = \{\sigma_1, \dots \sigma_n\}$, где $\sigma_{u_1} = \dots = \sigma_{u_k} = 1$, и остальные координаты равны 0. Очевидно: $\Omega \leftrightarrow \tilde{\omega}, \{\Omega\}_A \leftrightarrow \{\tilde{\omega}\}_A$.

Введем понятия $\tilde{\omega}$-части объекта S и таблицы $\|a_{ij}\|_{m\times n}$:
$\tilde{\omega}$-часть строки $S = (a_1, \dots a_n)$ — обозначение $\tilde{\omega}S$ — это набор

$$\tilde{\omega}S = (a_{u_1}, \dots, a_{u_k}),\ \tilde{\omega}(\|a_{ij}\|_{m\times n}) = \begin{pmatrix} \tilde{\omega}S_1 \\ \vdots \\ \tilde{\omega}S_m \end{pmatrix} = \begin{pmatrix} a_{1u_1} & \dots & a_{1u_k} \\ a_{2u_1} & \dots & a_{2u_k} \\ \dots & \dots & \dots \\ a_{mu_1} & \dots & a_{mu_k} \end{pmatrix}.$$

Будем также использовать обозначение $\tilde{\omega}T_1$, T_1 — таблица обучения.

II. Заданием функции близости $\mathcal{N}(\tilde{\omega}S, \tilde{\omega}S_i)$ между $\tilde{\omega}$-частями распознаваемого объекта S_i и эталонного объекта S. В дальнейшем рассматриваются только функции близости, принимающие значения $0, 1$. Тогда корректно введение функции $\overline{\mathcal{N}}(\tilde{\omega}S, \tilde{\omega}S_i)$, указывающей на отсутствие близости между $\tilde{\omega}S$ и $\tilde{\omega}S_i$.

Обычно рассматриваются три вида функций близости:

1) введем неотрицательные параметры $\epsilon_1, \dots \epsilon_n$. Пусть $\tilde{\omega}S = (a_{u_1}, \dots, a_{u_k})$, $\tilde{\omega}S_i = (a_{iu_1}, \dots, a_{iu_k})$. Тогда

$$\mathcal{N}(\tilde{\omega}S, \tilde{\omega}S_i) = \begin{cases} 1, & \text{если} \quad \begin{matrix} \rho(a_{u_1}, a_{iu_1}) \le \epsilon_{u_1} \\ \dots \\ \rho(a_{u_k}, a_{iu_k}) \le \epsilon_{u_k} \end{matrix} \\ 0, & \text{если хотя бы одно из этих неравенств не выполнено.} \end{cases}$$

2) кроме параметров $\epsilon_1, \dots \epsilon_n$ введем неотрицательный целочисленный параметр ν: $\mathcal{N}(\tilde{\omega}S, \tilde{\omega}S_i) = 1$, если среди приведенных выше неравенств не более ν не выполнены, и равна 0 в остальных случаях.
Пусть $|\Omega|$ — число элементов в опорном множестве $\Omega \in \{\Omega\}_A$; пусть также

$$\min_{\Omega \in \{\Omega\}_A} |\Omega| = q.$$

Легко видеть, что следует рассматривать только значения ν, удовлетворяющие неравенству

$$0 \le \nu \le \left[\frac{q}{2}\right] - 1.$$

3) вместо параметра ν можно рассмотреть параметр ν^* и определить функцию близости следующим образом: $\mathcal{N}(\tilde{\omega}S, \tilde{\omega}S_i) = 1$ тогда и только тогда, когда из k неравенств

$$\begin{matrix} \rho(a_{u_1}, a_{iu_1}) \le \epsilon_{u_1} \\ \dots \\ \rho(a_{u_k}, a_{iu_k}) \le \epsilon_{u_k} \end{matrix}$$

не выполнены r неравенств, и $\frac{r}{k} < \nu^*$.
В практических системах распознавания, в основном, применяется функция близости, зависящая только от параметров $\epsilon_1, \dots \epsilon_n$.

III. Признакам $1, 2, \ldots, n$, описывающим объекты, присваиваются веса $w_1, w_2, \ldots, w_n$. Как правило, $w_i \geq 0,\ i = 1 \ldots\ n$. Но последнее ограничение не является обязательным. Объектам из исходной таблицы $T_1 : S_1, \ldots, S_m$ приписываются веса $w(S_1) = w^1, \ldots, w(S_m) = w^m$. Здесь $w^i \geq 0,\ i = 1 \ldots\ m$. Множеству $\tilde{\omega} S_i = (a_{iu_1}, \ldots, a_{iu_k})$ приписывается вес (число голосов) $\Gamma(\tilde{\omega} S_i) = w^i \cdot (w_{u_1} + \ldots + w_{u_k})$.

IV. При сравнении $\tilde{\omega} S$ и $\tilde{\omega} S_i$ возможны следующие случаи, сведенные в таблицу и оцененные параметрами $x_{ij},\ i = 0, 1,\ j = 0, 1$.

$\mathcal{N}$ / $S_i\ K_j$	$\mathcal{N}(\tilde{\omega} S, \tilde{\omega} S_i) = 1$	$\overline{\mathcal{N}}(\tilde{\omega} S, \tilde{\omega} S_i) = 0$
$S_i \in K_j$	x_{11}	x_{10}
$S_i \notin K_j$	x_{01}	x_{00}

При формировании оценки принадлежности S классу $K_j,\ 1 \leq j \leq l$ (величины $\Gamma_j(\tilde{\omega} S, \tilde{\omega} S_i)$) учитываются оценка $\Gamma(\tilde{\omega} S, \tilde{\omega} S_i)$ и то, какой из четырех указанных случаев имеет место: оценка $\Gamma(\tilde{\omega} S, \tilde{\omega} S_i)$ умножается на соответствующий параметр. Так, если $S_i \in K_j,\ \mathcal{N}(\tilde{\omega} S, \tilde{\omega} S_i) = 1$, то

$$\Gamma_j(\tilde{\omega} S, \tilde{\omega} S_i) = x_{11} \cdot \Gamma(\tilde{\omega} S, \tilde{\omega} S_i).$$

Аналогично формируются оценки в остальных трех случаях. Заметим, что естественно полагать

$$x_{11} \geq 0, \quad x_{00} \geq 0, \quad x_{01} \leq 0, \quad x_{10} \leq 0, \quad x_{00} < x_{11}.$$

Действительно, близость к объекту из K_j или отсутствие близости с объектом, не принадлежащим K_j, благоприятны для оценки вхождения S в K_j, причем второй случай не более благоприятен, чем первый. Два других случая: $S_i \notin K_j,\ \mathcal{N}(\tilde{\omega} S, \tilde{\omega} S_i) = 1$ и $S_i \in K_j,\ \overline{\mathcal{N}}(\tilde{\omega} S, \tilde{\omega} S_i) = 1$ не должны повышать оценку вхождения S в K_j.

Заметим, что приведенные здесь объяснения не являются строгими доказательствами, но лишь эвристическими правдоподобными рассуждениями, оправдывающими (в некоторой степени) даваемое нами определение класса алгоритмов вычисления оценок.

V. Оценка $\Gamma_j(S)$ вхождения объекта S в класс K_j задается следующей формулой

$$\Gamma_j(S) = \frac{1}{Q} \sum_{i=1}^{m} \sum_{\tilde{\omega} \in \{\tilde{\omega}\}_A} \Gamma_j(\tilde{\omega} S, \tilde{\omega} S_i), \tag{3.1}$$

Q — нормирующий множитель. Его величина не влияет на дальнейшие преобразования $\Gamma_j(S)$. Поэтому достаточно рассмотреть случай $Q = 1$.

VI. Решающее правило определяется числовыми параметрами $c_1, c_2,\ 0 < c_1 < c_2$. Алгоритм A формирует для S информационный (квази-информационный) вектор $(\beta_1(S), \ldots, \beta_j(S), \ldots, \beta_l(S))$ следующим образом:
$\beta_j(S) = 1 \to S \in K_j$, если $\Gamma_j(S) > c_2$;
$\beta_j(S) = 0 \to S \notin K_j$, если $\Gamma_j(S) < c_1$;
в остальных случаях $\beta_j(S) = \Delta$, что означает: алгоритм A отказался распознавать вхождение S в класс K_j.

Отметим, что алгоритм вычисления оценок (АВО) подразделяется на две части. После выполнения этапа V формируется числовой вектор оценок $(\Gamma_1(S), \ldots, \Gamma_j(S), \ldots, \Gamma_l(S)) = \overrightarrow{\Gamma_l}(S)$. Эту часть алгоритма принято называть распознающим оператором B:

$$B(I, S) = (\Gamma_1(S), \ldots, \Gamma_j(S), \ldots, \Gamma_l(S)) = \overrightarrow{\Gamma_l}(S).$$

Вторая часть алгоритма (этап VI) переводит $\overrightarrow{\Gamma_l}(S)$ в квази-информационный вектор. Это — решающее правило C:

$$C(\overrightarrow{\Gamma_l}(S)) = C(\Gamma_1(S), \dots, \Gamma_j(S), \dots, \Gamma_l(S)) = (C(\Gamma_1(S)), \dots, C(\Gamma_j(S)), \dots, C(\Gamma_l(S))) = \\ = (\beta_1, \dots, \beta_j, \dots, \beta_l).$$

Резюме. Алгоритм A определяется заданием системы $\{\Omega\}_A$ опорных множеств, параметров $\epsilon_1, \dots \epsilon_n$ (возможно и ν или ν^*), определяющих функцию близости, весов признаков $w_1, \dots, w_n$, весов эталонных объектов $w^1, \dots, w^m$, параметров x_{11}, x_{00}, "поощряющих" благоприятные ситуации, x_{01}, x_{10}, "штрафующих" за неблагоприятные ситуации, параметров c_1, c_2 порогового решающего правила, с помощью которых принимается окончательное решение о вхождении, невхождении распознаваемого объекта S в классы $K_1, \dots K_l$ или, для некоторых классов (может быть, для всех), — об отказе от распознавания.

Пример 3

	1	2	3	4	5	6	K_1	K_2
S_1	0	0	3	2	0.7	0.8	1	0
S_2	0	1	1	4	0.6	0.7	0	1
S_3	1	1	4	1	0.5	0.6	1	1

I

S	1	1	2	3	0.6	0.8

Алгоритм определяется следующими множествами и параметрами:

$$\{\Omega\}_A = \{(1,2), (3,4), (5,6)\};$$
$$\epsilon_1 = \epsilon_2 = 0, \quad \epsilon_3 = \epsilon_4 = 1, \quad \epsilon_5 = \epsilon_6 = 0.1;$$
$$w^1 = w^2 = 2, \quad w^3 = 1;$$
$$w_1 = w_2 = 1, \quad w_3 = w_4 = 2, \quad w_5 = w_6 = 3;$$
$$x_{11} = 3, \quad x_{00} = 1, \quad x_{01} = x_{10} = -1.$$

Таблица значений функции близости

	Ω_1	Ω_2	Ω_3
S_1	0	1	1
S_2	1	0	1
S_3	0	0	0

Таблица использования параметров $x_{ij},\ i = 0, 1;\ j = 0, 1$

x_{10}	x_{11}	x_{11}
x_{01}	x_{00}	x_{01}
x_{01}	x_{01}	x_{01}

Таблица T_Ω оценок $\Gamma_1(\tilde{\omega}S, \tilde{\omega}S_i)$

	Ω_1	Ω_2	Ω_3
S_1	4	8	12
S_2	4	8	12
S_3	2	4	6

Таблица $T_{x_{ij}}$ значений параметров x_{ij}

-1	3	3
-1	1	-1
-1	-1	-1

Умножим поэлементно матрицу T_Ω на матрицу $T_{x_{ij}}$ и сложим все элементы матрицы $T_\Omega \cdot T_{x_{ij}}$. Получим число 36. При $c_2 < 36$ алгоритм зачислит объект S в класс K_1.
Аналогично вычисляется оценка $\Gamma_2(S)$. Вычисление предоставляется выполнить читателю.

3.2 Эффективные формулы вычисления оценок

Прямое использование формулы (3.1) для вычисления оценок $\Gamma_j(S)$, $j = 1, \ldots l$ затруднительно при большом числе множеств. Так, если $\{\Omega\}_A$ состоит из всех непустых подмножеств множества $\{1, 2, \ldots, n\}$, то потребовалось бы вычислить $m \cdot (2^n - 1)$ слагаемых. Поэтому рассмотрим пути сокращения вычислений. Рассмотрим вычисления при фиксированной строке S_i:

$$\sum_{\tilde{\omega} \in \{\tilde{\omega}\}_A} \Gamma_j(\tilde{\omega} S, \tilde{\omega} S_i).$$

Разберем два различных случая: $1^\circ \quad S_i \in K_j$ и $2^\circ \quad S_i \notin K_j$. В случае 1° имеем два подслучая $1^\circ 1$ и $1^\circ 0$: $\mathcal{N}(\tilde{\omega} S, \tilde{\omega} S_i) = 1$ и $\mathcal{N}(\tilde{\omega} S, \tilde{\omega} S_i) = 0$.

$$1^\circ 1: \quad x_{11} \cdot w^j \cdot \sum_{\substack{\Omega:\, \mathcal{N}(\tilde{\omega} S, \tilde{\omega} S_i) = 1 \\ \Omega \in \{\Omega\}_A}} P(\Omega), \quad P(\Omega) = w_{i_1} + \ldots + w_{i_k}, \text{ если } \Omega = \{i_1, \ldots, i_k\}.$$

Обозначим через $R(\mathcal{N} = 1, t)$ число опорных множеств, содержащих t и участвующих в суммировании. Очевидно, величина w_t встретится при суммировании $R(\mathcal{N} = 1, t)$ раз, $t = 1, \ldots, n$. Поэтому формулу для $1^\circ 1$ можно переписать:

$$x_{11} \cdot w^j \cdot \sum_{t=1}^{n} w_t \cdot R(\mathcal{N} = 1, t).$$

В подслучае $1^\circ 0$ вместо множителя x_{11} появится x_{10}, а величина $R(\mathcal{N} = 1, t)$ заменится на $R(\mathcal{N} = 0, t)$, где $R(\mathcal{N} = 0, t)$ — число опорных подмножеств Ω, содержащих t и таких, что $\mathcal{N}(\tilde{\omega} S, \tilde{\omega} S_i) = 0$. Окончательно при $S_i \in K_j$ получаем:

$$w^j (x_{11} \cdot \sum_{t=1}^{n} w_t \cdot R(\mathcal{N} = 1, t) + x_{10} \cdot \sum_{t=1}^{n} w_t \cdot R(\mathcal{N} = 0, t)) = \Gamma_j^1(s_i \in K_j). \tag{3.2}$$

При 2°, действуя в точности как в 1°, выводим:

$$w^j (x_{00} \cdot \sum_{t=1}^{n} w_t \cdot R(\mathcal{N} = 0, t) + x_{10} \cdot \sum_{t=1}^{n} w_t \cdot R(\mathcal{N} = 1, t)) = \Gamma_j^0(s_i \notin K_j). \tag{3.3}$$

Окончательно в формуле

$$\sum_{\Omega \in \{\Omega\}_A} \Gamma_j(\tilde{\omega} S, \tilde{\omega} S_i)$$

суммирование по опорным множествам заменяется на два суммирования n слагаемых и вычисление значений $R(\mathcal{N} = 1, t)$, $R(\mathcal{N} = 0, t)$.
Нами доказана

Теорема 3

$$\Gamma_j(S) = \frac{1}{Q} \left(\sum_{S_i \in K_j} \Gamma_j^1(S_i \in K_j) + \sum_{S_i \notin K_j} \Gamma_j^0(S_i \notin \mathcal{K}_j) \right),$$

где величины Γ_j^1, Γ_j^0 *определяются по формулам* (3.2),(3.3).

1. Пусть система $\{\Omega\}_A$ состоит из всех k-элементных подмножеств множества $\{1,2,\dots,n\}$. Функция близости определяется только параметрами $\epsilon_1,\epsilon_2,\dots,\epsilon_n$. Пусть также $S_i=(a_{i_1},\dots,a_{i_k},\dots,a_{i_n})$, $S=(a_1,\dots,a_k,\dots,a_n)$. Выпишем неравенства

$$\begin{aligned}&\rho_1(a_1,a_{i1})\le\epsilon_1\\&\dots\\&\rho_k(a_k,a_{ik})\le\epsilon_k\\&\dots\\&\rho_n(a_n,a_{in})\le\epsilon_n\end{aligned}$$

   Совокупность номеров признаков, для которых выполнены или, соответственно, не выполнены неравенства, обозначим через M^+, M^-, а мощности соответствующих множеств через $|M^+|$, $|M^-|$.

   Легко видеть, что $R(\mathcal{N}=1,t)$ равно $\binom{|M^+|-1}{k-1}$ для $t\in M^+$, и равно 0 для $t\in M^-$. Так как число подмножеств, содержаших t в $\{\Omega\}_A$, равно $\binom{n-1}{k-1}$, то

$$R(\mathcal{N}=0,t)=\begin{cases}\binom{n-1}{k-1}-\binom{|M^+|-1}{k-1}, & \text{для } t\in M^+\\ \binom{n-1}{k-1}, & \text{для } t\in M^-.\end{cases}$$

   Из последнего видно, что $\sum\limits_{t=1}^{n}w_t\cdot R(\mathcal{N}=1,t)$, $\sum\limits_{t=1}^{n}w_t\cdot R(\mathcal{N}=0,t)$ в рассматриваемом случае заменится более “простыми” суммами, так как величины $R(\mathcal{N}=1,t)$, $R(\mathcal{N}=0,t)$ для каждой строки S_i принимают только два различных значения. Так,

$$\sum_{t=1}^{n}w_t\cdot R(\mathcal{N}=1,t)=\left(\sum_{t\in M^+}P_t\right)\cdot\binom{|M^+|-1}{k-1}.$$

   Аналогично упрощаются и другие введенные ранее формулы.

2. Рассмотрим ту же, что и в 1, систему опорных множеств и функцию близости с параметрами $\epsilon_1,\dots,\epsilon_n$, ν, $\nu<[\frac{k}{2}]-1$. Аналогично предыдущему пункту, при сравнении строк S и S_i образуем подмножества признаков-координат M^+ и M^-.

   Пусть $t\in M^+$. Тогда функция близости равна 1, если она содержит $0,1,2,\dots,\min(\nu,|M^-|)$ признаков из M^- и, соответственно, $k-1,k-2,\dots,k-\min(\nu,|M^-|)-$ -1 признаков из M^+ (для упрощения выкладок мы рассматриваем только случай $k\ge\min(\nu,|M^-|)+1$). Тогда число опорных подмножеств, содержащих t и таких, что $\mathcal{N}(\tilde\omega S,\tilde\omega S_i)=1$, очевидно, равно

$$\binom{|M^+|-1}{k-1}\cdot\binom{|M^-|}{0}+\binom{|M^+|-1}{k-2}\cdot\binom{|M^-|}{1}+\ldots+\binom{|M^+|-r}{k-1-r}\cdot\binom{|M^-|}{r}+\ldots+$$
$$+\binom{|M^+|-1}{k-1-\nu}\cdot\binom{|M^-|}{\nu}.$$

   Все t из M^+ имеют одинаковый только что выписанный коэффициент при P_t. Остальные случаи вычисляются так же просто.

3. В качестве опорных рассматриваются все непустые подмножества множества $\{1, 2, \ldots, n\}$. Тогда

$$R(\mathcal{N}=1,t)=\begin{cases}2^{|M^+|-1}, & \text{для } t\in M^+\\ 0, & \text{для } t\in M^-.\end{cases}$$

$$R(\mathcal{N}=0,t)=\begin{cases}2^{|M^+|-1}\cdot(2^{|M^-|}-1), & \text{для } t\in M^+\\ 2^{n-1}, & \text{для } t\in M^-.\end{cases}$$

Подставляя полученные значения в (3.2), получаем

$$w^i\Bigg(x_{11}\cdot\Bigg(\sum_{t\in M^+} w_t\Bigg)\cdot 2^{|M^+|-1}+x_{10}\Bigg(\Bigg(\sum_{t\in M^+} w_t\Bigg)\cdot(2^{|M^+|-1})\cdot(2^{|M^-|}-1)+ +\Bigg(\sum_{t\in M^-} w_t\Bigg)\cdot 2^{n-1}\Bigg)\Bigg)$$

Аналогичный вид принимает после соответствующих подстановок формула (3.3).

4. Совокупность характеристических векторов опорных множеств образует интервал в E^n, т.е. удовлетворяет условию: $\mathcal{K}=1$, $\mathcal{K}$ — элементарная конъюнкция. Не ограничивая общности, можно полагать

$$\mathcal{K}=\overline{x_1}\cdot\ldots\cdot\overline{x_r}\cdot x_{r+1}\cdot\ldots\cdot x_{r+k}.$$

Тогда в систему опорных множеств не войдут признаки $1,\ldots,r$; в каждое из опорных множеств войдут признаки $r+1,\ldots,r+k$, и к ним последовательно присоединятся все подмножества (включая пустое, если $\mathcal{K}\neq 0$) или все непустые подмножества множества $r+k+1,\ldots,n$, если $\mathcal{K}=0$. Будем рассматривать $\mathcal{K}\neq 0$ и рассмотрим функцию близости с параметрами $\epsilon_1,\ldots,\epsilon_n$.

Если не выполнено включение $\{r+1,\ldots,r+k\}\subseteq M^+$, то $R(\mathcal{N}=1,t)=0$, $R(\mathcal{N}=0,t)=2^{n-(r+k+1)}$. Последнее следует из того, что $|\{\Omega\}_A|=2^{n-(r+k)}$, и хотя бы один из признаков любого опорного подмножества принадлежит M^-.

Пусть имеет место: $\{r+1,\ldots,r+k\}\subseteq M^+$. Тогда для $t\in\{r+1,\ldots,r+k\}$:

$$R(\mathcal{N}=1,t)=2^{|M^+|-k},$$
$$R(\mathcal{N}=0,t)=2^{|M^-|}-1.$$

Для $t\in M^+\backslash\{r+1,\ldots,r+k\}$:

$$R(\mathcal{N}=1,t)=2^{|M^+|-k-1},$$
$$R(\mathcal{N}=0,t)=2^{|M^+|-k-1}\cdot(2^{n-(k+r)}-1).$$

Наконец, для $t\in M^-$: $R(\mathcal{N}=1,t)=0$, $R(\mathcal{N}=0,t)=2^{|M^-|-1}$.

После соответствующих подстановок, получаем эффективные формулы вычисления $\Gamma_j(S),\ j=1,\ldots,l$.

Последний случай (4) можно использовать при решении прикладных задач. Рассмотрим булевскую функцию, равную 1 на элементах $\{\tilde{\omega}\}_A$. Если реализовать ее дизъюнктивной нормальной формой $\bigvee\limits_{i=1}^{r}\mathcal{K}_i$, где $\mathcal{K}_i$ — элементарные конъюнкции, и

$\mathcal{N}_{\mathcal{K}_u} \bigcap \mathcal{N}_{\mathcal{K}_v} = \varnothing$, то, написав по 4 формулы для каждой $\mathcal{K}_i$ и сложив их, получим формулу для вычисления $\Gamma_j(S),\ j = 1, \ldots, l$. Достаточно и реализации в классе д.н.ф., в которых интервалы некоторых пар конъюнкций пересекаются (интервалы конъюнкций из разных пар не пересекаются). Пусть оценка по системе опорных множеств, задаваемых конъюнкцией $\mathcal{K}$, есть $\Gamma_j(\mathcal{K}, S)$. Пусть, также, $\{\tilde{\omega}\}_A = \mathcal{K}_1 \bigcup \mathcal{K}_2,\ \mathcal{K}_1 \cdot \mathcal{K}_2 \not\equiv 0$. Тогда, очевидно, $\Gamma_j(S) = \Gamma_j(S, \mathcal{K}_1) + \Gamma_j(S, \mathcal{K}_2) - \Gamma_j(S, \mathcal{K}_1 \cdot \mathcal{K}_2)$.

Лекция 4

4.1 Вычисление характеристик, определяющих алгоритм вычисления оценок

Как было показано в лекции 3, алгоритм распознавания определяется заданием системы опорных множеств $\{\Omega\}_A$ и числовых параметров $\epsilon_1, \ldots, \epsilon_n$, $w_1, \ldots, w_n$, $w^1, \ldots, w^m$, x_{11}, x_{00}, x_{01}, x_{10}, c_1, c_2.

В своей работе алгоритм использует исходную (обучающую) информацию, состоящую из таблицы обучения $\|a_{ij}\|_{m\times n}$ и ее информационной матрицы $\|\alpha_{ij}\|_{m\times l}$. Рассматривается задача с l, вообще говоря, пересекающимися классами $K_1, \ldots, K_l$.

Параметры подбираются таким образом, чтобы обеспечить максимальную точность распознавания на определенном заранее множестве объектов.

I. Текущий контроль. Из исходной матрицы последовательно изымаются строки S_i, $i = 1 \ldots m$, вместе с информационным вектором $\tilde{\alpha}(S_i)$, и для строки S_i по оставшейся исходной информации строится квази-информационный вектор $\tilde{\beta}(S_i) = (\beta_{i1} \ldots \beta_{il})$, $i = 1 \ldots m$

   В матрице $\|\alpha_{ij} \dot{-} \beta_{ij}\|_{m\times l}$ определяется число единиц. Операция $\alpha_{ij} \dot{-} \beta_{ij}$ определяется следующим образом:

$\alpha_{ij} \backslash \beta_{ij}$	0	1	Δ
0	0	1	1
1	1	0	1

   Алгоритм A подбирается таким образом, чтобы число единиц (т.е. сумма ошибок и отказов) была бы минимальной.

II. Независимый контроль. формируется контрольное множество $S^1, \ldots, S^q$, $S_i = (b_{i1} \ldots b_{in})$, $i = 1 \ldots q$ и таблицы информационных векторов $\|\beta_{ij}\|_{q\times l}$. С использованием алгоритма A и всей исходной информации формируется совокупность $\tilde{\gamma} = (\gamma_{i1} \ldots \gamma_{il})$ квази-информационных векторов и минимизируется число единиц в матрице $\|\beta_{ij} \dot{-} \gamma_{ij}\|_{q\times l}$.

   1) Система опорных множеств определяется (задается) экспертами и последовательно рассматриваются алгоритмы в наборе всех k-элементарных подмножеств: $k = 2, 3, \ldots, r$. Как правило, достаточно ограничиться $r \leq 3\sqrt{n}$.

   2) Параметры x_{ij} определяются перебором вариантов. Как правило, рассматриваются целочисленные значения $x_{11} = 1, 2, 3, 4, 5, 6, 7$; $x_{00} \leq [\frac{1}{2}x_{11}]$, x_{01}, x_{10} принимают отрицательные или нулевые значения. В большинстве действующих систем полагают: $x_{11} = 1$, $x_{00} = x_{01} = x_{10} = 0$.

3) Пусть определены все характеристики за исключением $\epsilon_1, \ldots, \epsilon_n$ (о них позднее). Рассмотрим задачу с независимым контролем и напишем систему неравенств

$$\begin{aligned}
&\Gamma_j(S^i) > c_2, \forall \beta_{ij} = 1 \\
&\Gamma_j(S^i) < c_1, \forall \beta_{ij} = 0 \\
&i = 1, 2, \ldots, q, j = 1, \ldots, l
\end{aligned} \tag{4.1}$$

   В левой части системы неравенств (4.1) находятся билинейные формы $\Sigma w_u \cdot w^v$. Полагаем $w_1 = \ldots = w_n = 1$. Получаем систему линейных относительно $w^1 \ldots w^q$ неравенств. Находим максимальную совместную подсистему и ее решение $w_1^1, \ldots, w_1^q$. Подставляем эти значения в левую часть (4.1) и получаем линейную относительно $w_1, \ldots, w_n$ систему. Находим совместную максимальную подсистему и ее решение $w_{11}, \ldots, w_{n1}$. Подставляем эти значения левую часть (4.1) и т.д. Процесс заканчивается либо когда удастся получить наборы параметров, удовлетворяющие всем неравенствам (4.1), либо когда после очередной итерации число неравенств в совместной подсистеме уменьшится.

4) Для определения $\epsilon_1, \ldots, \epsilon_n$ существует большое число эвристических методов. Приведем один из них (может быть, не лучший). Оставим в таблице обучения и контроля только k-е столбцы:

$$\begin{pmatrix} a_{1k} \\ a_{2k} \\ \vdots \\ a_{mk} \end{pmatrix}, \begin{pmatrix} b_{1k} \\ b_{2k} \\ \vdots \\ b_{lk} \end{pmatrix}$$

   Для каждой пары $(a_{ik} b_{jk})$ вычислим $\|(\tilde{\alpha}(S_i) + \tilde{\beta}(S^j))_{mod2}\|$, то есть число различных координат в этом векторе.
   Если $l - \|\tilde{\alpha}(S_i) + \tilde{\beta}(S^j)\| > \|\tilde{\alpha}(S_i) + \tilde{\beta}(S^j)\|$, то формируем неравенство

$$|a_{ik} - b_{jk}| < \epsilon_k$$

   При изменении знака:

$$|a_{ik} - b_{jk}| > \epsilon_k.$$

   При $l - \|\tilde{\alpha}(S_i) + \tilde{\beta}(S^j)\| = \|\tilde{\alpha}(S_i) + \tilde{\beta}(S^j)\|$ выписываем одно из неравенств

$$|a_{ik} - b_{jk}| \leq \epsilon_k, \qquad |a_{ik} - b_{jk}| \geq \epsilon_k.$$

   Значение ϵ_k находится из условия: данное значение удовлетворяет наибольшему числу построенных неравенств.

4.2 Алгебры над алгоритмами

Ранее мы видели, что алгоритм вычисления оценок делится на две части: распознающий оператор B и решающее правило C: $A = B \cdot C$. $A(I, S) = (\Gamma_1(S), \ldots, \Gamma_l(S)) = \vec{\Gamma}_l(S)$, $C(\vec{\Gamma}_l(S)) = (C(\Gamma_1(S)), \ldots, C(\Gamma_l(S)))$

$$C(\Gamma_j(S)) = \begin{cases} 1, & \Gamma_j(S) > c_2 \\ 0, & \Gamma_j(S) < c_1 \qquad c_1 < c_2 \\ \Delta, & c_1 \le \Gamma_j(S) \le c_2 \end{cases}$$

Оказывается, что подобное представление имеет место для большого класса алгоритмов.

Пусть A — алгоритм, работающий с исходной информацией $I \in \{I\}$, $A \in \{A\}$, и каждый из A по любой $I \in \{I\}$ должен получить ответ на фиксированные вопросы $Q_1, \ldots Q_l$, причем число возможных ответов равно трем: 1, 0, Δ. Тогда

Теорема 4 *Каждый A может быть представлен в виде $A = B \cdot C$, $B(I) = (a_1 \ldots a_j \ldots a_l)$ — числовой вектор $\vec{a}$, $C(\vec{a}) = (C(a_1), \ldots, C(a_l))$, причем*

$$C(a_i) = \begin{cases} 1, & a_i > c_2 \\ 0, & a_i < c_1 \\ \Delta, & c_1 \le a_i \le c_2 \end{cases}$$

где c_1 и c_2 — константы, фиксированные для всех алгоритмов. Заметим, что $A(I) = (C(a_1), \ldots, C(a_l)) = (\delta_1 \ldots \delta_l) = \vec{\delta}$

Доказательство. Введем вспомогательный оператор $C^{-1}(\vec{\delta}) = (C^{-1}(\delta_1) \ldots C^{-1}(\delta_l)) = (a_1, \ldots, a_l) = \tilde{a}$. Тогда $B = A \cdot C^{-1}$, $A = (A \cdot C^{-1}) \cdot C$. Теорема доказана.

Класс алгоритмов $\{A\}$ порождает класс операторов $\{B\}$, которые можно складывать, умножать, умножать на число.

Действительно, если $B_1(I) = (a_{11} \ldots a_{1l})$, $B_2(I) = (a_{21} \ldots a_{2l})$, то $(B_1 + B_2)(I) = (a_{11} + a_{21}, \ldots, a_{1l} + a_{2l})$, $B_1 \cdot B_2(I) = (a_{11} \cdot a_{21}, \ldots, a_{1l} \cdot a_{2l})$, $(d \cdot B_1)(I) = (d \cdot a_{11}, \ldots, d \cdot a_{1l})$.

Нетрудно видеть, что используя операторы из $\{B\}$, $A = B \cdot C$, можно построить полиномы

$$\tilde{B} = \Sigma c_{i_1 \ldots i_k} \cdot B_{i_1}^{r_1} \cdot \ldots \cdot B_{i_k}^{r_k},$$

где роль переменных играют операторы; $c_{i_1 \ldots i_k}$ — константы.

Совокупность $\{\tilde{B}\}$ называют алгебраическим замыканием семейства $\{B\}$, а $\{\tilde{B}\} \cdot C$ — алгебраическим замыканием класса алгоритмов $\{A\}$ — обозначения $\mathfrak{U}\{B\}$, $\mathfrak{U}\{A\}$.

Оказывается (Ю. И. Журавлев), что в $\mathfrak{U}\{A\}$ при выполнении простых легко проверяемых условий можно построить алгоритм, не делающий ошибок на контрольной совокупности.

Алгоритм имеет вид:

$$(d \cdot \Sigma(c_i B_i)^{k_i}) \cdot C.$$

B_i представимы линейными формами от распознающих операторов вычисления оценок.

Условия:

Пусть $I = \{\|a_{ij}\|_{m \times n} \|\alpha_{ij}\|_{m \times l}\}$.

Контрольный материал: $\|b_{uv}\|_{q \times n}$, $\|\beta_{ij}\|_{q \times l}$.

1. в матрице $\|\alpha_{ij}\|_{m\times l}$ нет одинаковых столбцов.

2. для каждой пары S^u, S^v контрольных объектов, $S^u = (b_{u1}\dots b_{un})$, $S^v = (b_{v1}\dots b_{vn})$ найдется $S_r \in I$, $S_r = (a_{r1}\dots a_{rl})$ и признак k, $1 \le k \le n$, $r = r(u,v)$, $k = k(u,v)$ такие, что

$$\rho_k(a_{rk}, b_{uk}) \neq \rho_k(a_{rk}, b_{vk})$$

Лекция 5

5.1 Построение алгоритмов распознавания, корректных для заданной контрольной выборки

Рассматривается задача распознавания (или прогноза) со стандартной обучающей информацией:

$$
\begin{aligned}
&I_0 = \{S_1, \dots, S_m, \tilde{\alpha}(S_1), \dots, \tilde{\alpha}(S_m)\},\\
&S_i = (a_{i1}, \dots, a_{in}), \qquad i = 1, 2, \dots, m,\\
&a_{ij} \in M_j, \qquad j = 1, 2, \dots, n.
\end{aligned}
$$

Здесь $S_1, \dots, S_m$ — описания объектов, составляющих обучающий материал, $\tilde{\alpha}(S_i)$, $i = 1, 2, \dots, m$, — информационные векторы объектов S_i по свойствам $P_j \equiv S_i \in K_j$. Другими словами, если $\alpha(S_i) = (\alpha_{i1}, \dots, \alpha_{ij}, \alpha_{il})$, $j = 1, 2, \dots, l$, то

$$
\alpha_{ij} = \begin{cases} 1, & S_i \in K_j,\\ 0, & S_i \bar{\in} K_j. \end{cases}
$$

Задача распознавания Z определяется начальной информацией I_0 и конечной выборкой $\tilde{S}^q = (S^1, \dots, S^q)$, $S^i = (b_{i1}, \dots, b_{in})$, $i = 1, 2, \dots, q$, т.е. $Z = \{I_0, \tilde{S}^q\}$.

Требуется для каждого объекта S^i из $\tilde{S}^q$ вычислить его информационный вектор $\tilde{\beta}(S^i)$ или, что то же самое, значение свойства $P_j(S^i) \equiv S^i \in K_j$, $j = 1, 2, \dots, l$.

Далее будем считать, что информационные векторы для $\tilde{S}^q$ известны, и, основываясь на этом, строить алгоритм, который правильно вычисляет эти свойства.

I. Пусть дано множество $\{A\}$, вообще говоря, некорректных алгоритмов для решения задач распознавания, представленных в виде $A = B \cdot C$, где B — распознающий оператор, C - решающее правило. Напомним, что $B(I_0, S^q) = \|\Gamma_{ij}\|_{q \times l}$. Здесь Γ_{ij} — действительные числа, $C(\|\Gamma_{ij}\|)_{q \times l} = \|\beta_{ij}\|_{q \times l}$; $\beta_{ij} \in \{0, 1, \Delta\}$, $\beta_{ij} = \Delta$ означает, что алгоритм A отказался от вычисления свойства $P_j(S^i)$; $\beta_{ij} = \beta$, $\beta \in \{0, 1\}$, означает, что алгоритм A вычислил свойство $P_j(S^i)$ равным β. При этом мы допускаем возможность ошибки.

   Известно, что каждый алгоритм A может быть представлен в виде $B \cdot C$ и что по исходному семейству $\{A\}$ с помощью операций сложения, умножения и умножения на скаляр можно построить алгебраическое расширение

$$
\mathfrak{U}\{A\} = \mathfrak{U}\{B\} \cdot \{C\}
$$

   класса алгоритмов $\{A\}$ и алгебраические расширения конечных степеней

$$\mathfrak{U}^k\{A\} = \mathfrak{U}^k\{B\} \cdot \{C\}.$$

При выполнении некоторых условий для задачи Z и исходного семейства $\{A\}$ в расширении $\mathfrak{U}^k\{A\}$ можно построить алгоритм A^*, правильно вычисляющий все значения $P_j(S^i)$, $i = 1, 2, \ldots, q$, $j = 1, 2, \ldots, l$. Если в качестве исходного семейства $\{A\}$ рассмотреть класс алгоритмов вычисления оценок, то искомый алгоритм A^* представим в виде

$$A^* = \left[(c_1 + c_2) \sum_{i=1}^{q} \sum_{j=1}^{l} \beta_{ij} (B_{ij})^k \right] C(c_1, c_2),$$

где c_1, c_2 — параметры решающего правила C,

$$\begin{aligned}
&\beta_{ij} P_j(S^i), \qquad i = 1, 2, \ldots, q, \qquad j = 1, 2, \ldots, l,\\
&B_{ij} = B_j + B_i^j,\\
&B_j = B_{j1} + \ldots + B_{j,j-1} + B_{j,j+1} + \ldots + B_{jl},\\
&B_i^j = B_{i1}^j + \ldots + B_{i,i-1}^j + B_{i,i+1}^j + \ldots + B_{iq}^j;
\end{aligned}$$

здесь каждый оператор B_{ij} является оператором вычисления оценок, каждый оператор B_{iv}^j является либо оператором вычисления оценок, либо разностью двух операторов вычисления оценок. Для величины k также имеется формула, приводить которую здесь нет необходимости.

Каждый оператор вычисления оценок кодируется значениями $2n + 3m + 3$ параметров, где n — число признаков, описывающих объекты, m — число объектов в I_0. Нетрудно также видеть, что приведенная выше формула для алгоритма A^* включает в себя по крайней мере $l(l-1) + lq(q-1)$ операторов вычисления оценок. Следовательно, для полной записи кода алгоритма A^* требуется по крайней мере $(2n + m + 3)l[(l-1) + q(q-1)]$ чисел. Указанная величина может быть несколько уменьшена с помощью специальных приемов, однако она все-таки остается большой, и это неудобно при машинной реализации алгоритма, если величины n, m, l, q велики. Поэтому при реальном синтезе корректного алгоритма A^* будет использоваться только его принципиальная запись, данная выше, а реализация операторов типа B_{ij} будет проводиться другими методами. В дальнейшем будут использоваться только пороговые решающие правила $C(c_1, c_2)$: $C(\|\Gamma_{ij}\|_{q\times l}) = \|C(\Gamma_{ij})\|_{q\times l}$,

$$C(\Gamma_{ij}) = \begin{cases} 1, & \Gamma_{ij} > c_2,\\ 0, & \Gamma ij < c_1, \qquad 0 < c_1 < c_2.\\ \Delta, & c_1 \le \Gamma_{ij} \le c_2 \end{cases}$$

II. Рассмотрим информационную матрицу $\|\beta_{ij}\|_{q\times l}$ выборки $\tilde{S}^q$ в задаче Z. Положим

$$\begin{aligned}
&M = \{(i,j)\}, \qquad i = 1, 2, \ldots, q, \qquad j = 1, 2, \ldots, l,\\
&M_\alpha = \{(i,j)\} : \beta_{ij} = \alpha, \qquad \alpha = 0, 1.
\end{aligned}$$

Очевидно, $M = M_0 \cup M_1$.

Пусть B — распознающий оператор и

$$B(Z) = \|\Gamma_{rt}(B)\|_{q\times l},$$

где Γ_{rt} — действительные числа.

Определение 1 *Оператор B называется допустимым для задачи Z, если существует хотя бы одна пара (u, v) из M_1 такая, что для всех (i, j) из M_0*

$$\Gamma_{uv}(B) > |\Gamma_{ij}(B)|.$$

Пара (u, v) называется в этом случае отмеченной в B. Совокупность всех пар, отмеченных в B, обозначим через $M(B)$.

Пусть

$$\Gamma^0_{max}(B) = \max_{(i,j)\in M_0} |\Gamma_{ij}(B)|,$$
$$\Gamma^1_{min}(B) = \min_{(i,j)\in M(B)} \Gamma_{ij}(B).$$

Положим

$$\Gamma(B) = [\Gamma^1_{min}(B)]^{-1}. \tag{5.1}$$

По оператору B построим оператор B':

$$B' = \Gamma(B)\cdot(B). \tag{5.2}$$

Пусть $B'(Z) = \|\Gamma'_{ij}\|_{q\times l}$. Тогда из (5.2) легко следует, что

$$\Gamma'_{ij} = \Gamma(B)\cdot\Gamma_{ij}(B). \tag{5.3}$$

Лемма 1 *Если (u, v) отмечена в B, то $\Gamma'_{uv}(B) \geq 1$; если (u, v) не отмечена в B, то*

$$\Gamma'_{uv}(B) \leq \frac{\Gamma^0_{max}(B)}{\Gamma^1_{min}(B)} = \Gamma(B) < 1.$$

Доказательство. Если (u, v) отмечена в B, то для $\Gamma_{uv}(B)$ выполнено неравенство

$$\Gamma_{uv}(B) \geq \min_{(i,j)\in M(B)} \Gamma_{ij}(B) = \Gamma^1_{min}(B).$$

Из этого неравенства и соотношений (5.1)-(5.3) легко следует первое утверждение леммы.

Если пара (u, v) не является отмеченной в B, то

$$|\Gamma_{uv}(B)| \leq \max_{(i,j)\in M_0} \Gamma_{ij}(B) = \Gamma^0_{max} < \min_{(i,j)\in M(B)} \Gamma_{ij}(B) = \Gamma^1_{min}(B).$$

Из последних неравенств и соотношений (5.1)-(5.3) легко следует второе утверждение леммы.

В дальнейшем положим

$$\Gamma^0_{max}(B)/\Gamma^1_{min}(B) = Q(B).$$

Пусть $\{B\}$ — произвольная конечная система распознающих операторов.

Определение 2 *Система $\{B\}$ называется базисной для Z, если*

$$M_1 = \bigcup_{B\in\{B\}} M(B).$$

По базисной для Z системе $\{B\}$ построим алгоритм A^*, корректный для задачи Z.

Введем для операторов B', $B' = \Gamma(B)\cdot B$, $B \in \{B\}$ целые числа $k(B')$ так, чтобы выполнялось неравенство

$$[Q(B)]^{k(B')} < \frac{c_1}{(c_1+c_2)|\{B\}|}. \tag{5.4}$$

Для этого достаточно положить

$$k(B') = \frac{\ln(c_1+c_2) + \ln|\{B\}| - \ln c_1}{|\ln Q(B)|} + 1.$$

Очевидно, что в этом случае неравенство (5.4) выполнено. Напомним, что величины c_1, c_2 суть пороги решающего правила $C(c_1, c_2)$.

Теорема 5 *Алгоритм*

$$A = \left\{(c_1+c_2)\sum_{B\in\{B\}} (B')^{k(B')}\right\}\cdot C(c_1, c_2)$$

является корректным для Z.

Доказательство. Распознающим оператором B^* в алгоритме A^* является оператор

$$(c_1+c_2)\sum_{B\in\{B\}} (B')^{k(B')}.$$

Пусть $B^*(Z) = \|\Gamma^*_{ij}\|_{q\times l}$. Тогда

$$\Gamma^*_{ij} = (c_1+c_2)\sum_{B\in\{B\}} (\Gamma'_{ij}(B))^{k(B')}.$$

Случай 1. $P_j(S^i) = \beta_{ij} = 1$, $1 \le i \le q$, $1 \le j \le l$,

$$\Gamma^*_{ij} = (c_1 + c_2)\left[\sum_{B \in B(i,j)} (\Gamma'_{ij}(B))^{k(B')} + \sum_{B \bar{\in} B(i,j)} (\Gamma'_{ij}(B))^{k(B')}\right].$$

Здесь $B(i,j)$ — совокупность всех операторов из $\{B\}$, в которых пара (i,j) является отмеченной. Так как $\{B\}$ — базисная система для Z, то множество $B(i,j)$ непусто. Поэтому

$$\sum_{B \in B(i,j)} (\Gamma'_{ij}(B))^{k(B')} \ge 1, \tag{5.5}$$

$$\left|\sum_{B \bar{\in} B(i,j)} (\Gamma'_{ij}(B))^{k(B')}\right| < |\{B\}| \frac{c_1}{(c_1 + c_2)|\{B\}|}. \tag{5.6}$$

Неравенство (5.5) следует из определения отмеченной пары, неравенство (5.6) — из (5.4). Окончательно получаем

$$\Gamma^*_{ij} > (c_1 + c_2)\left(1 - \frac{c_1}{c_1 + c_2}\right) = c_2.$$

Но тогда из определения порогового решающего правила следует $C(\Gamma^*_{ij}) = 1 = \beta_{ij} = p_j(S^j)$.

Случай 2. $p_j(S^i) = \beta_{ij} = 0$, $0 \le i \le q$, $1 \le j \le l$. В этом случае пара (i,j) не является отмеченной ни в одном операторе B. Поэтому

$$\Gamma^*_{ij} = (c_1 + c_2) \sum_{B \bar{\in} B(i,j)} (\Gamma'_{ij}(B))^{k(B')} < (c_1 + c_2)|\{B\}| \times \frac{c_1}{(c_1 + c_2)|\{B\}|} = c_1.$$

Из определения $C(c_1, c_2)$ следует

$$C(\Gamma^*_{ij}) = 0 = \beta_{ij} = P_j(S^i).$$

Теорема доказана.

Определение 3 *Базисная система $\{B\}$ называется неприводимой для Z, если никакая собственная часть $\{B\}$ не является базисной для Z.*

Очевидно,

$$|\{B\}| \le |M_1| \le q \times l.$$

Поэтому неравенство (5.4) для неприводимых систем записывается в виде

$$Q(B)^{k(B')} < \frac{c_1}{(c_1 + c_2)|M_1|}.$$

Этому неравенству удовлетворяет

$$k(B') = \frac{\ln|M_1| + \ln(c_1 + c_2) - \ln c_1}{\ln Q(B)} + 1.$$

Формулировка теоремы 1 при новых $k(B')$ сохранится без изменений.

Из теоремы 1 следует, что для построения эффективного корректного алгоритма достаточно построить систему из небольшого числа просто выполняемых распознающих операторов B, базисную для Z. Из базисной системы затем нетрудно получить неприводимую систему. Для построения базисной системы подходит любой оператор, отмечающий непустое множество пар (i, j) таких, что $\beta_{ij} = 1$. В [X] для любой задачи $Z = \{I_0, \tilde{S}^q\}$, $I_0 = \{S_1, \ldots, S_m, \tilde{\alpha}(S_1), \ldots, \tilde{\alpha}(S_m)\}$, в которой объекты из $\tilde{S}^q$ попарно неизоморфны относительно I_0 и информационная матрица $\|\alpha_{ij}\|_{m\times l}$ в I_0 состоит из попарно различны столбцов, указана базисная система из $|M_1|$ операторов. Каждый оператор этой базисной системы гарантирует отметку, вообще говоря, ровно одной пары (i, j), $\beta_{ij} = 1$. Можно указать случаи, когда один оператор гарантирует отметку существенно большего числа пар. Это возможно и для простых распознающих операторов.

Литература

[1] Труды Мат. ин-та им. В.А.Стеклова, том 51, 1958 г.

[2] Ю.И. Журавлев. Избранные научные труды. Москва, из-во Магистр, 1996 г., стр. 378–384

[3] Ю.И. Журавлев, И.В. Исаев. Построение алгоритмов распознавания, корректных для заданной контрольной выборки. Журнал вычислительной математики и математической физики, том 19. №3, 1979 г.