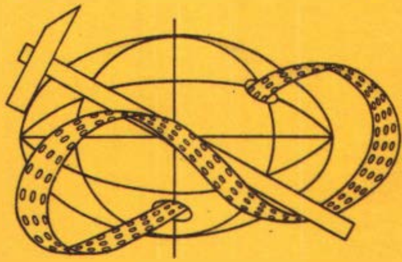
АКАДЕМИЯ НАУК СССР
СИБИРСКОЕ ОТДЕЛЕНИЕ
ИНСТИТУТ ГЕОЛОГИИ И ГЕОФИЗИКИ

ПРОГРАММНЫЕ КОМПЛЕКСЫ ДЛЯ ЦЕЛЕВОЙ ОБРАБОТКИ ИНФОРМАЦИИ

(ОПЕРАТИВНО-ИНФОРМАЦИОННЫЙ МАТЕРИАЛ)



НОВОСИБИРСК-1977

АКАДЕМИЯ НАУК СССР
СИБИРСКОЕ ОТДЕЛЕНИЕ
ИНСТИТУТ ГЕОЛОГИИ И ГЕОФИЗИКИ

ПРОГРАММНЫЕ КОМПЛЕКСЫ ДЛЯ ЦЕЛЕВОЙ ОБРАБОТКИ ИНФОРМАЦИИ

(ОПЕРАТИВНО-ИНФОРМАЦИОННЫЙ МАТЕРИАЛ)

СБОРНИК НАУЧНЫХ ТРУДОВ

НОВОСИБИРСК–1977

А Н Н О Т А Ц И Я

В оперативно-информационном материале представлены алгоритмы и реализующие их программы для логико-математической обработки разнообразной геологической информации (в основном для решения задач прогноза и поиска полезных ископаемых).

Программы написаны для ЭВМ типов: М-220, М-222, БЭСМ-4М, БЭСМ-6.

Материал предназначен для широкого круга научных и инженерно-технических работников, интересующихся вопросами математической обработки геолого-геофизической информации.

Ответственный редактор
А.Н.Дмитриев

Редакционная коллегия: В.О.Красавчиков,
Е.А.Смертин, Т.И.Штатнова

Составители:

В.В.Бабич, А.А.Бишаев, А.Н.Дмитриев, В.В.Зуенко, В.Д.Карбышев, В.Н.Кандыба, В.О.Красавчиков, Е.И.Ланда, Ю.Я.Лыков, С.В.Макаров, Л.С.Мальцева, Н.А.Осипов, А.И.Прокопенко, А.Д.Соколов, Е.А.Смертин, А.Н.Фиш, Н.А.Чернецкая, М.Л.Шемякин.

Печатается по решению секции
стратиграфии, литологии и осадочных
полезных ископаемых Ученого Совета
Института геологии и геофизики СО АН СССР

Предисловие

Оперативно-информационный материал состоит из шести комплексов алгоритмов и программ. Публикуются программы, прошедшие широкое апробирование при решении конкретных задач геологического профиля. Зачастую это задачи по прогнозированию и поиску полезных ископаемых, по классификации геологических объектов, минимизации признаковых пространств[ж)], а также задачи, связанные с геометрическими трактовками границ геологических тел и некоторыми статистическими аспектами обработки геологических данных[жж)]. Проведен также комплекс (УI) алгоритмов, модифицирующий широко известный тестовый подход.

Программы составлены для ЭВМ М-220, М-222, М-222М, БЭСМ-4М, БЭСМ-6 .

I. Разработки, представленные комплексом I, являются дальнейшим углублением и расширением метода согласованных оценок. Здесь приведено изложение алгоритмов (авторы Е.А.Смертин, А.Н.Дмитриев, С.В.Макаров) по центрированным качельным процедурам для оценки строк и столбцов, а также даны процедуры расчета коэффициентов распознавания. Алгоритмы реализованы тремя программами (программист Кандыба В.Н.): по первой - вычисляются согласованные оценки строк и столбцов, по второй - коэффициенты распознавания для бинарных таблиц, а по третьей - для числовых таблиц.

2. Комплекс II представляет собой дальнейшие разработки метода согласования и приуроченности объектов в направлении развития целевого классифицирования и упорядочения объектов (автор Бабич В.В.). В алгоритмическую основу положена система итерационных процедур, специализация которых сводится к некоторым задачам распознавания. Комплекс программ(программисты: Соколов А.Д., Ланда Е.И., Бабич В.В.) подразделен на две группы - вспомогательные программы (III-II4) и основные (II5-II6). Вспомогательные предназначены для записи исходной информации, ее транспонирования, организации в исходную таблицу, контроля вводимой информации. Основные программы реализуют процедуры: по оценке призна -

ж) Комплексы I, II, Ш

жж)Комплексы IУ,У

ков исходных таблиц, минимизации пространства признаков, распознавания.

3. Метод целевой итерационной классификации составляет алгоритмический и программный комплекс Ш. Этот комплекс предназначен для решения задач автоматической классификации и распознавания. Алгоритмическая разработка и ее программная реализация осуществлена Бишаевым А.А. Комплекс из двух программ позволяет проводить выбор программной системы характеристических признаков в соответствии с известными значениями целевого признака исследуемой совокупности объектов-эталонов. На последующих шагах производится принятие решений о значениях целевого признака для объектов-проб на основе информативной системы признаков.

4. Комплекс IУ объединяет собой программы, нацеленные на автоматические способы трактовок некоторых геометрических свойств геологических поверхностей и разграничений тел. Программы ПI (Прокопенко А.И., Фиш А.М., Лыков Ю.Я.) и П2 (Прокопенко А.И.,Мальцева Л.С., Лыков Ю.Я.), реализуют два подхода к аппроксимации кривых для прогнозирования структурных планов: приближение полиномом заданной степени (ПI) и сплайн-функцией (П2). Среди дополнительных параметров вычисляются центры мгновенных радиусов кривизны и др. Программа П3 (Осипов Н.А., Шемякин М.Л.) производит построение системы треугольников, вершинами которых являются заданные точки наблюдений. Полученная треугольная сетка дает возможность произвести простые и эффективные построения карт изолиний.

5. Комплекс У объединяет собой пять программ. Первые четыре программы (программист Карбышев В.Д.) подразделяются на вспомогательные (ПI и П2) и на основные (П3, П4). Основные программы предназначена для вычисления статистических параметров, позволяющих проводить проверку гипотезы о нормальном распределении для вычисления матрицы парных коэффициентов корреляции, частных и сводных коэффициентов корреляции первого порядка. Программа (П5) позволяет производить разбиение всего множества исходных объектов на однородные совокупности (в смысле выполнения критерия) по одному или нескольким признакам. Программа составлена Чернецкой Н.Л. и Зуенко В.В.

6. В комплексе УI описаны алгоритмы (автор Красавчиков В.О.), реализующие частные случаи схемы распознавания по Т-свойствам. Они

предназначены для анализа геологических данных, представленных бинарными и многозначными признаками. Основная идея схемы модификация широко известного тестового подхода на основе понятия Т - свойства.

В целом комплексы, изложенные в данном оперативно-информационном материале, предназначены для решений широкого круга геологических задач, связанных с обработкой больших массивов информации. Гибкость алгоритмов и программ позволяет проводить многоцелевое исследование исходных данных, сведенных в табличную форму.

I. КОМПЛЕКС ПРОГРАММ ПО МЕТОДУ СОГЛАСОВАННЫХ ОЦЕНОК

Алгоритмические разработки и математические справки к программам

Представленный комплекс программ естественным образом продолжает программное обеспечение и алгоритмические разработки метода согласованных оценок, опубликованные в [1-6]. Данный комплекс объединяет программы по оценке строк и столбцов описаний с помощью центрированных качельных процедур (ЦКП) [3] с программами процедур распознавания с помощью этих оценок. Изложим алгоритм оценки строк и столбцов таблиц.

I. Алгоритм центрированных качельных процедур

Дана таблица T из числовых элементов t_{ij}, стоящих на пересечении i-й строки – s_i и j-го столбца – x_j таблицы T, $i = 1,2,\dots,m$, $j = 1,2,\dots,n$; m – число строк, а n – число столбцов таблицы. Далее, пусть t_{ij} – любые вещественные числа – $-\infty < t_{ij} < \infty$. Строки таблицы T соответствуют некоторым объектам, а столбцы – признакам объектов. Итак, значение t_{ij} отражает выраженность признака x_j на объекте s_i (для удобства мы отождествляем обозначения строк s_i и объектов и столбцов x_j с признаками). Условимся, что $m \geq 2$, $n \geq 2$ и что в таблице T отсутствуют строки и столбцы, заполненные полностью одним и тем же значением. Введем числовую меру для оценки строк и столбцов таблицы T.

Числовую меру для объектов и признаков можно задавать линейной, согласной значениям объектов и признаков, что приводит к получению весов [1,2].

Другой способ задания числовой меры – задать её с помощью согласования значений весов с величинами отклонений значений элементов столбцов и строк от их средних значений. Тогда мы приходим к так называемым центрированным качельным процедурам [3], которые ведут учет не абсолютным значениям элементов таблицы, а их отклонениям от средних значений по строкам или по столбцам.

Предварительно таблица T преобразуется в таблицу U с помощью следующей нормировки её столбцов:

$$u_{ij} = \frac{t_{ij} - t_{j\,min}}{t_{j\,max} - t_{j\,min}}, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n,$$

где $t_{j\,min}$ – минимальный, а $t_{j\,max}$ – максимальный элемент столбца x_j, откуда видно, что u_{ij} попадают в промежуток $0 \le u_{ij} \le 1$. Веса строк и столбцов для таблицы T будем получать согласно таблице U.

Один вариант подсчета весов по таблице U ведется по следующим формулам

$$a_i^{(r+1)} = \frac{\sum\limits_{j=1}^{n} b_j^{(r)} \cdot (u_{ij} - \bar{x}_j)^2}{\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} b_j^{(r)} \cdot (u_{ij} - \bar{x}_j)^2}, \quad b_j^{(r+1)} = \frac{\sum\limits_{i=1}^{m} a_i^{(r)} \cdot (u_{ij} - \bar{s}_i)^2}{\sum\limits_{j=1}^{n} \sum\limits_{i=1}^{m} a_i^{(r)} \cdot (u_{ij} - \bar{s}_i)^2} \qquad (1)$$

где r – номер итерации; $\bar{s}_i$ – среднее арифметическое значение строки s_i; $\bar{x}_j$ – среднее арифметическое значение столбца x_j; причем $a_i^{(0)} = b_j^{(0)} = 1$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$.

Другой вариант центрированной качельной процедуры задается такими формулами:

$$a_i^{(r+1)} = \frac{\sum\limits_{j=1}^{n} b_j^{(r)} \cdot (u_{ij} - \bar{s}_i)^2}{\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} b_j^{(r)} \cdot (u_{ij} - \bar{s}_i)^2}, \quad b_j^{(r+1)} = \frac{\sum\limits_{i=1}^{m} a_i^{(r)} \cdot (u_{ij} - \bar{x}_j)^2}{\sum\limits_{j=1}^{n} \sum\limits_{i=1}^{m} a_i^{(r)} \cdot (u_{ij} - \bar{x}_j)^2}, \qquad (2)$$

которые отличаются от формул (I) перестановкой центрирования по строкам и центрирования по столбцам.

В программе "Вычисление согласованной системы информационных весов по ЦКП" реализованы оба варианта подсчета весов, как заданный формулами (I), так и заданный формулами (2), но на печать выдаются не значения $a_i^{(r)}$, $b_j^{(r)}$, а значения $\alpha_i^{(r)} = \sqrt{a_i^{(r)}}$, $\beta_j^{(r)} = \sqrt{b_j^{(r)}}$, которые более выразительно характеризуют строки и столбцы таблиц.

Перейдем к изложению алгоритмов распознавания с помощью ЦКП.

2. Алгоритмы распознавания

Пусть даны две таблицы T1 и T2 и таблица проб T_n. Требуется для каждой строки S_n из T_n указать к какой из таблиц T1 и T2 строка S_n тяготеет в большей степени.

а) Проведем распознавание в бинарном варианте ЦКП, когда таблицы T1, T2, T_n состоят из символов алфавита $\{0,1\}$. Соответствующая схема процедур распознавания описана в [2], процедура же ЦКП употребляется здесь для вычисления таких оценок ρ_j^ℓ столбцов и оценок π_i^ℓ строк таблиц T_ℓ, $\ell = 1,2$, которые удовлетворяли бы следующим соотношениям:

$$\frac{\sum_{j=1}^{n}\left[\rho_j^\ell\cdot(t_{ij}^\ell - \varrho_j^\ell)\right]^2}{\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\rho_j^\ell\cdot(t_{ij}^\ell-\varrho_j^\ell)\right]^2}, \quad (\rho_j^\ell)^2 = \frac{\sum_{i=1}^{m}\left[\pi_i^\ell\cdot(t_{ij}^\ell - Q_i^\ell)\right]^2}{\sum_{j=1}^{n}\sum_{i=1}^{m}\left[\pi_i^\ell\cdot(t_{ij}^\ell-Q_i^\ell)\right]^2}, \tag{3}$$

где ϱ_j^ℓ – частота встречаемости "1" в j-м столбце таблицы T_ℓ, а Q_i^ℓ – частота "единиц" в i-й строке T_ℓ.

б) Распознавание по числовым значениям проводится с процедур предварительной нормировки этих значений согласно такому приему

$$u_{ij} = \frac{t_{ij} - t_{j\,min}}{t_{j\,max} - t_{j\,min}}$$

где t_{ij} – подлежащий нормировке элемент из j-го столбца и i-й строки таблицы T, которая состоит из строк таблиц T1, T2, T_n, помещенных так, что строки таблицы T1 располагаются выше, а строки из T_n ниже, чем строки таблицы T_2; здесь $t_{j\,min}$ – минимальный элемент j-го столбца таблиц T1, T2, а $t_{j\,max}$ – максимальный элемент j-го столбца таблиц T1, T2. В результате этой нормировки все элементы таблиц T1, T2 попадают в интервал $[0,1]$, а элементы T_n могут стать меньше 0 или больше 1.

Обозначим получившиеся в результате нормировки таблицы теми же обозначениями $T1$, $T2$, T_n, которые были до нормировки.

Употребляется 2 варианта процедур ЦКП для проведения распознавания строк T_n на принадлежность к $T1$ или $T2$.

В первом варианте ЦКП служит для вычисления таких оценок строк π_i^ℓ и оценок столбцов ρ_j^ℓ таблиц T_ℓ, $\ell = 1,2$, которые удовлетворяли бы следующим соотношениям

$$(\pi_i^\ell)^2 = \frac{\sum_{j=1}^{n} [\rho_j^\ell \cdot (t_{ij}^\ell - \bar{s}_i^\ell)]^2}{\sum_{i=1}^{m_\ell} \sum_{j=1}^{n} [\rho_j^\ell (t_{ij}^\ell - \bar{s}_i^\ell)]^2}, \quad (\rho_j^\ell)^2 = \frac{\sum_{i=1}^{m_\ell} [\pi_i^\ell \cdot (t_{ij}^\ell - \bar{x}_j^\ell)]^2}{\sum_{j=1}^{n} \sum_{i=1}^{m_\ell} [\pi_i^\ell \cdot (t_{ij}^\ell - x_j^\ell)]^2} \quad (4)$$

где $\bar{x}_j^\ell$ – среднее арифметическое значений j-го столбца, а $\bar{s}_i^\ell$ – среднее арифметическое значений i-й строки таблицы T_ℓ.

Во втором варианте ЦКП соответствующие оценки должны удовлетворить таким равенствам

$$(\pi_i^\ell)^2 = \frac{\sum_{j=1}^{n} [\rho_j^\ell \cdot (t_{ij}^\ell - \bar{x}_j^\ell)]^2}{\sum_{i=1}^{m_\ell} \sum_{j=1}^{n} [\rho_j^\ell (t_{ij}^\ell - \bar{x}_j^\ell)]^2}, \quad (\rho_j^\ell)^2 = \frac{\sum_{i=1}^{m_\ell} [\pi_i^\ell \cdot (t_{ij}^\ell - \bar{s}_i^\ell)]^2}{\sum_{j=1}^{n} \sum_{i=1}^{m_\ell} [\pi_i^\ell \cdot (t_{ij}^\ell - \bar{s}_i^\ell)} \quad (5)$$

В соответствии с указанными способами вычисления оценок строк и столбцов таблиц T_ℓ для распознавания используются различного вида характеристики строк. В первом варианте (формулы (4)) употребляются величины

$$\bar{z}(\rho^\ell, s) = \sum_{j=1}^{n} [(t_j - \bar{s}) \cdot \rho_j^\ell]^2,$$

где $\bar{s} = \frac{1}{n}(t_1 + \ldots + t_n)$ – среднее арифметическое компонент строки $S = (t_1, \ldots, t_n)$, а во втором (формулы (5)) – величины

$$\bar{z}(\rho^\ell, s) = \sum_{j=1}^{n} [(t_j - \bar{x}_j^\ell) \cdot \rho_j^\ell]^2, \quad \ell = 1, 2.$$

В обоих вариантах составляется отношение

$$\bar{R}(\rho^1,\rho^2,S) = \frac{\bar{\varepsilon}(\rho^1, S)}{\bar{\varepsilon}(\rho^2, S)},$$

которое имеет смысл при $\bar{\varepsilon}(\rho^2, S) \neq 0$ и характеризует удаленность S от TI.

Указанное отношение используется для распознавания, начиная с нахождения таких чисел

$$\bar{\alpha}_1 = \max_{S \in T_1} \bar{R}(\rho^1,\rho^2,S), \quad \bar{\alpha}_2 = \min_{S \in T_2} \bar{R}(\rho^1,\rho^2,S).$$

Возможны 2 случая: либо $\bar{\alpha}_1 < \bar{\alpha}_2$, либо $\bar{\alpha}_1 \geq \bar{\alpha}_2$.
Для произвольной строки S_n из T_n в первом случае ($\bar{\alpha}_1 < \bar{\alpha}_2$) принимаются следующие решения:

а) если $\bar{R}(\rho^1,\rho^2,S_n) \leq \bar{\alpha}_1$, то S_n относится к TI;

б) если $\bar{R}(\rho^1,\rho^2,S_n) \geq \bar{\alpha}_2$, то S_n относится к T2;

в) если $\bar{\alpha}_1 < \bar{R}(\rho^1,\rho^2,S_n) < \bar{\alpha}_2$, то S_n относится к зоне отказа от решения.

Во втором случае ($\bar{\alpha}_1 \geq \bar{\alpha}_2$) решение таково:

а) если $\bar{R}(\rho^1,\rho^2,S_n) < \bar{\alpha}_2$, то S_n относится к TI;

б) если $\bar{R}(\rho^1,\rho^2,S_n) > \bar{\alpha}_1$, то S_n относится к T2;

в) если $\bar{\alpha}_2 \leq \bar{R}(\rho^1,\rho^2,S_n) < \bar{\alpha}_1$, то S_n попадает в зону отказа от принятия решения.

В программе "Расчет коэффициентов распознавания с помощью ЦКП для бинарных таблиц" выполняется процедура, охарактеризованная в пункте а), выдаются на печать веса π_i^ℓ, ρ_j^ℓ, заданные формулами (3) и величины R_I, R_{II}, R_{III}, R_{IV}, получаемые с помощью весов ρ_j^ℓ, как описано в [2].

Описанные в пункте б) 2 варианта вычисления отношения $\bar{R}(\rho^1,\rho^2,S)$ (пользуясь формулами (4) и (5)) реализованы в программе "Распознавание с помощью ЦКП". На печать выдаются значения π_i^ℓ, $(\pi_i^\ell)^2$, ρ_j^ℓ, $(\rho_j^\ell)^2$, удовлетворяющие соотношениям (4), (5), а также значения $\bar{R}(\rho^1,\rho^2,S_n)$ для всех строк из таблицы проб T_n.

Литература

I. Ю.Л.Васильев, А.Н.Дмитриев. Спектральный подход к сравнению объектов, охарактеризованных набором признаков. "ДАН СССР", 1972, т.206, № 6, с.1309-1312.

2. А.Н.Дмитриев, В.О.Красавчиков. Программы метода согласованных оценок. – В кн.: Логико-математическая обработка геологической информации. Новосибирск, 1975, с.6-13.

3. С.В.Макаров, Е.А.Смертин. Центрированная качельная процедура для нахождения согласованной системы информационных весов. – В кн.: Логико-математическая обработка геологической информации. Новосибирск, 1976, с.92-100.

4. В.Н.Кандыба, Программа ПI "Качели для бинарных таблиц". – В кн.: Логико-математическая обработка геологической информации. Новосибирск, 1975, с.14-19.

5. В.Н.Кандыба. Программа П2 "Расчет коэффициентов". – В кн.: Логико-математическая обработка геологической информации. Новосибирск, 1975, с.20-26.

6. Е.А.Смертин. Программа П3 "Симметрия". – В кн.: Логико-математическая обработка геологической информации. Новосибирск, 1975, с.27-35.

КОМПЛЕКС ПРОГРАММ (П1-П3).

Программа "Вычисление согласованной системы информационных весов по ЦКП" (П1).

(Язык Альфа-6, ЭВМ БЭСМ-6)

Назначение.

Программа предназначена для вычисления информационных весов строк a_i , α_i и столбцов b_j , β_j по двум вариантам для числовых и бинарных таблиц.

Инструкция к пользованию.

Колода комплектуется в соответствии с правилами, принятыми в транслирующей системе Альфа-6 для ЭВМ БЭСМ-6 [I] .

Порядок ввода:

СI - число задач

$$C2 - \begin{cases} C2 = I \text{ для бинарных таблиц} \\ C2 > I \text{ для числовых таблиц} \end{cases}$$

m - число строк в t

n - число столбцов в t

t - исходная таблица

Порядок вывода: I вар. - b_j, a_i, β_j, α_i ;

П вар. - b_j, a_i, β_j, α_i .

Контрольные примеры:

$$t_1 = \begin{bmatrix} I & 0 & 0 & II \\ I & 0 & 7 & 7 \\ I & 2 & 9 & 9 \\ I & 2 & 5 & I \\ 0 & 2 & I0 & 2,5 \end{bmatrix} \qquad t_2 = \begin{bmatrix} 0 & 0 & I & I & I & I \\ 0 & I & 0 & 0 & 0 & I \\ I & I & I & 0 & 0 & I \end{bmatrix}$$

Перфорация выполняется на УПП. Для контрольных примеров вводимая информация имеет вид:

2;

2; 5; 4;

I, 0, 0, II, I, 0, 7, 7, I, 2, 9, 9, I, 2, 5, I, 0,2,I0,2.5;

I; 3; 6;

0,0,I,I,I,I,0,I,0,0,0,I,I,I,I,0,0,I ;

Результаты счета контрольных примеров:

Задача № I.

I	b_j	0,26	0,33	0,2I	0,2I		
	a_i	0,3I	0,20	0,008	0,2I	0,27	
	β_j	0,5I	0,57	0,45	0,46		
	α_i	0,56	0,44	0,09	0,46	0,52	
II	b_j	0,28	0,29	0,20	0,24		
	a_i	0,28	0,I3	0,II	0,I4	0,33	
	β_j	0,53	0,53	0,44	0,49		
	α_i	0,53	0,36	0,33	0,37	0,58	

Задача № 2

I	b_j	0,2I	0,2I	0,I8	0,2I	0,2I	0
	a_i	0,37	0,27	0,37			
	β_j	0,45	0,45	0,42	0,45	0,45	0
	α_i	0,6I	0,52	0,6I			
II	b_j	0,2I	0,26	0,08	0,I6	0,I6	0,I4
	a_i	0,50	0,2I	0,29			
	β_j	0,46	0,5I	0,29	0,39	0,39	0,37
	α_i	0,7I	0,46	0,54			

Допустимый объем таблицы: $m \times n + 5m + 6n \leq 30247_{10}$. Время трансляции I9 сек. Время решения, для $m=4$, $n=9$, составляет 3 сек.

ТЕКСТ ПРОГРАММЫ П1.

```
НАЧАЛО ЦЕЛ I,J,N,M,K1,C,C1,C2;
¯I¯N¯P¯U¯T(0,C1);
НАЧАЛО ДЛЯ C:=1,...,C1 ЦИКЛ НАЧАЛО ¯I¯N¯P¯U¯T(0,C2,M,N);
НАЧАЛО ВЕЩ МАССИВ T[1:M,1:N],*АЛЬФА*,A,¯A,T1,*СИГМА*1[1:M],TMN,TMX,*БЕТА*,B,T2,*СИГМА*2
[1:N]; ВЕЩ S;
¯I¯N¯P¯U¯T(0,T);
ЕСЛИ C2=1 ТО НА ¯M1 ИНАЧЕ НА ¯M2; ¯M1:
ДЛЯ J:=1,...,N ЦИКЛ НАЧАЛО TMN[J]:=T[1,J];TMX[J]:=0; ДЛЯ I:=1,...,M ЦИКЛ НАЧАЛО ЕСЛИ
TMN[J]>T[I,J]
ТО TMN[J]:=T[I,J]; ЕСЛИ TMX[J]<T[I,J] ТО TMX[J]:=T[I,J] КОНЕЦ КОНЕЦ;
ДЛЯ J:=1,...,N ЦИКЛ ДЛЯ I:=1,...,M ЦИКЛ T[I,J]:=(T[I,J]-TMN[J])/(TMX[J]-TMN[J]); ¯M2:
ДЛЯ K1:=1,2 ЦИКЛ НАЧАЛО ДЛЯ I:=1,...,M ЦИКЛ T1[I]:=0; ДЛЯ I:=1,...,M ЦИКЛ ДЛЯ J:=
1,...,N ЦИКЛ
T1[I]:=T1[I]+T[I,J]; ДЛЯ I:=1,...,M ЦИКЛ T1[I]:=T1[I]/N; ДЛЯ J:=1,...,N ЦИКЛ T2[J]:=
ДЛЯ J:=1,...,N ЦИКЛ ДЛЯ I:=1,...,M ЦИКЛ T2[J]:=T2[J]+T[I,J]; ДЛЯ J:=1,...,N ЦИКЛ T
2[J]:=T2[J]/M;
ДЛЯ J:=1,...,N ЦИКЛ B[J]:=1; ДЛЯ I:=1,...,M ЦИКЛ A[I]:=1;
¯M3:
ДЛЯ I:=1,...,M ЦИКЛ *СИГМА*1[I]:=0; ДЛЯ I:=1,...,M ЦИКЛ ДЛЯ J:=1,...,N ЦИКЛ ЕСЛИ K1=1 ТО
```

```
*СИГМА*1[I]:=*СИГМА*1[I]+((T[I,J]-T1[I])↑2)×B[J] ИНАЧЕ
*СИГМА*1[I]:=*СИГМА*1[I]+((T[I,J]-T2[J])↑2)×B[J];
 ДЛЯ I:=1,...,M ЦИКЛ ¯A[I]:=A[I]; ДЛЯ J:=1,...,N ЦИКЛ *СИГМА*2[J]:=0;
 ДЛЯ J:=1,...,N ЦИКЛ  ДЛЯ I:=1,...,M ЦИКЛ  ЕСЛИ K1=1 ТО
*СИГМА*2[J]:=*СИГМА*2[J]+((T[I,J]-T2[J])↑2)×A[I] ИНАЧЕ
*СИГМА*2[J]:=*СИГМА*2[J]+((T[I,J]-T1[I])↑2)×A[I];
S:=0; ДЛЯ I:=1,...,M ЦИКЛ S:=S+*СИГМА*1[I]; ДЛЯ I:=1,...,M ЦИКЛ A[I]:=*СИГМА*1[I]/S;
S:=0; ДЛЯ J:=1,...,N ЦИКЛ S:=S+*СИГМА*2[J]; ДЛЯ J:=1,...,N ЦИКЛ B[J]:=*СИГМА*2[J]/S;
 ДЛЯ I:=1,...,M ЦИКЛ  ЕСЛИ ABS(¯A[I]-A[I])≥0.0000001 ТО  НА ¯M3;
 ЕСЛИ K1=1 ТО  НАЧАЛО ¯O¯U¯T¯P¯U¯T(0,'5/','15','T','¯Ц¯Е¯Н¯Т¯Р¯И¯Р¯О¯В¯А¯Н¯Н¯Ы¯Е
¯К¯А¯Ч¯Е¯Л¯И (¯Н¯О¯Р¯М¯И¯Р¯О¯В¯А¯Н¯Н¯Ы¯Е)','4/');
¯O¯U¯T¯P¯U¯T(0,'T','1 ¯В¯А¯Р¯И¯А¯Н¯Т','3/') КОНЕЦ
 ИНАЧЕ ¯O¯U¯T¯P¯U¯T(0,'3/','T','2 ¯В¯А¯Р¯И¯А¯Н¯Т','2/');
¯O¯U¯T¯P¯U¯T(0,'2/','T','B','/','E',B,'2/','T','A','/','E',A);
 ДЛЯ J:=1,...,N ЦИКЛ *БЕТА*[J]:=SQRT(B[J]); ДЛЯ I:=1,...,M ЦИКЛ *АЛЬФА*[I]:=SQRT(A[I
]);
¯O¯U¯T¯P¯U¯T(0,'T','*БЕТА*[J]','E',*БЕТА*,'2/','T','*АЛЬФА*[I]','E',*АЛЬФА*) КОНЕЦ  КОНЕЦ
 КОНЕЦ  КОНЕЦ  КОНЕЦ
```

Программа "Расчет коэффициентов распознавания с помощью ЦКП для бинарных таблиц"(П2).

(Язык Альфа-6, ЭВМ БЭСМ-6)

Назначение.

В программе вычисляются информационные веса строк π_i^2, π_i и столбцов ρ_j^2, ρ_j для двух классов эталонных таблиц и коэффициенты распознавания – R_I, R_{II}, R_{III}, R_{IV} для каждого объекта таблицы проб.

Инструкция к пользованию.

Колода формируется по правилам, принятым в транслирующей системе Алфа-6 для ЭВМ БЭСМ-6 [I] .

Порядок ввода:

CI – число задач
n – число столбцов
m_1 – число строк в t
m_2 – число строк в t_4
m_3 –число строк в t_3
ф – обобщенная реплика для t и t_4 (ф = Ф1 + Ф2)ж
t_3 – таблица проб
t – таблица эталонов I кл.
t_4 – таблица эталонов П кл.

Порядок вывода: I вариант – b_j, a_i – для таблицы I кл.; b_j, a_i – для таблицы эталонов П кл.; R_I, R_{II}, R_{III}, R_{IV}, $\beta_j \alpha_i$ – для табл. эталонов I кл., $\beta_j \alpha_i$ – для табл.эталонов П кл., и то же самое для П варианта (где $b_j - \rho_j^2$, $a_i - \pi_i^2$, $\beta_j - \rho_j$, $\alpha_i - \pi_i$)

Контрольный пример

t:

0	0	I	I	I	I
0	I	0	0	0	I
I	I	I	0	0	I
Ф1: 0	I	I	0	0	I

t_4:

I	0	I	I	I	I
I	0	0	0	0	I
0	0	I	0	0	I
Ф2: I	0	I	0	0	I

t_3:

0	0	0	0	I	I
0	I	I	0	0	I
I	I	0	0	0	I
0	I	0	I	0	I

ж) См. введение в [3] .

Перфорация выполняется на УПП. Для контрольных примеров информация представляется в следующем виде:

I;

6; 3; 3; 4;

0,I,I,0,0,I,I,0,I,0,0,I;

0,0,0,0,I,I,0,I,I,0,0,I,I,I,0,0,0,I,0,I,0,I,0,I;

0,0,I,I,I,I,0,I,0,0,0,I,I,I,I,0,0,I;

I,0,I,I,I,I,I,0,0,0,0,I,0,0,I,0,0,I;

Результаты счета контрольного примера.

I	b_j	I кл.	0,2I	0,2I	0,I8	0,2I	0,2I	0
	a_i	I кл.	0,37	0,27	0,37			
	b_j	П кл.	0,34	0	0,34	0,I6	0,I6	0
	a_i	П кл.	0,06	0,47	0,47			
	R_I		2,5I	I,56	0,93	3,77		
	R_{II}		I,00	I,50	I,00	2,00		
	R_{III}		0	I,I4	0,6I	20,59		
	R_{IV}		I,00	I,50	I,00	2,00		
	β_j	I кл.	0,45	0,45	0,42	0,45	0,45	0
	α_i	I кл.	0,6I	0,52	0,61			
	β_j	П кл.	0,58	0	0,58	0,40	0,40	0
	α_i	П кл.	0,24	0,69	0,69			
П	b_j	I кл.	0,I7	0,25	0,I7	0,07	0,07	0,27
	a_i	I кл.	0,32	0,34	0,34			
	b_j	П кл.	0,I7	0,25	0,I7	0,07	0,07	0,27
	a_i	П кл.	0,32	0,34	0,34			
	R_I		0,86	I,73	I,I4	2,24		
	R_{II}		I,00	I,50	I,00	2,00		
	R_{III}		I,00	I,58	I,I9	I,95		
	R_{IV}		I,00	I,50	I,00	2,00		
	β_j	I кл.	0,4I	0,50	0,4I	0,27	0,27	0,52
	α_i	I кл.	0,56	0,58	0,58			
	β_j	П кл.	0,4I	0,50	0,4I	0,27	0,27	0,52
	α_i	П кл.	0,56	0,58	0,58			

Допустимый максимальный объем задачи:

$2m(n+3)+m3(n+4)+7n \le 29959_{10}$, если $m_1 > m_2$, то $m_1 = m_2$ иначе $m = m_1$.

Время решения задачи при $n=30$, $m_1 = m_2 = 7$, $m_3 = 26$ составляет 4 сек. Время трансляции 29 сек.

ТЕКСТ ПРОГРАММЫ П2.

```
НАЧАЛО  ЦЕЛ I,J,N,M,M1,M2,M3,K,K1,C,C1;¯I¯N¯P¯U¯T(0,C1,N,M1,M2,M3);
 НАЧАЛО  ДЛЯ C:=1,...,C1 ЦИКЛ  ЕСЛИ M2>M1 ТО M:=M2 ИНАЧЕ M:=M1;
 НАЧАЛО  ВЕЩ  МАССИВ T,T4[1:M,1:N],T3[1:M3,1:N],
*АЛЬФА*1 *АЛЬФА*2,A,-A,T ,*СИГМА*1[1:M],B,T2,*СИГМА*2[1:N],*БЕТА*,Ф[1:2,1:N],
R1,R2,B1,B2[1:M3]; ВЕЩ S;
¯I¯N¯P¯U¯T(0,Ф,T3,T,T4);
 ДЛЯ K1:=1,2 ЦИКЛ  НАЧАЛО  ЕСЛИ K1=1 ТО ¯O¯U¯T¯P¯U¯T(0,'T','1 ¯В¯А¯Р¯И¯А¯Н¯Т','3/');
 ЕСЛИ K1=2 ТО ¯O¯U¯T¯P¯U¯T(0,'2/','T','2 ¯В¯А¯Р¯И¯А¯Н¯Т','2/');
 ДЛЯ K:=1,2 ЦИКЛ  НАЧАЛО  ЕСЛИ K=1 ТО M:=M1 ИНАЧЕ  НАЧАЛО M:=M2;
 ДЛЯ I:=1,...,M ЦИКЛ  ДЛЯ J:=1,...,N ЦИКЛ T[I,J]:=T4[I,J] КОНЕЦ ;
 ДЛЯ I:=1,...,M ЦИКЛ T1[I]:=0; ДЛЯ I:=1,...,M ЦИКЛ  ДЛЯ J:=1,...,N ЦИКЛ
T1[I]:=T1[I]+T[I,J]; ДЛЯ I:=1,...,M ЦИКЛ T1[I]:=T1[I]/N; ДЛЯ J:=1,...,N ЦИКЛ T2[J]:=0;

 ДЛЯ J:=1,...,N ЦИКЛ  ДЛЯ I:=1,...,M ЦИКЛ T2[J]:=T2[J]+T[I,J]; ДЛЯ J:=1,...,N ЦИКЛ T
2[J]:=T2[J]/M;
 ДЛЯ J:=1,...,N ЦИКЛ B[J]:=1; ДЛЯ I:=1,...,M ЦИКЛ A[I]:=1;
¯M3:
```

```
¯ДЛЯ I:=1,...,M ЦИКЛ *СИГМА*1[I]:=0; ДЛЯ I:=1,...,M ЦИКЛ ДЛЯ J:=1,...,N ЦИКЛ ЕСЛИ
K1=1 ТО
*СИГМА*1[I]:=*СИГМА*1[I]+((T[I,J]-T1[I])↑2)×B[J] ИНАЧЕ
*СИГМА*1[I]:=*СИГМА*1[I]+((T[I,J]-T2[J])↑2)×B[J];
 ДЛЯ I:=1,...,M ЦИКЛ ¯A[I]:=A[I]; ДЛЯ J:=1,...,N ЦИКЛ *СИГМА*2[J]:=0;
 ДЛЯ J:=1,...,N ЦИКЛ ДЛЯ I:=1,...,M ЦИКЛ ЕСЛИ K1=1 ТО
*СИГМА*2[J]:=*СИГМА* [J]+((T[I,J]-T2[J])↑2)×A[I] ИНАЧЕ
*СИГМА*2[J]:=*СИГМА* [J]+((T[I,J]-T1[I])↑2)×A[I];
S:=0; ДЛЯ I:=1,...,M ЦИКЛ S:=S+*СИГМА*1[I]; ДЛЯ I:=1,...,M ЦИКЛ A[I]:=*СИГМА*1[I]/S;
S:=0; ДЛЯ J:=1,...,N ЦИКЛ S:=S+*СИГМА*2[J]; ДЛЯ J:=1,...,N ЦИКЛ B[J]:=*СИГМА*2[J]/S;
 ДЛЯ I:=1,...,M ЦИКЛ ЕСЛИ ABS(¯A[I]-A[I])≥0.0000001 ТО НА ¯M3;
 ЕСЛИ K=1 ТО НАЧАЛО ДЛЯ J:=1,...,N ЦИКЛ *БЕТА*[1,J]:=B[J]; ДЛЯ I:=1,...,M1 ЦИКЛ *АЛЬФА*
1[I]:=A[I] КОНЕЦ
 ИНАЧЕ НАЧАЛО ДЛЯ J:=1,...,N ЦИКЛ *БЕТА*[2,J]:=B[J]; ДЛЯ I:=1,...,M2 ЦИКЛ *АЛЬФА*2
[I]:=A[I] КОНЕЦ ;
¯O¯U¯T¯P¯U¯T(0,'2/','T','B','/','E',B,'2/','T','A','/','E',A) КОНЕЦ ;
 ДЛЯ K:= ,2 ЦИКЛ НАЧАЛО ДЛЯ I:=1,...,M3 ЦИКЛ НАЧАЛО B1[I]:=B2[I]:=0; ДЛЯ J:=1,...
,N ЦИКЛ ЕСЛИ Ф[K,J]=T3[I,J]
 ТО НАЧАЛО B1[I]:=B1[I]+*БЕТА*[K,J];B2[I]:=B2[I]+1 КОНЕЦ КОНЕЦ ;
```

```
ДЛЯ I:=1,...,M3 ЦИКЛ ЕСЛИ K=1 ТО НАЧАЛО R1[I]:=B1[I];R2[I]:=B2[I] КОНЕЦ
ИНАЧЕ НАЧАЛО R1[I]:=R1[I]/B1[I];R2[I]:=R2[I]/B2[I] КОНЕЦ КОНЕЦ ;
-O-U-T-P-U-T(0,‘2/’,‘T’,‘R1’,‘2/’,‘E’,R1,‘5/’,‘T’,‘R2’,‘2/’,‘E’,R2);
ДЛЯ K:=1,2 ЦИКЛ НАЧАЛО ДЛЯ I:=1,...,M3 ЦИКЛ НАЧАЛО B1[I]:=B2[I]:=0; ДЛЯ J:=1,...
,N ЦИКЛ ЕСЛИ Ф[K,J]=T3[I,J]=1 ТО
НАЧАЛО B1[I]:=B1[I]+*БЕТА*[K,J];B2[I]:=B2[I]+1 КОНЕЦ КОНЕЦ ;
ДЛЯ I:=1,...,M3 ЦИКЛ ЕСЛИ K=1 ТО НАЧАЛО R1[I]:=B1[I];R2[I]:=B2[I] КОНЕЦ ;
ДЛЯ I:=1,...,M3 ЦИКЛ ЕСЛИ K=2 ТО НАЧАЛО ЕСЛИ B2[I]≠0 ТО НАЧАЛО R2[I]:=R2[I]/B2[I];
ЕСЛИ B1[I]=0 ТО R1[I]:=R1[I]/0.01 ИНАЧЕ R1[I]:=R1[I]/B1[I] КОНЕЦ ИНАЧЕ R2[I]:=R1[I
]:=0 КОНЕЦ КОНЕЦ;
-O-U-T-P-U-T(0,‘2/’,‘T’,‘R1’,‘2/’,‘E’,R1,‘5/’,‘T’,‘R2’,‘2/’,‘E’,R2);
ДЛЯ K:=1,2 ЦИКЛ ДЛЯ J:=1,...,N ЦИКЛ *БЕТА*[K,J]:=SQRT(*БЕТА*[K,J]);
ДЛЯ I:=1,...,M1 ЦИКЛ *АЛЬФА*1[I]:=SQRT(*АЛЬФА*1[I]); ДЛЯ I:=1,...,M2 ЦИКЛ *АЛЬФА*2[
I]:=SQRT(*АЛЬФА*2[I]);
-O-U-T-P-U-T(0,‘/’,‘T’,‘*БЕТА*[K,J]’,‘/’,‘E’,*БЕТА*,
‘2/’,‘T’,‘*АЛЬФА* [I]’,‘/’,‘E’,*АЛЬФА*1,
‘2/’,‘T’,‘*АЛЬФА*2[I]’,‘/’,‘E’,*АЛЬФА*2);
КОНЕЦ КОНЕЦ КОНЕЦ КОНЕЦ
```

Программа "Распознавание с помощью ЦКП для числовых таблиц" (П3).

(Язык Альфа-6, ЭВМ БЭСМ-6)

Назначение. В программе вычисляются информационные веса строк $-\pi_i^2$, π_i и столбцов $-\rho_j^2$, ρ_i для двух классов эталонных таблиц и коэффициент распознавания $-\bar{R}(\rho^1, \rho^2, S)$ для каждого объекта таблицы проб.

Инструкция к пользованию. Таблицы исходных данных перфорируются по строкам. В число проб обязательно вводятся таблицы эталонов I и П классов. Колода формируется по правилам, принятым в транслирующей системе Альфа-6 [I].

Порядок ввода:

cI – число задач,

n – число столбцов,

m_1 – число строк в t_1,

m_2 – число строк в t_2,

m_3 – число строк в t_3,

$t - (t_1 + t_2)$

t_3 – таблица проб, где t_1 – матрица эталонных объектов I класса,

t_2 – матрица эталонных объектов П класса.

Порядок вывода: для I и П вариантов – b_1, a_1, α_1, b_2, a_2, α_2, β; z_1, z_2, где $b_1 - \rho_j^2$ (для эталонов I кл.), $a_1 - \pi_i^2$ (для эталонов I кл.), $\alpha_1 - \pi_i$ (для эталонов I кл.), $b_2 - \rho_j^2$ (для эталонов П кл.), $a_2 - \pi_i^2$ (для эталонов П кл.), $\alpha_2 - \pi_i$ (для эталонов П кл.), $\beta[i,j] - \rho_j$ (n – значений для I кл., и n следующих значений для П кл.); $z_1 - \bar{R}(\rho_1, \rho_2, S)$ I кл. и $z_2 - \bar{R}(\rho_1, \rho_2, S)$ П кл.

Контрольный пример.

$$t_1 = \begin{bmatrix} 5,6 & 7,2 & 16,7 \\ 6,1 & 8,1 & 21,1 \\ 4,3 & 3,3 & 40,9 \end{bmatrix} \qquad t_2 = \begin{bmatrix} 4,4 & 10,1 & 33,3 \\ 4,6 & 9,7 & 47,6 \\ 4,8 & 6,1 & 91,7 \end{bmatrix}$$

$$t_3 = \begin{bmatrix} 8,1 & 8,6 & 17,2 \\ 3,1 & 5,4 & 27,3 \\ 5,6 & 5,1 & 30,4 \\ 7,2 & 3,1 & 51,6 \end{bmatrix}$$

Перфорация выполняется на УПП. Контрольный пример подготовленный для перфорации имеет вид:

I;

3; 3; 3; I0;

5.6, 7.2, I6.7, 6.I, 8.I, 2I.I, 4.3, 3.3, 40.9, 4.4, I0.I, 33.3, 4.6, 9.7, 47.6, 4.8, 6.I, 9I.7;
5.6, 7.2, I6.7, 6.I, 8.I, 2I.I, 4.3, 3.3, 40.9, 4.4, I0.I, 33.3, 4.6, 9.7, 47.6, 4.8, 6.I, 9I.7, 8.I, 8.6, I7.2, 3.I, 5.4, 27.3, 5.6, 5.I, 30.4, 7.2, 3.I, 5I.6;

Результаты решения контрольного примера

I	b_j	0,64	0,3I	0,049			
	a_i	0,33	0,6I	0,063			
	$\alpha 1_i$	0,58	0,78	0,25			
	b_j	0,047	0,36	0,59			
	a_i	0,42	0,22	0,35			
	$\alpha 2_i$	0,65	0,47	0,59			
	$\beta[1,j]$	0,80	0,55	0,55			
	$\beta[2,j]$	0,22	0,60	0,77			
II	b_j	0,3I	0,064	0,62			
	a_i	0,08	0,29	0,62			
	$\alpha 1_i$	0,29	0,54	0,79			
	b_j	0,28	0,34	0,37			
	a_i	0,29	0,074	0,64			
	$\alpha 2_i$	0,54	0,27	0,80			
	$\beta[1,j]$	0,56	0,25	0,79			
	$\beta[2,j]$	0,53	0,59	0,6I			
	τ_1	0,57	0,7I	0,4I	I,27	I,68	0,55
		I,46	2,85	2,I5	2,84		
	τ_2	0,086	0,23	0,58	I,89	7,92	3,88
		0,65	I,45	0,046	0,52		

Максимальный объем вводимой информации – $(n+2)\cdot(m_1+m_2)+m_3(n+6)+\mathrm{II}\,n+2\,m \leq 29647_{\mathrm{I0}}$.

Время трансляции 33 сек.

Литература

I. А.О.Буда, Т.С.Васючкова, А.А.Грановский, С.Э.Козловский, В.И.Шелехов. Руководство к пользованию системой автоматизации программирования. Альфа-6. Под ред. А.П.Ершова, ВЦ СО АН, Новосибирск, I974, 187 с.

3. Логико-математическая обработка геологической информации. ИГиГ СО АН, Новосибирск, I975, 189 с.

ТЕКСТ ПРОГРАММЫ П3.

```
НАЧАЛО ЧЕЛ I,J,N,M,M1,M2,M3,C,C1,K,K1,P;
-I-N-P-U-T(0.C1,N,M1,M2,M3);
НАЧАЛО ДЛЯ C:=1,...,C1 ЦИКЛ ЕСЛИ M2>M1 ТО M:=M2 ИНАЧЕ M:=M1;
НАЧАЛО ВЕЩ МАССИВ T[1:(M1+M2),1:N],T3[1:M3,1:N],*АЛЬФА*1,*АЛЬФА*2[1:M],
P1,P2,TMN,TMX,B,T2,*СИГМА*2[1:N],*БЕТА*1,*БЕТА*[1:2,1:N],
R1,R2,B1,B2[1:M3],A,-A,*СИГМА*1,T1[1:(M1+M2)]; ВЕЩ S;
-I-N-P-U-T(0,T,T3);
M:=M1+M2;
ДЛЯ J:=1,...,N ЦИКЛ НАЧАЛО TMN[J]:=T[1,J];TMX[J]:=0; ДЛЯ I:=1,...,M ЦИКЛ НАЧАЛО ЕСЛИ
TMN[J]>T[I,J]
ТО TMN[J]:=T[I,J]; ЕСЛИ TMX[J]<T[I,J] ТО TMX[J]:=T[I,J] КОНЕЦ КОНЕЦ ;
ДЛЯ J:=1,...,N ЦИКЛ ДЛЯ I:=1,...,M ЦИКЛ T[I,J]:=(T[I,J]-TMN[J])/(TMX[J]-TMN[J]);
ДЛЯ K1:=1,2 ЦИКЛ НАЧАЛО ЕСЛИ K1=1
ТО НАЧАЛО -O-U-T-P-U-T(0,'4/','12B','T','-Р-А-С-П-О-З-Н-А-В-А-Н-И-Е -П-О-Ц-Е-Н-Т-Р
-И-Р-О-В-А-Н-Н-Ы-М
-К-А-Ч-Е-Л-Я-М','/','24B',
'T','-Д-Л-Я -М-Н-О-Г-О-З-Н-А-Ч-Н-Ы-Х -Т-А-Б-Л-И-Ц');
```

```
¯O¯U¯T¯P¯U¯T(D,'2/','T','1 ¯B¯A¯P¯И¯A¯H¯T','2/') КОНЕЦ
 ИНАЧЕ ¯O¯U¯T¯P¯U¯T(D,'2/','T','2 ¯B¯A¯P¯И¯A¯H¯T','2/');
 ДЛЯ K:=1,2 ЦИКЛ  НАЧАЛО  ЕСЛИ K=1 ТО  НАЧАЛО P:=1;M:=M1 КОНЕЦ  ИНАЧЕ  НАЧАЛО P:=M1+
1;M:=M1+M2 КОНЕЦ ;
T1[]:=0; ДЛЯ I:=P,...,M ЦИКЛ  ДЛЯ J:=1,...,N ЦИКЛ T1[I]:=T1[I]+T[I,J];
 ДЛЯ I:=P,...,M ЦИКЛ T1[I]:=T1[I]/N; ДЛЯ J:=1,...,N ЦИКЛ T2[J]:=0;
 ДЛЯ J:=1,...,N ЦИКЛ  ДЛЯ I:=P,...,M ЦИКЛ T2[J]:=T2[J]+T[I,J];
A[]:=0;¯A[]:=0;
 ДЛЯ J:=1,...,N ЦИКЛ  ЕСЛИ K=1 ТО T2[J]:=T2[J]/M1 ИНАЧЕ T2[J]:=T2[J]/M2;
 ДЛЯ J:=1,...,N ЦИКЛ  ЕСЛИ K=1 ТО P1[J]:=T2[J]
 ИНАЧЕ P2[J]:=T2[J]; ДЛЯ J:=1,...,N ЦИКЛ B[J]:=1; ДЛЯ I:=P,...,M ЦИКЛ A[I]:=1;
¯M1:
 ДЛЯ I:=P,...,M ЦИКЛ *СИГМА*1[I]:=0; ДЛЯ I:=P,...,M ЦИКЛ  ДЛЯ J:=1,...,N ЦИКЛ  ЕСЛИ
K1=1 ТО
*СИГМА*1[I]:=*СИГМА*1[I]+((T[I,J]-T1[I])↑2)×B[J] ИНАЧЕ
*СИГМА*1[I]:=*СИГМА*1[I]+((T[I,J]-T2[J])↑2)×B[J];
 ДЛЯ I:=P,...,M ЦИКЛ ¯A[I]:=A[I]; ДЛЯ J:=1,...,N ЦИКЛ *СИГМА*2[J]:=0; ДЛЯ J:=1,...,N
 ЦИКЛ  ДЛЯ I:=P,...,M ЦИКЛ
 ЕСЛИ K1=1 ТО *СИГМА*2[J]:=*СИГМА*2[J]+((T[I,J]-T2[J])↑2)×A[I]
```

```
ИНАЧЕ *СИГМА*2[J]:=*СИГМА*2[J]+((T[I,J]-T1[I])↑2)*A[I];
S:=0; ДЛЯ I:=P,...,M ЦИКЛ S:=S+*СИГМА*1[I]; ДЛЯ I:=P,...,M ЦИКЛ A[I]:=*СИГМА*1[I]/S;
S:=0; ДЛЯ J:=1,...,N ЦИКЛ S:=S+*СИГМА*2[J]; ДЛЯ J:=1,...,N ЦИКЛ B[J]:=*СИГМА*2[J]/S;
ДЛЯ I:=P,...,M ЦИКЛ ЕСЛИ ABS(¯A[I]-A[I])≥0.0000001 ТО НА ¯M1;
¯O¯U¯T¯P¯U¯T(0,'2/','T','B','/','E',B);
¯O¯U¯T¯P¯U¯T(0,'2/','T','A','/','E',A);
ЕСЛИ K=1 ТО НАЧАЛО ДЛЯ J:=1,...,N ЦИКЛ *БЕТА*[1,J]:=SQRT(B[J]);
ДЛЯ I:=1,...,M1 ЦИКЛ *АЛЬФА*1[I]:=SQRT(A[I]) КОНЕЦ
ИНАЧЕ НАЧАЛО ДЛЯ J:=1,...,N ЦИКЛ *БЕТА*[2,J]:=SQRT(B[J]); ДЛЯ I:=1,...,M2 ЦИКЛ *АЛЬФА*
2[I]:=SQRT(A[I+M1]) КОНЕЦ ;
ЕСЛИ K1=1 ТО ДЛЯ J:=1,...,N ЦИКЛ ДЛЯ I:=1,2 ЦИКЛ *БЕТА*1[I,J]:=*БЕТА*[I,J];
ЕСЛИ K=1 ТО
¯O¯U¯T¯P¯U¯T(0,'/','T','*АЛЬФА*1','/','E',*АЛЬФА*1);
ЕСЛИ K=2 ТО НАЧАЛО
¯O¯U¯T¯P¯U¯T(0,'/','T','*АЛЬФА*2','/','E',*АЛЬФА*2);
¯O¯U¯T¯P¯U¯T(0,'/','T','*БЕТА*[I,J]','/','E',*БЕТА*) КОНЕЦ КОНЕЦ КОНЕЦ ;
ДЛЯ I:=1,2 ЦИКЛ ДЛЯ J:=1,...,N ЦИКЛ НАЧАЛО *БЕТА*[I,J]:=*БЕТА*[I,J]↑2;*БЕТА*1[I,J
]:=*БЕТА*1[I,J]↑2 КОНЕЦ ;
ДЛЯ J:=1,...,N ЦИКЛ ДЛЯ I:=1,...,M3 ЦИКЛ T3[I,J]:=(T3[I,J]-TMN[J])/(TMX[J]-TMN[J]);
```

```
ДЛЯ I:=1,...,M3 ЦИКЛ R1[I]:=0; ДЛЯ I:=1,...,M3 ЦИКЛ ДЛЯ J:=1,...,N ЦИКЛ R1[I]:=R1[
I]+T3[I,J];
ДЛЯ I:=1,...,M3 ЦИКЛ R1[I]:=R1[I]/N;
B1[]:=0;B2[]:=0;
ДЛЯ I:=1,...,M3 ЦИКЛ ДЛЯ J:=1,...,N ЦИКЛ B1[I]:=B1[I]+*БЕТА*1[1,J]×((T3[I,J]-R1[I]
)↑2);
ДЛЯ I:=1,...,M3 ЦИКЛ ДЛЯ J:=1,...,N ЦИКЛ B2[I]:=B2[I]+*БЕТА*1[2,J]×((T3[I,J]-R1[I]
)↑2);
R1[]:=0;
ДЛЯ I:=1,...,M3 ЦИКЛ R1[I]:=B1[I]/B2[I];
¯O¯U¯T¯P¯U¯T(0,'2/','T','R1','/','E',R1);
ДЛЯ I:=1,...,M3 ЦИКЛ НАЧАЛО B1[I]:=B2[I]:=0 КОНЕЦ ;
ДЛЯ I:=1,...,M3 ЦИКЛ ДЛЯ J:=1,...,N ЦИКЛ B1[I]:=B1[I]+*БЕТА*[1,J]×((T3[I,J]-P1[J])
↑2);
ДЛЯ I:=1,...,M3 ЦИКЛ ДЛЯ J:=1,...,N ЦИКЛ B2[I]:=B2[I]+*БЕТА*[2,J]×((T3[I,J]-P2[J])
↑2);
ДЛЯ I:=1,...,M3 ЦИКЛ R2[I]:=B1[I]/B2[I];
¯O¯U¯T¯P¯U¯T(0,'2/','T','R2','/','E',R2) КОНЕЦ КОНЕЦ ;
КОНЕЦ
```

П. КОМПЛЕКС ПРОГРАММ ИТЕРАЦИОННОГО МЕТОДА ЦЕЛЕВОГО КЛАССИФИЦИРОВАНИЯ И УПОРЯДОЧЕНИЯ ОБЪЕКТОВ

Алгоритмическое описание итерационного метода классифицирования и упорядочения объектов. ("Каскад П").

В последние годы, характеризующиеся широким внедрением математических методов в практику геологических исследований, все большее значение при решении задач прогнозного профиля приобретают методы распознавания образов. Одним из примеров реализации такого подхода является метод целевого классифицирования и упорядочения объектов ("Каскад-I") [I].

Несмотря на неплохие результаты при решении практических задач [2,3,4] "Каскад-I" обладает рядом недостатков, главным из которых является индивидуальный подход к оценке информативности признаков, хотя вполне очевидно, что в этой процедуре целесообразно учитывать также влияние на важность каждого отдельного признака присутствия всех остальных.

Настоящая статья посвящена описанию итерационного метода целевого классифицирования и упорядочения объектов ("Каскад-П"), позволяющего преодолеть отмеченный недостаток. В содержательном отношении предлагаемый метод является естественным развитием метода "целевое классифицирование и упорядочение объектов ("Каскад-I")". Блок-схема обработки информации по алгоритму "Каскад-П" показана на рис. I. В процедурном отношении описанный алгоритм можно рассматривать как совмещенную модификацию "метода согласованных оценок ("Качели")" [5, 6,7], "метода целевой итерационной классификации ("Цикл")"[8], "метода целевого классифицирования и упорядочения объектов ("Каскад-I")". Элементом, сближающим его с алгоритмом "Качели", является итерационный принцип поправки исходных информационных весов признаков с учетом структуры строк (цикл, состоящий из блоков 3,5 и 6 на рис. I). Элементом, сближающим его с алгоритмом "Цикл" является процедура вычисления поправочного коэффициента, используемого для поправки исходных информационных весов признаков (блок 4). Остальные блоки являются новыми (3а,4а,5а) или заимствованными из алгоритма "Каскад-I".

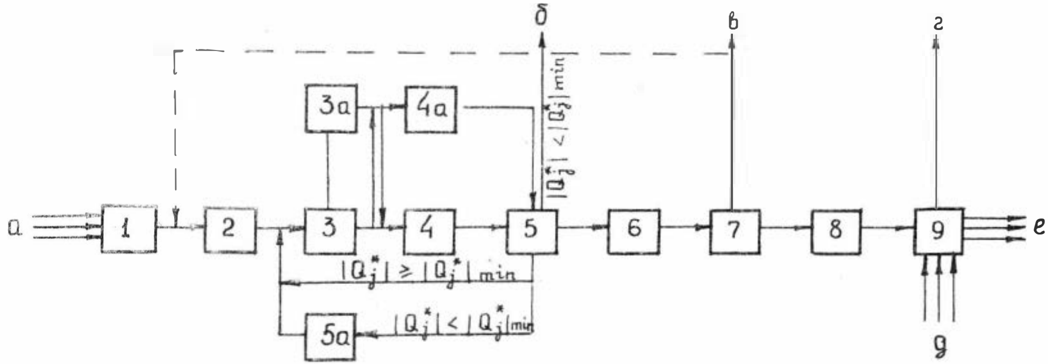


Рис. 1. Блок-схема алгоритма "Каскад-П".

Подготовительный этап: I – нормирование (бинарное кодирование) исходных данных; Этап обучения: 2 – вычисление исходных информационных весов признаков (Q_j^*); 3 – вычисление информационных весов строк ($J[Q_j^*]_i^*$); 3а – нормирование информационных весов строк ($J[Q_j^*]_i^{**}$); 4 – вычисление коэффициентов поправки для всех объектов (K_{iz} и K_{iz}^*); 4а – вычисление коэффициентов поправки для "худших" объектов ($K_{i\,xyд.\,z}$ и $K_{i,\,xyд.\,z}^*$); 5 – вычисление новых информационных весов признаков и их минимизация (б); 5а – смена знака информационных весов признаков; 6 – получение окончательных информационных весов признаков; этап оценки качества обучения: 7 – внутренний экзамен (ρ^e , ψ_{max} , $\tilde{\psi}_{max}, \psi_s$) и минимизация объектов обучения (в); этап оценки надежности распознавания: 8 – внешний экзамен ($\lambda_{кл}$ и $\lambda_{уп}$); этап распознавания: 9 – отбраковка (г) и распознавание объектов-проб.

а – информация об эталонах, д – информация о пробах, е – результаты распознавания.

Подготовка информации к обработке

Предлагаемый метод, как и "Каскад-I", позволяет обрабатывать исходный материал качественного, порядкового и количественного характера, а также смешанные таблицы, включающие информацию всех видов. При наличии в исходной таблице порядковых*) и количественных признаков для приведения их к сопоставимому виду производится нормировка вида:

$$t^*_{ij} = \frac{t_{ij} - t_{j\,min}}{t_{j\,max} - t_{j\,min}} \qquad (I)$$

где t_{ij} и t^*_{ij} – соответственно исходное и нормированное значения j -го признака i -ой строки; $t_{j\,min}$ и $t_{j\,max}$ – минимальное и максимальное значения j -го признака, встречающиеся в исходной таблице. Исходные значения целевого признака, независимо от его вида, нормируются следующим образом:

$$x^*_{i,n+1} = \frac{x_{i,n+1} - x_{n+1\,min}}{x_{n+1\,max} - x_{n+1\,min}} + C \qquad (2)$$

где C – некоторая константа, равная, например, 0,001. Добавление постоянной величины необходимо для того, чтобы в последующих процедурах избежать умножения на нуль. Если исходную таблицу необходимо обработать в бинарном виде, то вместо процедуры нормировки производится бинарное кодирование характеристических признаков по методике, описанной в [I].

После этих преобразований исходная информация готова к обработке описываемым методом.

Исходные информационные веса признаков

Исходные информационные веса признаков, в зависимости от вида решаемой задачи [I], подсчитываются по одной из нижеприведенных формул:

$$\check{P}_j = \frac{\sum\limits_{s=1}^{L-1}\sum\limits_{p=s+1}^{L}\sum\limits_{i=1}^{m_s}\sum\limits_{i'=1}^{m_p}(t^{*s}_{ij} - t^{*p}_{i'j})(x^{*s}_{i,n+1} - x^{*p}_{i',n+1})}{m_1 \cdot m_2 \cdot \ldots \cdot m_L} \qquad (3)$$

$$\varphi_j = \frac{\sum\limits_{i=1}^{m}\sum\limits_{i'=1}^{m}(t^*_{ij} - t^*_{i'j})(x^*_{i,n+1} - x^*_{i',n+1})}{m^2} \qquad (4)$$

---

*) Подразумевается частный случай шкалы порядка, когда признаки принимают значения на заранее заданном конечном множестве (например, "малая", "средняя", "сильная" степень проявленности).

где P_j^{v} – оценка для решения задачи классифицирования ($L \geq 2$); φ_j – оценка для решения задачи упорядочивания ($L = I$); t_{ij}, t_{ij}^{*} и $x_{i,n+1}^{*}$, $x_{i',n+1}^{*}$ – нормированные значения характеристических и целевого признаков для объектов эталонных классов; $m_1, \ldots, m_L$ – число объектов в эталонных классах; L – число эталонных классов; m – число объектов в упорядоченном наборе (при $L = I$).

Указанные оценки информативности являются своеобразными мерами корреляционной зависимости между целевым и характеристическим признаками, смысловое содержание которых наиболее отчетливо раскрывается для случая бинарных признаков при $L \leq 2$ [I].

В принципе, исходные информационные веса признаков могут задаваться произвольным вектором с ненулевыми компонентами, поскольку окончательные информационные веса, а также их знак, вычисляются по предлагаемому алгоритму автоматически. Но решение контрольных примеров и практических задач показало,что итерационный процесс сходится значительно быстрее, если исходные веса признаков задавать не произвольным образом, а по формулам (3,4).

Вычисленные начальные веса нормируются по формуле:

$$Q_j^{*} = Q_j \Big/ \sum_{j=1}^{n} |Q_j| , \tag{5}$$

где Q_j – значение любой из двух оценок, привлеченных для решения задачи. Такая нормировка позволяет сравнивать информационные веса на различных итерациях и оказывается удобной при процедуре минимизации признакового пространства.

Подсчет строчечных нагрузок

Подсчет строчечных нагрузок (информационных весов строк) производится по формуле:

$$J[Q_j^{*}]_i = \sum_{j=1}^{n} |Q_j^{*}| \cdot \hat{t}_{ij} \tag{6}$$

при $\hat{t}_{ij} = t_{ij}^{*}$, если $Q_j^{*} > 0$ и $\hat{t}_{ij} = I - t_{ij}^{*}$, если $Q_j^{*} < 0$. Информационный вес строки $J[Q_j^{*}]$ служит показателем проявленности целевого свойства у i-го объекта, что используется при распознавании объектов-проб. Можно выделить два способа оценки соответствия вычисленных строчечных нагрузок значениям целевого признака.

I. Со значениями целевого признака сравниваются нормированные по формуле (2) нагрузки строк:

$$x^*_{i,n+1} - J[Q^*_j]^{**}_i = \psi, \text{ где } J[Q^*_j]^{**}_i = \frac{J[Q^*_j]_i - J[Q^*_j]_{min}}{J[Q^*_j]_{max} - J[Q^*_j]_{min}} + C \quad (7)$$

Назовем такой способ "оценкой относительного соответствия".

2. Со значениями целевого признака сравниваются ненормированные нагрузки строк:

$$x^*_{i,n+1} - J[Q^*_j]^*_i = \psi_i, \text{ где } J[Q^*_j]^*_i = J[Q^*_j]_i + C \quad (8)$$

Назовем такой способ "оценкой абсолютного соответствия".

Очевидно, что величина $J[Q^*_j]^*_i$ является частным случаем $J[Q^*_j]^{**}_i$, но вовлечение в дальнейший цикл обработки той или иной величины приводит к различным результатам.

Вычисление коэффициентов соответствия

Коэффициент соответствия k (коэффициент поправки) вычисляется согласно работе [8], как

$$k = \frac{x^*_{i,n+1}}{J^н_i} \quad (9)$$

где $J^н_i$ - строчечная нагрузка $J[Q^*_j]^*_i$ или $J[Q^*_j]^{**}_i$. Он оценивает степень совпадения полученных нагрузок строк с соответствующими значениями целевого признака и используется в последующих итерациях для поправки информационных весов признаков с целью улучшения этого совпадения.

Следует указать два пути улучшения соответствия строчечных нагрузок целевому признаку.

I. Коэффициенты поправки вычисляются лишь для объектов с наихудшим соответствием с целевым признаком, в качестве которого выступает объект с максимальной по абсолютной величине ψ_i. В этом случае $k_{i\,худ.r.} = x^*_{i\,худ.n+1} / J[Q^*_j]^*_{i\,худ.}$ ("стремление к абсолютному соответствию")

$k^*_{i\,худ.r.} = x^*_{i\,худ.n+1} / J[Q^*_j]^{**}_{i\,худ}$ – ("стремление к относительному соответствию") r - номер итерации.

Для всех остальных объектов коэффициенты поправки считаются рав - ными единице. Таким образом, последующая поправка информационных весов признаков будет производиться лишь с учетом значений признаков, характеризующих "худшие" объекты. При решении задачи классифицирования минимальное число таких объектов равно L, где L - число классов, при решении задачи упорядочения - одному, хотя возможны случаи, когда их число может оказаться большим (присутствие сразу нескольких объектов с одинаково плохим соответствием).

2) Коэффициенты поправки вычисляются сразу для всех объектов обучения. В этом случае

$$K_{ir}=\frac{x^{*}_{i,n+1}}{\text{Э}[Q_j^{*}]_i^{*}} \quad \text{("стремление к абсолютному соответствию")}$$

$$i = 1,2,\dots,n$$

$$K^{*}_{ir}=\frac{x^{*}_{i,n+1}}{\text{Э}[Q_j^{*}]_i^{**}} \quad \text{("стремление к относительному соответствию")}$$

Таким образом, при данном подходе последующая поправка информационных весов признаков производится с одновременным учетом всех значений признаков, встречающихся в таблице обучения.

Вышеизложенные способы улучшения соответствия строчечных нагрузок значениям целевого признака можно отобразить следующей схемой:

Рис. 2.

Отметим, что с формальной точки зрения для решения задач классифицирования и упорядочения пригодны все четыре варианта. Однако с содержательных позиций при решении задач классифицирования, в которых главной целью является получение наилучшего разделения классов, наиболее целесообразны способы 2 и 4, что и подтверждается практически. При решении задач упорядо - чения целесообразнее использовать способы I и 3, однако качество результатов в этом случае во многом зависит от объема исходной информационной таблицы.

Вычисление новых информационных весов признаков и строчечных нагрузок

Выявленное несоответствие величин строчечных нагрузок, вычисленных по начальным информационным весам признаков, значениям целевого признака (при условии, что признаковое пространство отвечает целевому группированию объектов) частично объясняется учетом лишь индивидуальных разделительных способностей признаков без учета их взаимосвязи. Поэтому производится коррекция исходных информационных весов признаков в соответствии с коэффициентами поправки:

$$Q_{j\,r+1} = Q^*_{jr} \sum_{i=1}^{m} \hat{t}_{ij} \cdot \tilde{K}_{ir} \Big/ \sum_{i=1}^{m} \hat{t}_{ij} \qquad \text{(I0)}$$

где K_{ir} – любой из четырех предложенных коэффициентов поправки, соответствующий выбранному режиму решения задачи. После нормирования новых информационных весов признаков по формуле (5) производится подсчет нагрузок строк следующей итерации и вычисление новых поправочных коэффициентов (6–9).

Минимизация признакового пространства

Реализация итерационного метода "Каскад-П" позволяет, как и в других аналогичных итерационных алгоритмах, производить минимизацию признакового пространства в автоматическом режиме. Задается порог существенности информационного веса признака $|Q^*_j|_{min} = \frac{\beta}{n_r}$, где n_r – число признаков с ненулевым весом на r-ой итерации, а β – некоторая постоянная, равная, например, 0,0I. При достижении каким-либо признаком этого минимального допустимого значения, он удаляется из исходной таблицы и в последующих итерациях не учитывается. Меняя величину β, а, следовательно, и пороговое значение $|Q^*_j|_{min}$, можно контролировать ход процедуры минимизации. Минимизация признакового пространства начинает осуществляться не сразу, а с некоторой заранее заданной (например, с двадцатой) итерации. Это условие введено для того, чтобы избежать удаления на первом же шаге признаков, получивших запороговые начальные информационные веса.

Описанная процедура минимизации, как правило, приводит к значительному сокращению исходного признакового пространства с одновременным улучшением результатов обучения.

Решение практических примеров показывает, что для минимизации признакового пространства более целесообразно использовать следующий прием. При достижении некоторым признаком порогового значения информативности, знак его информативности меняется на противоположный. Исключение этого признака из дальнейшего рассмотрения производится лишь в том случае, если по истечении некоторого заранее заданного числа итераций (например, 20) его информативность, несмотря на смену знака, вновь становится по абсолютной величине ниже порогового значения $|Q_j^*|_{min}$. Этот прием позволяет осуществлять в автоматическом режиме не только подбор оптимального набора признаков и их информационных весов, но и выбирать знак информационных весов.

Получение конечных информационных весов и оценка качества обучения (внутренний экзамен)

Описанный итерационный процесс, соответствующий этапу обучения, в результате которого вычисляются конечные информационные веса признаков, прекращается при выполнении одного из следующих условий:

1. Изменение информационного веса каждого признака на последующей итерации не превышает некоторого заранее заданного предела α (например, $\alpha = 1 \cdot 10^{-5}$);

2. Произведено заранее заданное число итераций D (например, $D = 300$);

3. Качество обучения достигает достаточного для решения данной задачи уровня*).

Естественно, первые два условия являются более предпочтительными, поскольку заранее предугадать достаточный и возможный на заданной исходной информации уровень качества обучения вряд ли всегда возможно.

Для оценки качества обучения на любой итерации в задачах классифицирования используется величина

---

*) Близкие по смыслу условия реализованы в алгоритме "Цикл" [8].

$$\rho^{\ell} = \mathcal{J}[Q_j^*]^{*s}_{min} - \mathcal{J}[Q_j^*]^{*s+1}_{max} , \tag{II}$$

где $\mathcal{J}[Q_j^*]^{*s}_{min}$ – минимальная величина строчечной нагрузки среди объектов s -го класса, $\mathcal{J}[Q_j^*]^{*s+1}_{max}$ – максимальная величина строчечной нагрузки среди объектов s +I-го класса. Величина ρ^{ℓ} является показателем степени "растяжки" классов и чем она больше, тем более уверенно различаются классы эталонных объектов, тем выше считается качество обучения.

Для оценки качества обучения в задачах упорядочения используются два показателя:

$$\psi_{max} = \max_{i=1,\dots,m}\{|x_{i,n+1} - \mathcal{J}_i^H|\}$$
$$\psi_s = \sum_{i=1}^{m} |x^*_{i,n+1} - \mathcal{J}_i^H| . \tag{I2}$$

Величина ψ_{max}, определяющая максимальную зону несоответствия, указывает максимальное отклонение i -ой величины строчечной нагрузки от соответствующего значения целевого признака. Величина ψ_s определяет суммарное отклонение величин $\mathcal{J}_i^H$ от значений $x^*_{i,n+1}$ по всем объектам обучения. Естественно, что понижение этих коэффициентов характеризует улучшение качества обучения.

Минимизация объектов обучения

Вычисление показателей ψ_{max} и ρ^{ℓ} при окончательных информационных весах позволяет выявить в обучающей выборке объект, величина строчечной нагрузки которого обладает наихудшим соответствием со значением целевого признака. При большом ψ_{max} или малом (или отрицательном) ρ^{ℓ}, не позволяющим получить желаемую точность решения задачи, этот объект следует удалить из материала обучения и произвести переобучение по изложенной выше схеме (см. рис.I). Этот прием приводит, как правило, к улучшению качества обучения, но требует дополнительных ограничений на допуск объектов-проб к распознаванию на оставшемся материале обучения.

Внешний экзамен

Оценка надежности распознавания производится по процедуре внешнего экзамена. В алгоритме "Каскад-П" наряду с общепринятым способом внешнего экзамена [9] предусмотрен другой способ, когда последовательно каждый из объектов обучения (или выборочные объекты) удаляется из обучающей выборки и распознается после предварительного переобучения. Решение и в том и в другом случаях принимается в соответствии со значением величины $Э^{н}_{iэк}$ по правилам, сформулированным в работе [I]. Правильное отнесение к классу (в задачах классифицирования) оценивается единицей, неправильное - нулем, а попадание в зону неопределенности (отказ от распознавания) - 0,5. Окончательная оценка надежности распознавания проб находится как

$$\lambda_{кл} = \frac{\sum_{i=1}^{B} \lambda_i \cdot 100\,\%}{B} \tag{I3}$$

где λ_i - результат распознавания i-го эталонного объекта, B - число объектов экзамена. В задачах упорядочения правильно распознанным считается объект, для которого величина строчечной нагрузки $Э^{н}_{iэк}$ не выходит за пределы интервала $x^{*}_{i,n+1} \pm \tilde{\varphi}_{max}/2$, где $\tilde{\varphi}_{max} = \varphi_{max}$ в случае отсутствия пространственного пересечения или соприкосновения зон несоответствия, и $\tilde{\varphi}_{max} = |\varphi_i \cup \varphi_{i+1}|_{max}$ если присутствуют зоны, имеющие области пересечения. В противном случае распознавание считается неправильным. Окончательная оценка надежности распознавания проб при точности распознавания равной $\pm \tilde{\varphi}_{max}/2$ вычисляется как

$$\lambda_{уп} = \frac{\sum_{i=1}^{B} \lambda_i \cdot 100\,\%}{B} \tag{I4}$$

где $\lambda_i = I$ при правильном распознавании и $\lambda_i = 0$ - при неправильном.

Отбраковка и распознавание проб

К распознаванию на заданном материале обучения с вычисленными надежностью и точностью допускаются пробы, принадлежащие (или близкие) заданным на обучение классам объектов. В случае бинарной исходной информации разбраковка проб на пригодные и непригодные к распознаванию может быть осуществлена по процедуре логического анализа проб по блоковой структуре материала обучения [I]. В случае количественной или смешанной исходной информации следует иметь ввиду, что с формальной точки зрения к распознаванию могут быть допущены лишь те пробы, у которых значения всех признаков не выходят за пределы интервалов значений соответствующих признаков у объектов обучения. После отсеивания проб, не удовлетворяющих указанному требованию, оставшуюся информацию рекомендуется, посредством процедур кодирования, привести к бинарному виду и следующий этап отбраковки осуществить путем логического анализа блоковости таблицы. Очевидно, что в процедуре отбраковки проб существенную роль играет профессиональная оценка важности каждого используемого при отбраковке признака.

Распознавание оставшихся после отбраковки проб производится по процедуре подсчета строчечных нагрузок. Окончательное решение об отнесении к классу или о степени проявленности целевого признака у пробы принимается по соотношению величины строчечной нагрузки распознаваемого объекта с аналогичными величинами объектов обучения [I]. Оценкой надежности результатов распознавания в задачах классифицирования служит величина $\lambda_{кл}$, а в задачах упорядочения – $\lambda_{уп}$. Интервал ожидаемых значений целевого признака у пробы в последнем случае определяется по формуле:

$$x_{n+1,\,пр.} = \left[\left(J^{н}_{i\,пр} - C \right) \left(x_{n+1\,max} - x_{n+1\,min} \right) + x_{n+1\,min} \right] \pm \pm \frac{\tilde{\varphi}_{max} \left(x_{n+1\,max} - x_{n+1\,min} \right)}{2} . \qquad (I5)$$

Описанный алгоритм решения прогнозных задач позволяет обрабатывать большие массивы данных. Результаты распознавания представляются в легко воспринимаемой форме. "Каскад-П" снабжен комплексом программ для ЭВМ М-220, М-222, БЭСМ-4 и БЭСМ-4М, описанных в следующей статье сборника.

Литература

1. Бабич В.В., Федосеев Г.С. Метод целевого классифицирования и упорядочения объектов ("Каскад-I"). - В кн.: Логико-математическая обработка геологической информации. Новосибирск, 1976, с.42-70.

2. Бабич В.В., Федосеев Г.С. Прогнозная оценка железорудных объектов Кондомского района Горной Шории. - В кн.: Логико-математическая обработка геологической информации. Новосибирск, 1976, с.101-117.

3. Каштанов В.А., Соколов А.Д. Прогноз продуктивности локальных поднятий до их ввода в бурение (на примере мезозойско-кайнозойских отложений севера Западно-Сибирской плиты). - В кн.: Логико-математическая обработка геологической информации. Новосибирск, 1976, с.133-151.

4. Дмитриев А.Н., Федосеев Г.С., Бабич В.В. Логико-математические методы в прогнозировании железорудных месторождений. - В кн.: Рудные формации и геохимия рудообразующих процессов. Новосибирск, 1976, с.220-227.

5. Васильев Ю.Л., Дмитриев А.Н. Спектральный подход к сравнению объектов, охарактеризованных набором признаков. - "ДАН СССР", 1972, т.206, № 6, с.1309-1312.

6. Васильев Ю.Л., Дмитриев А.Н. Простой способ сравнения объектов, охарактеризованных набором признаков. - В кн.: Применение математических методов и ЭВМ для решения прогнозных задач нефтяной геологии (краткие тезисы докладов и конференции 12-14 февраля 1973 г.). Новосибирск, 1973, с.60-63.

7. Дмитриев А.Н., Красавчиков В.О. Программы метода согласованных оценок. - В кн.: Логико-математическая обработка геологической информации. Новосибирск, 1975, с.6-13.

8. Бишаев А.А. Метод "Целевая итерационная классификация" ("Цикл"). - В кн.: Логико-математическая обработка геологической информации. Новосибирск, 1976, с.70-92.

9. Константинов Р.М., Королева З.Е., Кудрявцев В.Б. О комбинаторно-логическом подходе к задачам рудоносности. - В кн.: Проблемы кибернетики, вып.31, изд."Наука", М.,1976,с.5-33.

КОМПЛЕКС ПРОГРАММ (ПI – П6).

Описываемый комплекс включает шесть программ (ПI–П6), написанных на α-языке для ЭВМ типа М-220, М-222, БЭСМ-4 и БЭСМ-4М. Термин "комплекс" применен в данном случае для обозначения совокупности программ, последовательное применение которых обеспечивает решение задачи в рамках конкретного метода, но отличается от классического понимания комплекса отсутствием автоматического оператора, что обусловлено техническими особенностями перечисленных выше типов ЭВМ. Набор программ, вошедших в комплекс , разделен на две группы – вспомогательные программы (ПI–П4) и основные (П5 и П6). Вспомогательные программы "Запись", "Транспонирование", "Выбор" и "Вывод" служат для: I) записи исходной информации на МЛ с целью ее долгосрочного хранения; 2) транспонирования исходной матрицы в том случае, если перфорация произведена по строкам, а не по столбцам (признакам), как того требуют основные программы; 3) выбора из исходной таблицы любой подтаблицы; 4) контроля вводимой с МЛ для счета по основным программам выбранной информации. Вспомогательные программы небольшие по объему и просты в обращении. Их использование снимает необходимость многократной перфорации исходных данных при изменении постановки задачи, уменьшает время ввода данных в ЭВМ, позволяет принимать к обработке информацию независимо от вида ее перфорации.

В группу основных входят две программы – "Обучение" и "Распознавание", реализующие основные процедуры итерационного метода целевого классифицирования и упорядочения объектов ж).Программа "Обучение" позволяет производить решение задачи в различных постановках (одноклассовая, двуклассовая, многоклассовая) и режимах ("абсолютное" или "относительное" соответствие, по всем или по "худшим" объектам и т.д.), которые задаются набором условных чисел. Начальные информационные веса признаков вычисляются по формуле

$$Q_j = \sum_{s=1}^{L-1} \sum_{p=s+1}^{L} \sum_{i=1}^{m_s} \sum_{i'=1}^{m_p} (t_{ij}^{*s} - t_{i'j}^{*p})(x_{i,n+1}^{*s} - x_{i',n+1}^{*p}).$$

ж) См. статью В.В.Бабича в настоящем сборнике.

Время решения по этой программе матрицы размером 53 x I5I при заданном числе итераций D = 250 и L =2 составило около I часа 40 минут. Программа "Распознавание" позволяет по результатам обучения производить распознавание объектов-проб и объектов экзамена.

Общей особенностью всех программ является способность обрабатывать значительные по объему матрицы ($m \times N$ = 48000 чисел) имеющиеся в публикуемых программах ограничения не носят принципиального характера и легко заменимы. Перфорация исходных данных должна производиться на цифровом перфорационном устройстве для ЭВМ М-222. При эксплуатации программ следует иметь в виду, что в них использована процедура обращения к МБ, применяемая в ВЦ ИГиГ СО АН СССР. Все программы апробированы при решении контрольных тестов и практических задач.

Программа П I.

"Запись"

Назначение.

Программа "Запись" предназначена для размещения исходной информации на магнитной ленте. Ввод исходных данных производится с перфокарт.

Инструкция к пользованию.

Порядок постановки перфокарт.

I. α - схема или рабочая программа.

2. ijk - начальный номер зоны на магнитной ленте.

3. N - число вводимых массивов.

4. m - длина вводимых массивов.

5. a - массивы.

На печать выводится: I. ijk ; 2. N ; 3. m .

Запись производится на первый программный магнитофон.

Текст программы

```
начало целый N, ¯N , IJK, J ;   целый M ;   массив
A [I:I00] ;   ввод (IJK, ¯N, M);   вывод (IJK, ¯N ,M,
истина); N : = - I; ¯M: N:= N + I;   ввод (A); ¯C¯ПОI75
(002I, A[I], A[M],I, IJK + N );   если N <¯N - I то на ¯M; конец*
```

Программа П 2

"Транспонирование"

Назначение.

Программа предназначена для транспонирования матрицы ис -

ходных данных в том случае, если перфорация вводимой информации произведена по строкам. Исходная информация должна быть записана на МЛ, транспонированная матрица также записывается на МЛ.

Инструкция к пользованию.

Порядок постановки перфокарт.

I. α - схема или рабочая программа.

2. ijk - начальный номер зоны считывания на МЛ.

3. ij - начальный номер зоны записи транспонированной матрицы.

4. N - длина строки нетранспонированной матрицы.

5. m - количество строк нетранспонированной матрицы.

На печать выводится: I. ijk ; 2. N ; 3. m. Считывание исходной информации производится с нулевого программного магнитофона, а запись - на первый программный магнитофон.

Текст программы.

```
начало_ целый I ,J , K ,IJK,IJ ,¯N,M;  вещественный_в;
массив_A [1:250] ;  ввод (IJK,IJ ,¯N ,M);  вывод (IJK,
IJ ,¯N ,M, истина_);  для I := 1,...,M цикл_начало  ¯C¯ПО
175 (0016, A[1] ,A [¯N],0,IJK +I-1);  ¯M¯З¯У (A,1,¯N ,0,4,
¯N×(I -1) + 1,48,0);  конец ;  для K:= I,...,¯N цикл_
начало_  для J :=1,...,M цикл_  ¯M¯З¯У (A, J , J ,0,4,
¯N × ( J -1) + K,32,0):  ¯C¯ПОI75 (002I,A[1],A[M] ,1, IJ +
K - I) конец конец_*
```

Программа П-3 "Выбор"

Назначение.

Программа предназначена для выбора из исходной матрицы любой подматрицы.

Инструкция к пользованию.

Порядок постановки перфокарт.

I. α - схема или рабочая программа.

2. N - число столбцов подматрицы.

3. m - число строк подматрицы.

4. m_1 - число строк исходной матрицы.

5. Код - порядковые номера столбцов исходной матрицы, подлежащих выбору.

6. Ков - порядковые номера строк исходной матрицы, подлежащих выбо-

7. ijk - начальный номер зоны считывания на МЛ.

8. ij - начальный номер зоны записи подматрицы.

На печать выводится: I. N ; 2. m; 3. m_1 ; 4. ijk; 5. ij ; 6. Код; 7. Ков.

Использование этой программы оказывается полезным в том случае, если в процессе решения задачи возникает необходимость в ее перестановке, влекущей за собой перекомпоновку исходных данных.

Текст программы.

начало целый I , J , N , IJK , IJ ,M,M1; целый ¯N ; целый массив код [1:250] , ков [1:100] ; массив A[1:100] ; массив B [1:100] ; ввод (¯N ,M, M1, код, ков, IJK , IJ); вывод (¯N,M, M1, IJK, IJ); ¯C¯ПОI76 (320I, код[I], код [¯N], 0,0); ¯C¯ПОI76 (320I, ков[1], ков[M],0,0); вывод (истина); для I:=1,...,¯N цикл начало N := IJK + код[I]- - I; ¯C¯ПОI75 (0016,A[1], A[M1] ,0, N); для J :=I,...,M цикл B [J] :=A[ков [J]] ; ¯C¯ПОI75(002I, B[1], B[M], 1, IJ + I -I) конец конец *

Программа П-4 "Вывод"

Назначение.

Данная программа предназначена для вывода хранящейся на МЛ информации и служит для контроля обрабатываемой исходной транспонированной или выбранной матрицы.

Инструкция к пользованию.

Порядок постановки перфокарт.

I. α -схема или рабочая программа.

2. ijk - начальный номер зоны считывания на МЛ.

3. m - количество строк.

4. N - количество столбцов.

На печать выводится контролируемая матрица.

Считывание производится с первого программного магнитофона.

Текст программы.

начало целый I ,¯N, IJK,M; массив A[1:250] ; ввод (IJK, M,¯N); для I :=0,...,¯N -1 цикл начало ¯C¯ПО175 (0016, A [1] ,A[M], 1,IJK+1); ¯C¯ПО176 (1153, A[1] , A[M],0,0) конец; конец *

Программа П-5

"Обучение"

Назначение.

Программа предназначена для вычисления информационных весов и минимизации исходного набора признаков с целью получе -

ния наилучшего соответствия нагрузок строк со значениями целевого параметра. Программа рассчитана на работу в различных режимах. Задание режима определяется набором условных чисел.

Инструкция к пользованию.

Порядок постановки перфокарт.

I. α -схема или рабочая программа.

2. ус. - условное число, определяющее стремление к "относительному" (ус$\neq$ I) или "абсолютному" (ус = I) соответствию с целевым признаком. 3. ус. I- условное число, определяющее обращение (ус.I $\neq$ I) или необращение (ус. I=I) знака информативности признаков.

4. ус. 2 - условное число, определяющее способ введения исходной информации: ус.2 = I - ввод с перфокарт, ус. 2 $\neq$ I - ввод с МЛ.

5. ус. 3 - условное число, определяющее способ достижения наилучшего соответствия нагрузок строк со значениями целевого признака: ус. 3 $\neq$ I - по "худшим" объектам, ус. 3= I - по всем объектам.

6. С -регуляризирующая константа.

7. α - условие прекращения процесса итерирования по сходимости информационных весов признаков.

8. β - порог зануления или смены знака информационных весов признаков.

9. D - задаваемое число итераций.

10. $maxit$ - число итераций, в течение которого у информационного веса признака не происходит смены знака, несмотря на достижение им порога β .

II. m - число объектов.

I2. N - число признаков.

I3. x - целевой признак.

I4. ijk - начальный номер зоны считывания с МЛ.

I5. ij - число итераций, определяющее частоту вывода промежуточных результатов на АЦПУ.

I6. L - число заданных классов обучения.

I7. d - массив границ классов*).

I8. $mixit$ - число итераций, в течение которых не происходит зануление информационных весов признаков после обращения

---

*) Пример: в случае m =9, L =3 и m_1=I,...,3; m_2=4,...,6; m_3==7,...,9; массив d имеет вид: 0,3,6,9.

их знака_ .

I9. t – массивы исходной информации (при ус.2=I).

На печать выводится:

I. ус.; 2. ус.-I; 3. ус.-2; 4. ус.-3; 5. C ; 6. α ; 7. β ; 8. D; 9. $maxit$; I0. N ; II. ijk ; I2. m ; I3. ij ; I4. L ; I5. $mixit$; I6. x ; I7. d ; I8. нормированный целевой признак (x^*_{n+1}); I9. t_{jmin} и t_{jmax} ; 20. номер итерации (z); 2I. массив информационных весов (Q^*_{jz}); 22. массив строчечных нагрузок (J^H_i); 23. φ_{max}, φ_3 , номер строки (m_i), определяющий φ_{max} (при L=I); ρ^ℓ, номера строк ($m_i, \dots, m_{iL}$), определяющих ρ^ℓ соседних классов (при L > I); 24. В конце работы программы повторяются выводы 20–23; 25. J^H_{min} и J^H_{max} (в случае ус.≠ I).

Программа "Обучение" иллюстрируется результатами решения небольшого контрольного примера в различных постановках и режимах:

__Задача № I__ – задача классифицирования при L =2, m_1=5 и m_2= =4; целевой признак x^I_{n+1} ; режим – стремление к "абсолютному" соответствию по "худшим" объектам; 50 итераций.

__Задача № 2__ – задача классифицирования при L =3, $m_1 = m_2 = m_3$= =3; целевой признак x^{II}_{n+1}; режим – стремление к "абсолютному" соответствию по всем объектам; 50 итераций.

__Задача № 3__ – задача классифицирования при L =3, $m_1 = m_2 = m_3$= =3; целевой признак x^{II}_{n+1}; режим – стремление к "абсолютному" соответствию по "худшим" объектам; 50 итераций.

__Задача № 4.__ – задача упорядочения при L =I, m =9, целевой признак x^{III}_{n+1}; режим – стремление к "относительному" соответствию по "худшим" объектам; 50 итераций.

__Контрольный пример__

№	I	2	3	4	5	x^I_{n+1}	x^{II}_{n+1}	x^{III}_{n+1}
I	8	0,5	I02	I	5	I	3	0,84
2	I7	0,5	80	I	2	I	3	I,00
3	9	0,4	40	0	4	I	3	0,38
4	I3	0,I	54	I	3	I	2	0,I8
5	I0	0,2	79	I	6	I	2	0,40
6	I2	0,3	32	0	0	0	2	0,02
7	4	0,2	5I	I	4	0	I	0,08
8	8	0,4	94	0	3	0	I	0,I5
9	I4	0,5	68	I	I	0	I	0,80

Выдача для задачи № I

Начальные условия:

ус	ус-I	ус-2	ус-3	C	α	β	D	maxit	N	ijk	m	ij	L	mixit
I	3	I	2	0,0I	0,00005	0,075	50	I0	5	I00	9	I0	2	I0

X: I,I,I,I,I,0,0,0,0;

d: 0,5,9;

x^*_{n+1} :I.0I, I.0I, I.0I, I.0I, I.0I ,0.0I, 0.0I,0.0I,0.0I;

$t_{j\,min}$: 4, 0.I, 32, 0, 0;

$t_{j\,max}$: I7, 0.5, I02, I, 6;

Результаты счета:

z =I0

Q^*_j:+0.2992; -0.0095; +0.0930; +0.0527; +0.5453.

$J[Q^*_j]^*_i$:0.702; 0.607;0.50I;0.58I;0.8I5; 0.I98; 0.458;0.459;0.43I.

ρ^ℓ : +0.042I; m_i^I: 3; m_i^{II}: 8.

. . . .

. . . .

z=50

Q^*_j:+0.35I2; 0; -0.04I5; 0; +0.6072.

$J[Q^*_j]^*_i$:0.603; 0.604; 0.574; 0.592; 0.77I; 0.29I; 0.4I7,0.4I9; 0.425.

ρ^ℓ: +0.I490; m_i^I : 3; m_i^{II} : 9.

Таким образом, в процессе решения исходное признаковое пространство минимизировалось до 3 признаков, классы полностью разделились, величина растяжки увеличилась с +0.042I (при z =I0) до +0.I490 (при z=50).

Выдача для задачи № 2

Начальные условия:

ус	ус-I	ус-2	ус-3	C	α	β	D	maxit	N	ijk	m	ij	L	mixit
I	3	I	I	0,0I	0,00005	0,075	50	I0	5	I00	9	I0	3	I0

X : 3, 3, 3, 2, 2, 2, I, I, I;

d: 0, 3, 6, 9;

x^*_{n+1} : I.0I; I.0I; I.0I; 0.5I; 0.5I; 0.5I; 0.0I; 0.0I; 0.0I;

$t_{j\,min}$: 4, 0.I, 32, 0, 0;

$t_{j\,max}$:I7, 0.5, I02, I, 6;

Результаты счета:

z = I0

Q_j^*: +0.4607; +0.2569 ; +0.0I54; +0.I9 x $\mathrm{I0}^{-\mathrm{II}}$, +0.2668;

$J[Q_j^*]_i^*$:0.646;0.827;0.559;0.467;0.564;0.422;0.256;0.49I;0.673;

ρ^{ℓ}_{I-II} :-0.0044; ρ^{ℓ}_{II-III} :-0.25I7; m_i^{I} :3; m_i^{I} :6; m_i^{II} :5; m_i^{III} :9;

. . . .

. . . .

z =50

Q_j^* : +0.4957; +0.I430; 0; -0.0523; +0.3088;

$J[Q_j^*]_i^*$:0.563;0.752;0.565;0.507;0.583;0.438;0.25I;0.476;0.587;

ρ^{ℓ}_{I-II} :-0.0I96; ρ^{ℓ}_{II-III} :-0.I482; m_i^{I} :I; m_i^{II} :6; m_i^{II} :5; m_i^{III} :9;

Выдача для задачи № 3

Начальные условия:

ус	ус-I	ус-2	ус-3	c	α	β	D	maxit	N	ijk	m	ij	L	mixit
I	3	I	2	0,0I	0,00005	0,75	50	I0	5	I00	9	I0	3	I0

x : 3, 3, 3, 2, 2, 2, I, I, I;

d : 0, 3, 6, 9;

x^*_{n+1} :I.0I; I.0I; I.0I; 0.5I; 0.5I; 0.5I;0.0I; 0.0I; 0.01;

$t_{j\,min}$: 4, 0.I, 32, 0, 0;

$t_{j\,max}$:I7, 0.5, I02, I, 6;

Результаты счета:

z =I0

Q_j^* :0.2348; +0.2982; +0.0275; +0.157 x $\mathrm{I0}^{-\mathrm{II}}$; +0.4394;

$J[Q_j^*]_i^*$:0.774;0.708;0,62;0.40;0.65;0.303;0.384;0.55; 0.576;

ρ^{ℓ}_{I-II} :-0.307; ρ^{ℓ}_{II-III} :-0.2726; m_i^{I} :3; m_i^{II} :6; m_i^{II} :5; m_i^{III} :9;

. . . .

. . . .

z =50

Q_j^* :+0.2688; +0.I852; -0.0627; -0.0542; +0.4289;

$J[Q_j^*]_i^*$:0.635;0.626;0.648;0.453;0.629;0.385;0.387;0.507;0.504;

ρ^{ℓ}_{I-II} :-0.003I; ρ^{ℓ}_{II-III}:-0.I225, m_i^{I} :2; m_i^{II} :6; m_i^{II} :5; m_i^{III} :8;

Сравнение результатов разделения трех классов в режимах "по всем объектам" (задача № 2) и "по худшим объектам" (задача № 3) показывает, что во втором случае достигнуто лучшее ка – чество обучения.

Выдача для задачи № 4

Начальные условия:

ус	ус-I	ус-2	ус-3	C	α	β	D	$maxit$	N	ijk	m	ij	L	$mixit$
9	3	I	2	0,0I	0,00005	0,075	50	I0	5	I00	9	I0	I	I0

X: 0.84; I.0; 0.38; 0.I8; 0.4; 0.02; 0.08; 0.I5; 0.8;

a: 0.9;

$x^*_{пн}$: 0.84; I.0I; 0.37; 0.I7; 0.39; 0.0I; 0.07; 0.I4; 0.80;

$t_{j\,min}$: 4, 0.1; 32; 0, 0;

$t_{j\,max}$: I7, 0.5, I02, I, 6;

Результаты счета:

z =I0

Q^*_j: +0.I992; +0.4704; +0.II9I; +0.I845; +0.0268;

$J[Q^*_j]^{**}_i$: 0.863; I.0I; 0.I93; 0.045; 0.26; 0.0I9; 0.0I; 0.5I5; 0.89;

φ_{max}: 0.I83; φ_s: 0.792; m_i: 3;

. . . .

. . . .

z =50

Q^*_j: +0.2356; +0.4600; +0.0384; +0.I753; +0.0904;

$J[Q^*_j]^{**}_i$: 0.822; I.0I; 0.27; 0.066; 0.274; 0.052; 0.0I; 0.264; 0.879;

φ_{max}: 0.I22; φ_s: 0.659; m_i: 5;

$J[Q^*_j]^*_{min}$: 0.3556: $J[Q^*_j]^*_{max}$: 0.9307;

Таким образом, в процессе решения задачи наблюдается улучшение качества обучения, на что указывает уменьшение значений φ_{max} и φ_s .

ТЕКСТ ПРОГРАММЫ П5.

```
1 НАЧАЛО ЦЕЛЫЙ I,J,N,-N,M,ITER,MAXIT,-Д,IJK,IJ,K
 ,-L,L,I1,I2;
2 ВЕЩЕСТВЕННЫЙ B5;
3 ЛОГИЧЕСКИЙ *АЛЬФА*10,*АЛЬФА*11,*АЛЬФА*12,*АЛЬФА
 *13;
4 ЦЕЛЫЙ L42,L24;
5 ВЕЩЕСТВЕННЫЙ A10,B10;
6 ВЕЩЕСТВЕННЫЙ A,B,C,‾Q1,‾Q2,*АЛЬФА*,*БЭТА*,*ГАММ
 А*,*ДЕЛЬТА*P,Δ1;*ФИ*S,*ФИ*M;
7 МАССИВ T,X,‾I,‾K[1:100],P,P1[1:250],*PO*[1:9];
8 ЦЕЛЫЙ МАССИВ D[0:10];
9 ЦЕЛЫЙ МАССИВ D1,D2[1:9];
10 ЦЕЛЫЙ УС2;
11 ЦЕЛЫЙ УС3;
12 ЦЕЛЫЙ MIXIT,N1;
13 ВЕЩЕСТВЕННЫЙ *БЭТА*1;
14 ЦЕЛЫЙ МАССИВ ‾P1,‾P2[1:250];
15 ЦЕЛЫЙ МАССИВ ‾P[1:250]; ЦЕЛЫЙ УС,УС1;
16 ‾P[ ]:=0;
17 ‾P1[ ]:=0;
18 ‾P2[ ]:=0;
19 ВВОД(УС,УС1);
20 ВВОД(УС2);
```

```
21  ВВОД(УС3);
22  ВВОД(C,*АЛЬФА*,*БЭТА*,¯Д,MAXIT,M,¯N,X,IJK,IJ,¯L,
    D);
23  ВВОД(MIXIT);
24  ВЫВОД(УС,УС1);
25  ВЫВОД(УС2);
26  ВЫВОД(УС3);
27  ВЫВОД(C,*АЛЬФА*,*БЭТА*,¯Д,MAXIT,¯N,IJK,M,IJ,¯L);
28  ВЫВОД(MIXIT);
29  ¯C¯П0175(1153,X[1],X[M],0,0);
30  ¯C¯П0176(3201,D[0],D[-L],0,0); ВЫВОД( ИСТИНА );
31  A:=⏨6; B:=-⏨6;
32  P1[I]:=0;
33  ДЛЯ I:=1,...,M ЦИКЛ НАЧАЛО ЕСЛИ
    X[I]>B ТО B:=X[I];
34  ЕСЛИ X[I]<A ТО A:=X[I] КОНЕЦ ;
35  ДЛЯ I:=1,...,M ЦИКЛ X[I]:=(X[I]-A)
    /(B-A)+C ;
36  ¯C¯П0176(1153,X[1],X[M],0,0);
37  ВЫВОД( ИСТИНА );
38  N:=0; ¯M:N:=N+1; A:=⏨6; B:=-⏨6;
    P[N]:=0;
39  ЕСЛИ УС2=1 ТО
40  ВВОД(T)
```

```
41   ИНАЧЕ

42   ¯C¯П0175(0016,T[1],T[M],1,[JK+N-1);

43   ДЛЯ I:=1,...,M ЦИКЛ  НАЧАЛО  ЕСЛИ T[I]<A ТО A:=
T[I];

44   ЕСЛИ T[I]>B ТО B:=T[I] КОНЕЦ ;

45   ДЛЯ I:=1,...,M ЦИКЛ T[I]:=(T[I]-A)/(B-A);

46   ДЛЯ I:=1,...,M ЦИКЛ  ДЛЯ I:=1,...,M ЦИКЛ

47   P[N]:=P[N]+(T[I]-T[J])×(X[I]-X[J]); K:=(N-1)×M+1;

48   ¯M¯3¯У(T,1,M,0,4,K,48,0);

49   ВЫВОД(A,B);

50   ЕСЛИ N<¯N ТО  НА ¯M;

51   ВЫВОД( ИСТИНА );

52   N1:=0;

53   *ДЕЛЬТА*P:=0;

54   ДЛЯ N:=1,...,¯N ЦИКЛ *ДЕЛЬТА*P:=
*ДЕЛЬТА*P+ABS(P[N]);

55   P[]:=P[]/*ДЕЛЬТА*P;

56   ITER:=0;

57   ¯I¯T¯E¯R:ITER:=ITER+1; *ДЕЛЬТА*P:=0;

58   *БЭТА*1:=*БЭТА*/(¯N-N1);

59   ¯I[]:=0; N:=0; ¯M1:N:=N+1; K:=(N-1)
×M+1;

60   ¯M¯3¯У(T,1,M,0,4,K,32,0);
```

```
61   ДЛЯ I:=1,...,M ЦИКЛ

62   ¯I[I]:=¯I[I]+( ЕСЛИ P[N]>0 ТО P[N]×
     T[I] ИНАЧЕ (-P[N])×(1-T[I]));

63   ЕСЛИ N<¯N ТО НА ¯M1;

64   ЕСЛИ УС=1 ТО НАЧАЛО ДЛЯ I:=1,...
     ,M ЦИКЛ ¯I[I]:=-I[I]+C; НА ¯Ю КОНЕЦ;

65   A:=₁₀6; B:=-₁₀6;

66   ДЛЯ I:=1,...,M ЦИКЛ НАЧАЛО ЕСЛИ
     ¯I[I]<A ТО A:=¯I[I];

67   ЕСЛИ ¯I[I]>B ТО B:=¯I[I] КОНЕЦ ;

68   ДЛЯ I:=1,...,M ЦИКЛ ¯I[I]:=(¯I[I]
     -A)/(B-A)+C;

69   A10:=A; B10:=B;

70   ¯Ю:

71   ¯K[ ]:=L; B:=-₁₀6; *ФИ*S:=0;

72   ЕСЛИ ¯L=1 ТО НАЧАЛО ДЛЯ I:=1,...,M ЦИКЛ НАЧА
     ЛО A1:=ABS(X[I]-¯I[I]);

73   ЕСЛИ A1>B ТО НАЧАЛО B:=A1; *ФИ*M:=A1; I1:=I КО
     НЕЦ ;

74   ЕСЛИ УС3=1 ТО ¯K[I]:=X[I]/¯I[I];

75   *ФИ*S:=*ФИ*S+A1 КОНЕЦ ;

76   ДЛЯ I:=1,...,M ЦИКЛ ЕСЛИ ABS(¯I[I]-¯I[I1])<₁₀-6
     ТО ¯K[I]:=X[I]/¯I[I1] КОНЕЦ ;

77   ЕСЛИ ¯L>1 ТО НАЧАЛО

78   L:=0; D[0]:=0; ¯M5:A:=₁₀6; B:=-₁₀6; L:=L+1;

79   ДЛЯ I:=D[L-1]+1,...,D[L] ЦИКЛ

80   ЕСЛИ ¯I[I]<A ТО НАЧАЛО A:=¯I[I]; I1:=I КОНЕЦ ;
```

```
81  ДЛЯ I:=D[L]+1,...,D[L+1] ЦИКЛ
82  ЕСЛИ ⁻I[I]>B ТО НАЧАЛО B:=⁻I[I]; I2:=I
    КОНЕЦ;
83  ⁻PO⁻[L]:=A-B;
84  D1[L]:=I1;
85  D2[L]:=I2;
86  ДЛЯ I:=D[L-1]+1,...,D[L] ЦИКЛ
87  НАЧАЛО
88  ЕСЛИ ABS(⁻I[I]-A)<⏨-6 ТО ⁻K[I]:=X[I]/A;
89  ЕСЛИ УС3=1 ТО ⁻K[I]:=X[I]/⁻I[I];
90  КОНЕЦ;
91  ДЛЯ I:=D[L]+1,...,D[L+1] ЦИКЛ
92  НАЧАЛО
93  ЕСЛИ ABS(⁻I[I]-B)<⏨-6 ТО ⁻K[I]:=X[I]/B;
94  ЕСЛИ УС3=1 ТО ⁻K[I]:=X[I]/⁻I[I];
95  КОНЕЦ;
96  ЕСЛИ L<⁻L-1 ТО НА ⁻M5 КОНЕЦ;
97  N:=0; ⁻M2:N:=N+1; K:=(N-1)*M+1;
98  ⁻M⁻3⁻У(T,1,M,0,4,K,32,0);
99  ⁻Q1:=0; ⁻Q2:=0;
100 ДЛЯ I:=1,...,M ЦИКЛ НАЧАЛО ЕСЛИ P[N] ≥ 0
    ТО A:=T[I] ИНАЧЕ A:=(1-T[I]);
```

```
101  ¯Q1:=¯Q1+A×¯K[I]; ¯Q2:=¯Q2+A КОНЕЦ ;
102  P1[N]:=P[N]×¯Q1/¯Q2;  ЕСЛИ N<¯N ТО  НА ¯M2; B:=-10^6 ;
103  ДЛЯ N:=1,...,¯N ЦИКЛ *ДЕЛЬТА*P:=*ДЕЛЬТА*P+ABS(P1[N]);
104  P1[]:=P1[]/*ДЕЛЬТА*P;
105  ДЛЯ N:=1,...,¯N ЦИКЛ  НАЧАЛО A1:=ABS(P1[N]-P[N]);  ЕСЛИ A1>B ТО B:=A1 КОНЕЦ ;
106  B5:=ITER/IJ;  ЕСЛИ ABS(ENTIER(B5)-B5)<10^-6 ТО  НАЧАЛО
107  ВЫВОД(ITER, ИСТИНА );
108  ¯C¯П0176(1153,P[1],P[¯N],0,0); ВЫВОД( ИСТИНА );
109  ¯C¯П0176(1153,¯I[1],¯I[M],0,0); ВЫВОД( ИСТИНА );
110  ЕСЛИ ¯L=1 ТО ВЫВОД
111  (*ФИ*M,*ФИ*S,I1, ИСТИНА ) ИНАЧЕ НАЧАЛО
112  ¯C¯П0176(1153,*PO*[1],*PO*[¯L-1],0,0);
113  ¯C¯П0176(3201,D1[1],D1[¯L-1],0,0);
114  ¯C¯П0176(3201,D2[1],D2[¯L-1],0,0);
115  ВЫВОД( ИСТИНА ) КОНЕЦ ;
116  КОНЕЦ ;
117  ЕСЛИ УС1=1 ТО  НА ¯Ю1 ИНАЧЕ  НА ¯Ю2;
118  ¯Ю1:
119  P[]:=P1[];  ЕСЛИ ITER ≥ MAXIT ТО НАЧАЛО
120  N:=0; ¯M3:N:=N+1;
```

```
ЕСЛИ ABS(P[N])<*БЭТА*1 И ~P2[N]=0
ТО
НАЧАЛО P[N]:=0; N1:=N1+1; ~P2[N]:=
888 КОНЕЦ ;
ЕСЛИ N<~N ТО  НА ~М3 КОНЕЦ ;
НА ~Ю3;
~Ю2:P[]:=P1[]; N:=0; ~М7:N:=N+1;
ЕСЛИ ABS(P[N])<*БЭТА*1 И ~P1[N]=0
ТО ~P1[N]:=ITER ;
ЕСЛИ ABS(P[N])<*БЭТА*1 И ~P1[N]>0
 И ITER > P1[N] + MIXIT
И ~P[N]=0 ТО
НАЧАЛО P[N]:=-*БЭТА*1; ~P[N]:=ITER
КОНЕЦ ;
ЕСЛИ ABS(P[N])<*БЭТА*1 И ~P[N]>0 И
 ITER > ~P[N] + MAXIT
И ~P2[N]=0 ТО
НАЧАЛО P[N]:=0; N1:=N1+1; ~P2[N]:=888 КОНЕЦ ;
ЕСЛИ N<~N ТО  НА ~М7;
~Ю3:
ЕСЛИ В<*АЛЬФА* ИЛИ ITER ≥ ~Д ТО  НА ~Я;  НА ~I~
T~E~R;
~Я:
ВЫВОД(ITER, ИСТИНА );
~С~П0176(1153,P[1],P[~N],0,0); ВЫВОД( ИСТИНА );
~С~П0176(1153,~I[1],~I[M],0,0); ВЫВОД( ИСТИНА )
ЕСЛИ ~L=1 ТО ВЫВОД
```

```
(*ФИ*M,*ФИ*S,[1, ИСТИНА ) ИНАЧЕ  НАЧАЛО

¯C-П0176(1153,*PD*[1],*PD*[¯L-1],0,0);

¯C¯П0176(3201,D1[1],D1[¯L-1],0,0);

¯C¯П0176(3201,D2[1],D2[¯L-1],0,0);

ВЫВОД( ИСТИНА ) КОНЕЦ ;

ВЫВОД(A10,B10, ИСТИНА );

КОНЕЦ *
```

Программа П-6
"Распознавание"

Назначение.

Программа предназначена для вычисления строчечных нагрузок распознаваемых объектов по вычисленным по предыдущей программе информационным весам признаков. В случае наличия объектов экзамена, заданных отдельно от эталонов, эта же программа используется для определения надежности распознавания на заданной исходной информации.

Инструкция к пользованию.

Порядок постановки перфокарт.

I. α - схема или рабочая программа.
2. N - число признаков.
3. a - массив $t_{j\,min}$ } по результатам
4. b - массив $t_{j\,max}$ } обучения
5. m - число распознаваемых объектов.
6. p - массив информационных весов (по результатам обучения).
7. C - регуляризирующее число.
8. δ - условное число, определяющее нормирование (δ=I) или ненормирование ($\delta \neq$ I) полученных строчечных нагрузок.
9. Ma - J^{H}_{max} } по результатам
10. Mi - J^{H}_{min} } обучения (при $\delta = 1$).
II. t - описание объектов распознавания.

На печать выводится

I. m ; 2. c ; 3. δ ; 4. p ; 5. $J^{H}_{i\,пр}$.

Текст программы - П 6.

```
начало целый I ,к,М, "дельта" ,N ,¯N; вещественный_ С ,MA,MI;
массив_I ,T[I:IOO] ,A,в,p,[I:250]; ввод (¯N); ввод (A,в,M,p,
с, "дельта"); вывод (истина); вывод (M,с,"дельта" );¯C¯ПОI76
(II53, p[1] ,p [¯N], 0,0); вывод (истина); если_ "дельта"
=1 то ввод (MA,MI); ¯I[]:=0; N :=0;¯M:N:= N +1; ввод (T); для
I :=1,...,M цикл_ начало T[I]:=(T[I] -A[N])/(в[N] -A[N] );
¯I[I] :=¯I[I] + (если_ p[N] > 0 то p[N] x T[I] иначе (¯P[N] )
x (1-T[I] )) конец ; если_N<¯N то на¯M; если_ "дельта" =I то
начало_ для I :=1,...,M цикл_ ¯I[I]:=(¯I[I] -MI)/(MA-MI)+c
конец иначе началодля для I :=1,...,M цикл_¯I [I]:=¯I[I]+ с
конец; ¯C¯ПОI76 (II53, ¯I[1],¯I[M] ,0,0); вывод (истина);конец *.
```

Ш. КОМПЛЕКС ПРОГРАММ К МЕТОДУ "ЦЕЛЕВАЯ ИТЕРАЦИОННАЯ КЛАССИФИКАЦИЯ" (ЦИКЛ)

Данный метод предназначен для решения задач автоматической классификации и распознавания. Он включает в себя:

а) выбор информативной системы характеристических признаков в соответствии с известными значениями целевого признака объектов-эталонов;

б) принятие решений о значениях целевого признака для объектов-проб на основе информативной системы признаков (ИСП). Метод "Целевая итерационная классификация" ("Цикл") характеризуется следующими особенностями:

- во-первых, ослаблением ограничений на исходные данные, как в информационном плане (допускаются и качественные и количественные признаки и пропуски значений ("-") на некоторых объектах), так и в статистическом (отсутствуют предположения о независимости признаков и законе распределения их значений);
- во-вторых, заменой перебора большого числа сочетаний признаков (при выборе ИСП) последовательными приближениями в соответствующей итерационной процедуре;
- в третьих, целевым учетом, как индивидуальной значимости характеристических признаков, так и их дополнительности в сочетаниях (относительно получения для эталонных объектов характеристических оценок, "близких" или совпадающих с их оценками по целевому признаку), включая оценку информативности каждого признака и минимизацию их начальной системы [1]. Перечисленные особенности метода позволили расширить класс решаемых с помощью ЭВМ геологических задач.

Краткое алгоритмическое описание метода и математические справки к программам

На множестве эталонных объектов $S = \{S_i\}\ i = 1,\ldots,m$, где m - число объектов, заданы значения t_{ij} характеристических признаков x_j, $j = 1,2,\ldots,n$ и значения $t_{i,n+1}$ целевого признака x_{n+1}. Для каждого признака x_j, $j = 1, 2, \ldots, n$, на основе соответствующих ему значений t_{ij}, $i = 1, 2, \ldots, m$, фиксируется способ вычисления набора оценок $d(S, x_j)$, сравнимых с аналогичными оценками $d(S, x_{n+1})$, вычисленными для целевого признака x_{n+1}. Конкретно, эти оценки представляют собой либо нормированные расстояния для каждой пары объектов, либо нормированные значения признака для отдельных объектов [2, 3]. Способы вычисления таких оценок могут быть различны. В данном методе они находятся в соответствии с разными видами шкал значений признаков (наименований, порядка, интервалов и отношений) [1,4].

При вычислении этих оценок в программе "Цикл-1" каждой паре исходных объектов S_{i_1}, S_{i_2}, где $i_1 = 1, 2, \ldots, m-1$, $i_2 = i_1 + 1, \ldots, m$, ставится в соответствие строка $S'_k = (d_{k1}, d_{k2}, \ldots, d_{k,n+1})$, в которой элементы $d_{k,j}$ представляют собой нормированные расстояния между S_{i_1} и S_{i_2} по признаку x_j, $j = 1, 2, \ldots, n+1$, а номер k – есть номер пары (S_{i_1}, S_{i_2}) в последовательности пар $(S_1, S_2), \ldots, (S_1, S_m), (S_2, S_3), \ldots, (S_2, S_m), \ldots, (S_{m-1}, S_m)$, т.е. $1 \leq k \leq C_m^2$, $0 \leq d_{kj} \leq 1$.

В программе "Цикл-2" каждая строка $S'_k = (d_{k1}, \ldots, d_{k,n+1})$ соответствует одному исходному объекту S_k, где $k = 1, 2, \ldots, m$, т.е. k – номер объекта S_k, а элементы d_{kj} есть нормированные значения компонент t_{kj} объекта S_k, $j = 1, 2, \ldots, n + 1$. Получаемые при этом оценки $\{d_{k,j}\}$ неинвариантны относительно преобразований исходных значений признаков в виде $t_{ij} \to C_j - t_{ij}$ т.е. относительно направленности их кода. В таком случае, по каждому x_j находятся линейные связи a_j, $\bar{a}_j$ ряда оценок d_{kj}, $\bar{d}_{kj}$, $k = 1, 2, \ldots, m$, с рядом оценок $d_{k,n+1}$, $k = 1, 2, \ldots, m$:

$$a_j = \sum_{k=1}^{m} (1 - |d_{kj} - d_{k,n+1}|); \quad \bar{a}_j = \sum_{k=1}^{m} (1 - |\bar{d}_{kj} - d_{k,n+1}|),$$

где $\bar{d}_{kj} = 1 - d_{kj}$

Если $\bar{a}_j > a_j$, то код признака x_j меняется таким образом: $t_{ij} \to C_j - t_{ij}$, а если $\bar{a}_j \leq a_j$, то код признака x_j сохраняется неизменным. Это соответствует замене значений оценки $d_{kj} = 1, 2, \ldots m$ оценкой $\bar{d}_{kj}$ или сохранению исходных значений d_{kj}

I. Алгоритмическая основа метода ЦИКЛ

Пусть задана таблица оценок d_{kj}, $k = 1,2,\dots,R$, $R \in \{m, C^2_{m}\}$, $j = 1,2,\dots, n+1$, и заданы начальные значения $P_j^{(0)}$, $j = 1,2,\dots, n$, информационных весов характеристических признаков x_j при условии их нормировки $\sum_{j=1}^{n} P_j^{(0)} = 1$. Тогда алгоритмическая основа метода (или классифицирующий оператор K) строится из совокупности следующих шести операций:

1) Вычисляются характеристические оценки $\rho_k^{(0)}$ для строк по всей системе признаков:

$$\rho_k^{(0)} = \sum_{j=1}^{n} \left(P_j^{(0)} \cdot d_{kj} \right), \qquad k = 1,2,\dots, R. \tag{1}$$

2) Проводится сравнение целевой $d_{k,n+1}$ и характеристической $\rho_k^{(0)}$ оценок строки S_k' по величине отношения

$$D_k^{(0)} = \frac{d_{k,n+1}}{\rho_k^{(0)}}, \qquad k = 1,2,\dots, R \tag{2}$$

3) Аналогичные (1) и (2) операции проводятся и для таблицы оценок $\{\bar{d}_{k,j}\} = \{1 - d_{kj}\}$:

$$\bar{\rho}_k^{(0)} = \sum_{j=1}^{n} \left(P_j^{(0)} \cdot \bar{d}_{kj} \right); \quad \bar{D}_k^{(0)} = \frac{\bar{d}_{k,n+1}}{\bar{\rho}_k^{(0)}}, \quad k=1,2,\dots R \tag{3}$$

4) Оценивается информативность всей системы признаков по критерию:

$$J^{(0)} = 2R \Big/ \sum_{k=1}^{R} \left(D_k^{(0)*} + \bar{D}_k^{(0)*} \right), \tag{4}$$

где

$$D_k^{(0)*} = \begin{cases} D_k^{(0)} & \text{при } D_k^{(0)} \geq 1 \\ 1 & \text{при } D_k^{(0)} < 1 \end{cases} \qquad \bar{D}_k^{(0)*} = \begin{cases} \bar{D}_k^{(0)} & \text{при } \bar{D}_k^{(0)} \geq 1 \\ 1 & \text{при } \bar{D}_k^{(0)} < 1 \end{cases}$$

5) Производится поощрение индивидуальных вкладов $(P_j^{(0)} \cdot d_{kj}^*)$, $(P_j^{(0)} \cdot \bar{d}_{kj}^*)$ признаков x_j, $j = 1,\dots, n$, в оценки $\rho_k^{(0)}, \bar{\rho}_k^{(0)}$ с помощью коэффициентов $D_k^{(0)*}, \bar{D}_k^{(0)*}$ и находятся прямые $z_j^{(1)}$, обратные $\bar{z}_j^{(1)}$ и совмещенные $G_j^{(1)}$ целевые нагрузки столбцов по формулам:

$$r_{kj}^{(1)} = P_j^{(0)} \cdot d_{kj}^{*} \cdot D_k^{(0)*}, \qquad \bar{r}_{kj}^{(1)} = P_j^{(0)} \cdot \bar{d}_{kj}^{*} \cdot \bar{D}_k^{(0)*},$$

$$r_j^{(1)} = \sum_{k=1}^{R} r_{kj}^{(1)}; \quad \bar{r}_j^{(1)} = \sum_{k=1}^{R} \bar{r}_{kj}^{(1)}; \quad \sigma_j^{(1)} = r_j^{(1)} + \bar{r}_j^{(1)}, \tag{5}$$

где $d_{kj}^{*} = \begin{cases} d_{kj} & \text{при } d_{kj} \le d_{k,n+1}, \\ d_{k,n+1} & \text{при } d_{kj} > d_{k,n+1}, \end{cases} \quad \bar{d}_{kj}^{*} = \begin{cases} \bar{d}_{kj} & \text{при } \bar{d}_{kj} \le \bar{d}_{k,n+1} \\ \bar{d}_{k,n+1} & \text{при } \bar{d}_{kj} > \bar{d}_{k,n+1} \end{cases}$

6) Получается первое приближение $P_j^{(1)}$, $j = 1,2,\ldots,n$, целевых информационных весов с помощью нормировки совмещенных целевых нагрузок $\sigma_j^{(1)}$:

$$P_j^{(1)} = \frac{\sigma_j^{(1)}}{\sum_{j=1}^{n} \sigma_j^{(1)}} \tag{6}$$

По аналогии с этой формулой, где $\sum_{j=1}^{n} P_j^{(1)} = 1$, в программах также предусмотрено получение информационных весов $P_j^{(1)\prime}$ и $P_j^{(1)\prime\prime}$ отдельно по прямым $r_j^{(1)}$ и обратным $\bar{r}_j^{(1)}$ целевым нагрузкам столбцов.

В перечисленных шести операциях описан процесс получения набора $P^{(1)} = (P_1^{(1)}, P_2^{(1)}, \ldots, P_n^{(1)})$ первого приближения весов n признаков, исходя из любого начального набора $P^{(0)} = (P_1^{(0)}, P_2^{(0)}, \ldots, P_n^{(0)})$ n чисел $0 \le P_j^{(0)} \le 1$, лишь бы $\sum_{j=1}^{n} P_j^{(0)} = 1$. Обозначим этот процесс буквой $\mathcal{K}$.

Тогда имеем $\mathcal{K}(P^{(0)}) = P^{(1)}$, что символически обозначает тот факт, что в результате процесса $\mathcal{K}$ вектор $P^{(0)}$ переходит в $P^{(1)}$, причем из построения видно, что $0 \le P_j^{(1)} \le 1$, $\sum_{j=1}^{n} P_j^{(1)} = 1$, а это значит, что и к вектору $P^{(1)}$ можно применить преобразование $\mathcal{K}$. Получим

$$\mathcal{K}(P^{(1)}) = \mathcal{K}(\mathcal{K}(P^{(0)})) = \mathcal{K}^2(P^{(0)})$$

Обозначая $\mathcal{K}(P^{(1)})$ символом $P^{(2)}$, $\mathcal{K}(P^{(2)})$ – через $P^{(3)}$ и т.д., получаем итерационный процесс вида

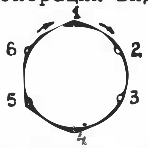
$$P^{(\tau+1)} = \mathcal{K}(P^{(\tau)}), \quad \tau = 0, 1, 2 \ldots$$

Запись этого же вида применима и к другим величинам, вычисляемым по формулам (I)-(6), например для формулы (4) имеем

$$J^{(\tau+1)} = \mathcal{K}\,(J^{(\tau)}), \quad \tau = 0, 1, 2, \ldots \qquad (7)$$

Определение.

Последовательность операций вида (I)-(6) образующих цикл



называется классифицирующим оператором $\mathcal{K}$.

Нужная точность сходимости описанного итерационного процесса получается за конечное число шагов ℓ, т.е. в результате ℓ шагов получим, что

$$\max_{1 \le j \le n} |P_j^{(\ell)} - P_j^{(\ell+1)}| < \varepsilon. \qquad (8)$$

Это было подтверждено как на множестве численных машинных экспериментов, так и при решении задач, содержащих реальную геологическую информацию (по программам "Цикл-I" и "Цикл-2").

В конце итерационного процесса (при $\tau = \ell$) мы получаем:

с одной стороны, минимальную систему признаков, состоящую из столбцов с максимальным индивидуальным целевым вкладом в характеристические оценки строк (например, если есть характеристический столбец, совпадающий с целевым, то процесс выбора ИСП всегда сходится на нем);

с другой стороны, информационные веса $P_j^{(\ell)}$ распределяются таким образом, что значения характеристических оценок $\rho_k^{(\ell)}, \bar{\rho}_k^{(\ell)}$ максимально приближаются к целевым оценкам $d_{k,n+1}$ в соответствии с критерием (4), где $J^{(\tau)} \to \max\{J^{(1)}, J^{(2)}, \ldots, J^{(\ell)}\}$ при $\tau \to \ell$. При $J^{(\tau)} = I$ система признаков с информационными весами $P^{(\tau)}$ считается полностью информативной, а стремление к выполнению первого условия в итерационном процессе способствует однозначности решения задачи выбора ИСП.

2. Правила распознавания

Принятие решений о значении t^n_{n+1} целевого признака x_{n+1} для любого допустимого к распознаванию [3] объекта-пробы $S^n = (t_1^n, \ldots, t_n^n)$ проводится в строгом соответствии с описанными выше способами получения характеристических оценок $\{d_{kj}, \bar{d}_{kj}\}$, $\{\rho_k^{(I)}, \bar{\rho}_k^{(I)}\}$, ($j = I,\ldots, n$; $k = I,\ldots, R$) для объектов-эталонов. Пусть заданы: система признаков с весами $P_1^*, \ldots, P_n^*$, такими, что $J^* \geq \varkappa$ ($\varkappa$ - заданный порог достаточной информативности этой системы признаков относительно эталонов); соответствующие оценки $\{d_{kj}^n, \bar{d}_{kj}^n\}, \{\rho_k^{n*}, \bar{\rho}_k^{n*}\}$ объекта - пробы ($j = I,\ldots, n$; $k = I,\ldots, R^n$, где $R^n \in \{1, m\}$.

В программе "Цикл-I" ($R^n = m$) оценки пробы обозначают нормированные (по каждому признаку x_j относительно эталонов) различия и сходства для каждой пары ($k = I,\ldots, m$) эталон-проба, где ρ_k^{n*} и $\bar{\rho}_k^{n*}$ есть меры (коэффициенты) различия и сходства по всей системе признаков. В связи с предусмотренными вари - антами получения ИСП (5), (6) отдельно для прямой, обратной и совмещенной таблиц оценок эталонов, на мерах $\rho_k^{n*}, \bar{\rho}_k^{n*}$ основаны три решающих правила:

а) а н а л о г и и; б) и с к л ю ч е н и я;
в) п р о п о р ц и о н а л ь н о с т и . Описание этих правил дано в работе [I].

В программе "Цикл-2" ($R^n = I$) оценки $\{d_j^n, \bar{d}_j^n\}, \{\rho^{n*}, \bar{\rho}^{n*}\}$ обозначают нормированные (относительно эталонов по каждому признаку x_j , $j = I,\ldots, n$) значения признаков самой пробы S^n , где ρ^{n*} и $\bar{\rho}^{n*}$ есть прямая и обратная характеристические нагрузки пробы по системе признаков с весами $P_1^*,\ldots, P_n^*$. В данном случае эти нагрузки выступают в качестве непосредственного критерия оценки нормированного значения d_{n+1}^n целевого признака для пробы S^n . То есть принятие решений здесь упрощается и для оценки значения целевого признака используются следующие три правила:

а) оценка сверху (позитивная) $\rho^{n*} \geq d_{n+1}^n - \varepsilon$,

б) оценка снизу (негативная) $\bar{\rho}^{n*} \geq \bar{d}_{n+1}^n - \bar{\varepsilon}$,

в) пропорциональная оценка (совмещенная) $\rho^{n*} = d_{n+1}^n \pm \varepsilon$.

Литература

I. А.А.Бишаев. Метод "Целевая итерационная классификация" ("Цикл").-В сб.: Логико-математические исследования в геологии. Новосибирск, 1976, стр. 70-91.

2. А.А.Бишаев. Итерационный способ нахождения информативной системы признаков для целевой классификация объектов. Ш Всесоюзная конференция по проблемам теоретической кибернетики (Тезисы докладов). Новосибирск, 1974, стр.185-187.

3. А.Н.Дмитриев и др. Организация и обработка геологической информации с помощью ЭВМ на основе построения тупиковых тестов. - В кн.: Логико-информационные решения геологических задач. М., "Наука", 1975, с.83-128.

4. П.Супес, Дж.Зинес. Основы теории изменений. - В кн.: Психологические измерения, М., "Мир", 1969, стр.I-II0.

К о м п л е к с   п р о г р а м м (ПI, П2)

Программа "Цикл-I" (ПI)
(Язык Альфа, ЭВМ М-222, БЭСМ-6)

Назначение.

Программа "Цикл-I" предназначена для выбора информативной системы признаков (ИСП) и распознавания по ИСП значений целевого признака объектов-проб н а   о с н о в е   вычисления о т н о с и т е л ь н ы х   о ц е н о к   объектов (мер р а з л и ч и я   и   с х о д с т в а   между ними). Для получения лучшего решения в связи с различием конкретных задач, в программе предусмотрена возможность ее работы в целом комплексе различных режимов с учетом всех особенностей соответствующего метода [I,2].

Инструкция к пользованию.

Дадим краткие пояснения к использованию и способам задания тех или иных числовых параметров, в связи с порядком постановки перфокарт (п/к) для счета.

I. Альфа-схема , кΣ .

Далее идут числа и числовые массивы.

2. m, кΣ – число всех объектов, включая пробы при решении задачи распознавания.

3. n, кΣ – число характеристических признаков.

4. $n1$, кΣ – число вариантов исходных информационных весов признаков.

5. ℓ, кΣ – число объектов-проб. При работе программы в режиме выбора ИСП $\ell = 0$.

6. Таблица $t[0:m, 1:n+1]$ – числовой массив исходных данных (значений признаков для объектов). Числа перфорируются по строкам. Нулевая строка (t_{0j}), $j=1,\dots,n+1$ обозначает вид столбца, где $j=n+1$ номер целевого признака, $t_{0j} \in \{I,2,3\}$ [I]. При t_{ij} = "–" ($i=1,\dots,m$; $j=1,\dots,n+1$) пробивается число 10^{10}. В режиме распознавания пробы ставятся в начале массива (строки с номерами $i=1,\dots,\ell$ и со значениями $t_{i,n+1}=10^{10}$).

7. $PH[1:n1, 1:n]$ – числовой массив значений исходных информационных весов характеристических признаков ($n1$ вариантов). Перфорируется по строкам начиная с (PH_{1j}) $j=1,\dots,n$ и т.д. Эти веса могут быть некормированными.

8. eps, кΣ – точность сходимости (значение "порога" для установления факта повторения наборов информационных весов при-

знаков на двух соседних итерациях). Например, $eps \le \frac{1}{100n}$

9. вых, $k\Sigma$ - режим работы программы: а) вых =1 - на прямой таблице оценок (различия), б) вых =2 - на обратной таблице оценок (сходства), в) вых =3 - на совмещенной таблице оценок (различия-сходства).

10. α, $k\Sigma$ -значение "порога" зануления оценок объектов, например, $\alpha \le 5 \cdot 10^{-2}$.

11. B, $k\Sigma$ -значение "порога" для зануления нормированного веса признаков (в итерациях), например, $B \le \frac{1}{7n'}$, где n' - число признаков с ненулевым весом.

12. $h1$, $k\Sigma$ - степень скорости изменения весов признаков (по итерациям), например, $0 \le h1 \le Const$, где $Const > 1$. При $h1 = 0$, веса признаков не изменяются.

13. C, $k\Sigma$ - в о п е р в ы х (при $C > 0$), условие работы программы в режиме выбора ИСП и ограничитель на число итераций в этом режиме; в о в т о р ы х (при $C = 0$), условие работы программы в режиме распознавания.

14. h_2, $k\Sigma$ - степень влияния принципа целевой дополнительности признаков на оценку их информационных весов (по итерациям), например, $0 \le h_2 \le Const$, где $Const > 1$. При $h2 = 0$, производится только индивидуальная оценка информативности каждого характеристического признака относительно целевого признака.

15. $ti\,[0:m]$ - массив значений целевого признака на объектах. С помощью этого массива можно задавать целевой признак независимо от значений , пробитых для столбца $(n+1)$ в таблице исходных данных (см.пункт 6). Значение $ti\,[0] \in \{1, 2, 3\}$ и обозначает вид целевого столбца.

16. $\gamma 1$, $k\Sigma$ - коэффициент влияния прямых целевых оценок объектов (различий) на выбор ИСП, где $0 \le \gamma 1 \le 1$. Например, $\gamma 1 = 0$ обозначает исключение прямой таблицы, что равнозначно режимному условию вых =2.

17. $\gamma 2$, $k\Sigma$ - коэффициент влияния обратных оценок объектов на выбор ИСП, где $0 \le \gamma 2 \le 1$. Например, $\gamma 2 = 0$ равнозначно вых=1.

Ограничения.

Объем исходных данных ($m \times n$) не должен превышать: для М-222- ($3 \cdot 10^3$), для БЭСМ-6 - ($2 \cdot 10^4$).

Время счета линейно зависит от C_m^2 и n. Например, для таблицы $m=30$, $n=50$ в различных режимах работы программы на ЭВМ М-222 время счета составляло (40-60) минут при пороге сходимости ($eps = 10^{-4}$) и числе итераций $\tau = (60-70)$.

Порядок выводов на печать в режиме выбора ИСП ($c > 0$).

1. $v \in \{1, \ldots, n1\}$ - номер варианта исходных весов признаков
2. τ' - номер итерации, выдаваемый на печать
3. $J^{(\tau')}$ - оценка информативности системы признаков
4. $(P_1^{(\tau')}, \ldots, P_n^{(\tau')})$- массив информационных весов признаков
5. $j' \in \{1, \ldots, n\}$ - номер признака с максимальной по абсолютной величине разностью его весов на текущей и предыдущей итерациях.
6. $a(j')$ - величина максимального изменения веса с учетом знака

Вывод на печать производится на итерациях с номерами $\tau' \in \left\{\frac{k \cdot (k+1)}{2}\right\}$ $(k = 0, 1, 2, \ldots)$, включая последнюю итерацию с номером $\tau' < c$ (при достижении сходимости) или с номером $\tau' = c$ (при ограничении по числу итераций).

Порядок выводов на печать в режиме распознавания ($c = 0$).

Если вых =1, то на печать выдаются:

1. i' - номер эталонного объекта с которого начинается класс эталонных объектов (за исключением первого класса эталонов, для которого этот номер не выдается, так как во всех случаях он равен $\ell + 1$, где ℓ - число проб).

При этом отметим, что классом считается любой набор таких объектов (начиная с единичного), для которых оценки по целевому признаку одинаковые с точностью до α.

2. $i1$ - номер пробы.
3. $\rho_{cp}(i_1', i)$- среднее значение прямых оценок (мер различия) пробы относительно эталонов одного класса. Для данной пробы эти три параметра выдаются на печать последовательно по всем классам эталонов. Затем аналогичная выдача производится для следующей пробы и т.д. до $i1 = \ell$.

Если вых =2, то на печать выдаются:

1. i - номер эталона
2. $i1$ - номер пробы
3. $\bar{\rho}(i, i1)$ - значение соответствующей обратной оценки (меры сходства).

Эти три параметра выдаются последовательно для всех номеров эталонов одного класса, а вслед за перечисленными параметрами, выдаваемыми для всего класса, следуют:

4. i'' – номер эталонного объекта, с которым проба имеет максимальную обратную оценку $\bar{\rho}$ в данном классе.

5. i_1 – номер пробы

6. $\bar{\rho}(i'', i1)$ – величина соответствующей оценки.

Последовательность таких выводов (I–6) на печать повторяется для каждого класса до $i = m$. Затем для всех последующих проб (до $i_1 = \ell$) производятся аналогичные выдачи.

Если $вых = 3$, то производится полная выдача сначала, как при $вых = I$, а затем – как при $вых = 2$.

Окончательное принятие решений проводится по правилам, изложенным в [I].

При всех режимах работы программы по окончанию счета на печать выдаются также входные параметры в следующей последовательности: eps, $вых$, α, β, $h1$, C, $h2$, m, n, $n1$, ℓ, PH.

Контрольный пример

Задан числовой массив исходных данных $\{t_{ij}\}$, $i = 0, \ldots, m$, $j = 1, \ldots, n+1$ в табличном виде:

$i \backslash j$	I	2	3	4
0	I	I	I	2
I	I	I	0	I0
2	I	0	I	4
3	0	I	I	2

$m(i) = 3$ – число объектов
где $n(j) = 3$ – число признаков
$n+1(j) = 4$ – целевой признак
$i = 0$ – строка (виды признаков)

Для других входных числовых параметров заданы следующие значения: $n1 = I$; $\ell = 0$; $PH = (0{,}33;\ 0{,}33;\ 0{,}33)$; $eps = 0{,}0033$; $вых = 3$; $\alpha = 0{,}05$; $\beta = 0{,}04$; $h1 = I$; $C = 30$; $h2 = I$; $t1 = (2,\ I0,\ 4,\ 2)$; $\gamma 1 = I$; $\gamma 2 = I$.

Процесс решения задачи выбора г.о.п., относительно полученных предварительно прямых (a_{ij}) и обратных ($\bar{a}_{kj}$) оценок и основные результаты счета по программе "Цикл-I" представлены на рисунке 1.

k	j	1 d	1 $\bar{d}$	2 d	2 $\bar{d}$	3 d	3 $\bar{d}$	4 d	4 $\bar{d}$	$\rho^{(0)}$	$\bar{\rho}^{(0)}$	$\rho^{(1)}$	$\bar{\rho}^{(1)}$	$\rho^{(3)}$	$\bar{\rho}^{(3)}$	$\rho^{(6)}$	$\bar{\rho}^{(6)}$	…	$\rho^{(14)}$	$\bar{\rho}^{(14)}$
1	d	0		1		1		0.75		0.66		0.81		0.81		0.79		…	0.75	
	$\bar{d}$		1		0		0		0.25		0.33		0.19		0.19		0.21	…		0.25
2	d	1		0		1		1		0.66		0.75		0.85		0.91		…	1	
	$\bar{d}$		0		1		0		0		0.33		0.25		0.15		0.09	…		0
3	d	1		1		0		0.25		0.66		0.44		0.34		0.30		…	0.25	
	$\bar{d}$		0		0		1		0.75		0.33		0.56		0.66		0.70	…		0.75

(τ')	P_1	P_2	P_3	J
(0)	0.33	0.33	0.33	$J^{(0)}=$ 0.74
(1)	0.19	0.25	0.56	$J^{(1)}=$ 0.86
(3)	0.19	0.15	0.66	$J^{(3)}=$ 0.89
(6)	0.21	0.09	0.70	$J^{(6)}=$ 0.93
…	…	…	…	…
(14)	0.25	0	0.75	$J^{(14)}$ 1

Рис.I. Основные результаты счета в процессе решения задачи выбора ИСП по программам "Цикл-I" и "Цикл-2" на совмещенной таблице оценок.

ТЕКСТ ПРОГРАММЫ

```
НАЧАЛО  ЦЕЛЫЙ M,N,N1;
ВВОД(M,N,N1);
НАЧАЛО  МАССИВ T[0:M,1:N+1],P1,P[1:N],ADG[1:N+1],НОРМ[1:N+1];
МАССИВ T1[0:M],PH[1:N1,1:N];
МАССИВ PP[0:N];
ЦЕЛЫЙ I,J,K,I1,J1,K1,TI,TJ,IT,ВЫХ,IAVA,JAVA,L,БЧ,C;  ЦЕЛЫЙ H2;
ВЕЩ A,A1,EPS,*АЛЬФА*,*БЭТА*,AI,AJ,AMAX,AMIN,DG1,DG2,*ПИ*,ИНФ1,ИНФ2;  ВЕЩ S1,H1,*ГАММА*1,
*ГАММА* 2; ВВОД(L,T,PH,EPS,ВЫХ,*АЛЬФА*,*БЭТА*,H1, C);
ВВОД(H2,T1,*ГАММА*1,*ГАММА*2);
ДЛЯ J:=0,...,M ЦИКЛ  НАЧАЛО T[J,N+1]:=T1[J] КОНЕЦ ;
ДЛЯ IAVA:=1,...,N1 ЦИКЛ  НАЧАЛО T1[]:=0;
ППО: ДЛЯ J:=1,...,N ЦИКЛ P[J]:=PH[IAVA,J];
ДЛЯ I:=1,...,N+1 ЦИКЛ  НАЧАЛО НОРМ[I]:=ю12 КОНЕЦ ; *ПИ*:=-1; IT:=0; Ц-И-К-Л:AMAX:=0;
ИНФ1:=0; ИНФ2:=0;
P1[]:=0; S1:=0;
T1[0]:=T1[1]:=0;
DG1:=DG2:=0; A:=0;  ДЛЯ J:=1,...,N ЦИКЛ A:=A+P[J];
```

```
JAVA:=0;  ДЛЯ J:=1,...,N ЦИКЛ  НАЧАЛО P[J]:=P[J]/A;
 ЕСЛИ IT=0 ТО PP[J]:=P[J];
 ЕСЛИ P[J]≠0 ТО JAVA:=JAVA+1 КОНЕЦ ;
 ЕСЛИ C=0 ТО K1:=L ИНАЧЕ K1:=M-1;  ДЛЯ I1:=1,...,K1 ЦИКЛ  НАЧАЛО
 ЕСЛИ C=0 ТО ЬЧ:=K1 ИНАЧЕ ЬЧ:=I1;  ДЛЯ I:=ЬЧ+1,...,M ЦИКЛ  НАЧАЛО
A:=A1:=0;
 ЕСЛИ IT=-1 ТО  НА П5;
 ЕСЛИ T1[I1]=1ORT1[I]=1 ТО  НА XX;
П5:
 ДЛЯ J:=1,...,N+1 ЦИКЛ  НАЧАЛО
J1:=T[0,J];  ЕСЛИ J1≠1 ТО  НА R2;  TI:=T[I1,J];  TJ:=T[I,J];
 ЕСЛИ TI>₁₀5ORTJ>₁₀5 ТО  НАЧАЛО ADG[J]:=*ПИ*;
 НА RK1 КОНЕЦ ;
 ЕСЛИ TI=TJ ТО ADG[J]:=0 ИНАЧЕ ADG[J]:=1;  НА RK1;  R2: ЕСЛИ НОРМ[J]<₁₀5 ТО  НА RK2;  AMAX:=-₁₀
10;  AMIN:=-AMAX;
 ДЛЯ K:=1,...,M ЦИКЛ  НАЧАЛО AI:=T[K,J];  ЕСЛИ T1[K]=1ORAI>₁₀5 ТО  НА RK3;  ЕСЛИ K<L ТО  НА R
K3;  ЕСЛИ AMAX<AI ТО AMAX:=AI;
 ЕСЛИ AMIN>AI ТО AMIN:=AI;
RK3: КОНЕЦ ;
```

```
 ЕСЛИ AMAX=AMIN ТО НОРМ[J]:=0 ИНАЧЕ
НОРМ[J]:=1/(AMAX-AMIN);
 ЕСЛИ J1=3 ТО НОРМ[J]:=AMAX×НОРМ[J];
RK2:AI:=T[1,J]; AJ:=T[1,J]; ЕСЛИ AI>₁₀5ORAJ>₁₀5 ТО НАЧАЛО ADG[J]:=*ПИ*;
 НА RK1 КОНЕЦ ;
ADG[J]:=ABS(AI-AJ)×НОРМ[J];
 ЕСЛИ J1=3ANDADG[J]≠0 ТО НАЧАЛО
 ЕСЛИ AI<AJ ТО ADG[J]:=ADG[J]/AJ
 ИНАЧЕ ADG[J]:=ADG[J]/AI КОНЕЦ ;
RK1:AMAX:=0       КОНЕЦ ;
 ЕСЛИ ВЫХ=2 ТО НА XX2;
AJ:=ADG[N+1];
PP[0]:=0;
 ЕСЛИ C=0 ТО НА Ж1;
 ЕСЛИ AJ<*АЛЬФА* ТО НА XX2;
Ж1:
 ДЛЯ J:=1,...,N ЦИКЛ НАЧАЛО AI:=ADG[J]; ЕСЛИ AI=-1 ТО НА УК1; A:=A+AI×P[J];
PP[0]:=PP[0]+AI×PP[J]; УК1: КОНЕЦ ;
 ЕСЛИ C=0 ТО НАЧАЛО
```

```
ЕСЛИ ABS(AJ-T1[0])>*АЛЬФА* ТО ПП4: НАЧАЛО
ЕСЛИ T1[0]=0 ТО S1:=S1-1;
ЕСЛИ S1>0 ТО НАЧАЛО ЕСЛИ S1≠0 ТО DG1:=DG1/S1;
ВЫВОД(I1,DG1,ЛОЖЬ,I) КОНЕЦ ; S1:=DG1:=0;
ЕСЛИ AJ=T1[0] ТО НАЧАЛО ВЫВОД(ЛОЖЬ); T1[0]:=0; НА XX КОНЕЦ ; T1[0]:=AJ КОНЕЦ ;
S1:=S1+1; DG1:=DG1+A;

ЕСЛИ I=M ТО НА ПП4; НА XX КОНЕЦ ; AJ:=AJ×*ГАММА*1;
ЕСЛИ 2×JAVA×A<AJ ТО НАЧАЛО
T1[I]:=1; ВЫВОД(IT,I1,I,ЛОЖЬ); ЕСЛИ IT<0 ТО IT:=-1; НА XX КОНЕЦ ;
DG1:=(AJ/A)↑H2;
PP[0]:=(AJ/PP[0])↑H2; S1:=S1+PP[0]×A;
ЕСЛИ DG1<1 ТО ИНФ2:=ИНФ2+1 ИНАЧЕ ИНФ2:=ИНФ2+DG1;
AMAX:=AMAX+ABS(PP[0]-AMIN);
T1[1]:=T1[1]+AJ; ИНФ1:=ИНФ1+DG1;
T1[0]:=T1[0]+1;
ДЛЯ J:=1,...,N ЦИКЛ НАЧАЛО AI:=ADG[J]; ЕСЛИ AI<0 ТО НА УК2;
P1[J]:=P1[J]+AI×DG1; УК2: К
                              ОНЕЦ ; XX2: ЕСЛИ ВЫХ=1 ТО НА XX; AJ:=1-ADG[N+1];
PP[0]:=0;
```

```
ЕСЛИ C=0 ТО НАЧАЛО AJ:=1-AJ; НА Ж2 КОНЕЦ ;
ЕСЛИ AJ<*АЛЬФА* ТО НА XX; Ж2:
ДЛЯ J:=1,...,N ЦИКЛ НАЧАЛО AI:=1-ABS(ADG[J]);
A1:=A1+AI*P[J];
PP[0]:=PP[0]+AI*PP[J] КОНЕЦ ;
ЕСЛИ C≠0 ТО НАЧАЛО
ЕСЛИ ABS(AJ-T1[0])>*АЛЬФА* ТО НАЧАЛО
ВЫВОД(IT,I1,AMAX,ЛОЖЬ); AMAX:=0; T1[0]:=AJ КОНЕЦ ;
ЕСЛИ A1>AMAX ТО НАЧАЛО AMAX:=A1; IT:=I КОНЕЦ ;
ВЫВОД(I,I1,A1); ЕСЛИ I=M ТО НАЧАЛО ВЫВОД(IT,I1,AMAX,ЛОЖЬ,ЛОЖЬ);
T1[0]:=0 КОНЕЦ ;
НА XX КОНЕЦ ; AJ:=AJ**ГАММА*2;
ЕСЛИ 2*JAVA*A1<AJ ТО НАЧАЛО T1[I]:=1; J:=-I;
ВЫВОД(IT,I1,J,ЛОЖЬ);
ЕСЛИ IT<0 ТО IT:=-1; НА XX КОНЕЦ ;
DG2:=(AJ/A1)↑H2; PP[0]:=(AJ/PP[0])↑H2; S1:=S1+PP[0]*A1; ЕСЛИ DG2<1 ТО ИНФ2:=ИНФ2+1 ИНАЧЕ И
НФ2:=ИНФ2+DG2;
AMAX:=AMAX+ABS(PP[0]-AMIN);
T1[1]:=T1[1]+AJ; ИНФ1:=ИНФ1+DG2;
T1[0]:=T1[0]+1;
```

```
ЕСЛИ DG2=0 ТО НА XX;
ДЛЯ J:=1,...,N ЦИКЛ НАЧАЛО AI:=1-ADG[J];
ЕСЛИ AI=2ORAI<*АЛЬФА* ТО НА УК4;
P1[J]:=P1[J]+AI×DG2; УК4: КОНЕЦ ; XX: КОНЕЦ КОНЕЦ ;
ЕСЛИ C=0 ТО НАЧАЛО ЕСЛИ ВЫХ=3 ТО НАЧАЛО
ВЫХ:=2; НА ЦИКЛ КОНЕЦ ; НА КОП КОНЕЦ ;
ЕСЛИ IT=-1 ТО НА ППО;
A:=0; ДЛЯ I:=1,...,N ЦИКЛ НАЧАЛО
ЕСЛИ *БЭТА*=0 ТО P1[I]:=P1[I]↑H1 ИНАЧЕ НАЧАЛО
ЕСЛИ *АЛЬФА*=*БЭТА* ТО P1[I]:=(P1[I]×P[I])↑H1 ИНАЧЕ
P1[I]:=P1[I]↑H1×P[I] КОНЕЦ ;
A:=A+P1[I] КОНЕЦ ; ДЛЯ I:=1,...,N ЦИКЛ P1[I]:=P1[I]/A;
A1:=0; ДЛЯ I:=1,...,N ЦИКЛ НАЧАЛО ЕСЛИ P1[I]≠0ANDP1[I]<*БЭТА* ТО P1[I]:=0; ЕСЛИ ABS(A1)
<ABS(P1[I]-P[I]) ТО НАЧАЛО
A1:=P1[I]-P[I]; БЧ:=I КОНЕЦ КОНЕЦ ; ЕСЛИ IT=0 ТО AMIN:=ИНФ1/T1[0];
A:=ИНФ1; ИНФ1:=S1/(AMIN×T1[1]);
AMIN:=A/T1[0];
A:=1/H2;
ИНФ2:=-1[0]/ИНФ2;
```

```
ИНФ2:=ИНФ2↑A;
AMAX:=(AMAX/T1[0])↑A;  ЕСЛИ ABS(A1)>EPSANDIT<C ТО  НАЧАЛО
 ЕСЛИ IT=0 ТО K:=L;  ЕСЛИ L=-2 ТО  НА ПП2;
 ЕСЛИ IT=K*(K+1)/2 ТО ПП2: НАЧАЛО K:=K+1;
ВЫВОД(IAVA,IT,ИНФ1,ИНФ2,P,ЬЧ,A1,ЛОЖЬ) КОНЕЦ ;  ДЛЯ I:=1,...,N ЦИКЛ  НАЧАЛО PP[I]:=P[I]; P[I
]:=PI[I] КОНЕЦ ;
IT:=IT+1;  НА -Ц-И-К-П КОНЕЦ ; ВЫВОД(T1[0],AMAX);
ВЫВОД(IAVA,IT,     ,ИНФ2,P,ЬЧ,A1,ЛОЖЬ);
КОП: КОНЕЦ ;
ВЫВОД(EPS,ВЫХ,*АЛЬФА*,*БЭТА*,H1,
C);
ВЫВОД(H2,M,N,N1,L,PH);
 КОНЕЦ  КОНЕЦ
```

Программа "Цикл-2" (П2)
(Язык Альфа, ЭВМ М-222, БЭСМ- 6)

Назначение.

Программа "Цикл-2" предназначена для выбора информативной системы признаков (ИСП) и распознавания по ИСП значений целевого признака объектов-проб на основе вычисления абсолютных оценок отдельных объектов (нормированных значений признаков на объектах). В связи с различием практических задач в программе предусмотрена возможность ее использования в различных режимах. При выборе режима работы следует учитывать особенности конкретной задачи и самого метода [I,2] .

Инструкция к пользованию

Порядок постановки перфокарт и смысл входных числовых характеристик те же самые, что и в программе "Цикл-I". Исключение составляет параметр ℓ , который используется здесь лишь при выборе ИСП в качестве параметра, контролирующего вывод результатов тех или других итераций на печать. ℓ принимает значения из множества $\{-2, 0, C \geq I\}$,где C - верхняя граница числа итераций.

Если ℓ =-2, то выдача результатов счета производится на всех итерациях; при ℓ =0 - на итерациях с номерами $\tau' \in \left\{ \frac{k(k+1)}{2} \right\}$ (к=0,I,2;...); при ℓ = C - только на последней итерации. К шестнадцати входным характеристикам программы "Цикл-I" добавляются:

I7. $X1, K\Sigma$ - коэффициент дополнительной нормировки характеристических оценок ($\rho_k^{(\tau)}$) на каждой итерации $k = 1, \ldots, m$.

I8. $X2, K\Sigma$ аналогичный коэффициент для ($\bar{\rho}_k$) $k = 1, \ldots, m$.

Объекты-пробы входят в общее число (m) объектов исходной таблицы. Они могут задаваться и в режиме выбора ИСП, тогда их оценка (т.е. распознавание) проводится параллельно с оценкой эталонов на ка дой итерации.

Ограничения - аналогичны ограничениям программы "Цикл-I".

Время счета линейно зависит от m и n . Например, для исходной таблицы m =30, n =50 оно составляло (7-I2) минут в различных режимах работы программы на ЭВМ М-222 при пороге сходимости ($eps = 10^{-4}$) и числе итераций τ =(80-90). Отметим, что время работы программы "Цикл-2" (также, как и "Цикл-I") в режиме распознавания примерно равно времени ее работы на одной итерации режима выбора ИСП при тех же значениях m и n.

Порядок выводов на печать в режиме выбора ИСП.
Перед началом итерационного процесса на печать выводятся:
I) j - номер признака;
2) $t(j)\in\{-1,+1\}$ - значение знака линейной связи j-го признака с целевым;
3) $P_{(j)}^{(0)}$ - значение начального информационного веса признака. В этом порядке данные параметры выдаются для всех $j=I,\ldots,n$
Затем выдаются начальные оценки объектов:
I) K - номер объекта; 2) $d_{n+1}(K)$ - прямая оценка объекта по целевому признаку; 3) $\rho_K^{(0)}$ - прямая оценка объекта по начальной системе признаков; 4) $\bar{\rho}_K^{(0)}$ - обратная оценка этого объекта. В этой последовательности перечисленные параметры выдаются для всех $K=I,\ldots,m$.

В итерационном процессе для итерации с номером (τ') на печать выводятся: I) $v\in\{1,\ldots,n1\}$, 2) τ', 3) j', 4) $a(j')$, 5) $J^{(\tau')}$, 6) $P_1^{(\tau')},\ldots,P_n^{(\tau')}$. Этот вывод на печать повторяется для всех номеров (τ'), в соответствии со значением входного параметра ℓ. Смысловое содержание данных характеристик аналогично выдаваемым параметрам (I-6) по программе "Цикл-I".

По окончанию итерационного процесса на печать выдается та же последовательность характеристик, что и перед его началом, но для конечных значений $(P_1^{(*)},\ldots,P_n^{(*)})$ информационных весов признаков и для соответствующих оценок объектов по этим весам.

В режиме распознавания на печать выводятся параметры, аналогичные выводимым в режиме выбора ИСП до начала итерационного процесса.

Контрольный пример.

Таблицу оценок, полученную при решении задачи выбора ИСП по программе"Цикл-I", можно рассматривать как исходную таблицу для решения той же самой задачи по программе "Цикл-2". Поэтому процесс решения и основные результаты счета по программе "Цикл-2" (для таблицы оценок, представленной на рис.I) будут такие же, как по программе "Цикл-I".

Текст программы.

В вышеизложенном алгоритмическом описании метода ЦИКЛ показано, что программа "Цикл-2" имеет общую алгоритмическую основу с программой "Цикл-I". Кроме того, текст программы "Цикл-2" является модификацией текста программы "Цикл-I" и поэтому здесь не приводится.

IУ. КОМПЛЕКС ПРОГРАММ АППРОКСИМАЦИИ КРИВЫХ ДЛЯ ПРОГНОЗИРОВАНИЯ СТРУКТУРНЫХ ПЛАНОВ

Программа аппроксимации кривых, получения параметров, радиусов кривизны, центров мгновенных радиусов кривизны, среднего радиуса кривизны

(Язык Алгол-60, ЭВМ- М-220)

НАЗНАЧЕНИЕ

При изучении структур различного порядка в нефтегазоносных бассейнах исследователь часто сталкивается с их криволинейной формой. Всё разнообразие кривых, которые встречаются при изучении пликативных дислокаций, можно аппроксимировать и свести к семейству конических сечений [3,4]. Аппроксимированная кривая, отражающая степень искривления слоистой толщи,характеризуется радиусом кривизны в каждой точке, средним радиусом кривизны и т.д. Эти параметры прямо отражают степень деформированности слоистой толщи и позволяют выделить районы с деформациями различного класса [3].

ОПИСАНИЕ АЛГОРИТМА

Анализируемые кривые аппроксимируются полиномом заданной степени [2]. Если от интерполяционного многочлена не требуется, чтобы он обязательно совпадал с аппроксимируемой функцией в заданных точках, то предполагается, что зависимость y от x имеет вид многочлена:

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m$$

с неизвестными коэффициентами.

Говоря, что функция задана, мы подразумеваем при этом, что известны её значения лишь в некоторых точках. Требуется найти по результатам наблюдений наиболее вероятные значения коэффициентов $a_0, a_1, \ldots, a_m$. Нам важен случай, когда число наблюдений n много больше степени многочлена. Получающаяся в этом случае задача может быть решена методом наименьших квадратов. При $n \leq m+1$ нет смысла применять метод наименьших квадратов. В этом случае система уравнений для определения коэффициентов a_i совместна, и сумма квадратов будет равна нулю при любых решениях системы.

Необходимо найти коэффициенты $a_0, a_1, \ldots, a_m$, дающие минимум функции

$$\Phi = \sum_{i=1}^{n} [y_i - (a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_m x_i^m)]^2.$$

Продифференцируем сумму квадратов отклонений по $a_0, a_1, \ldots, a_m$ и производные приравняем к нулю.

После ряда элементарных преобразований, учитывая известные правила дифференцирования, получим систему уравнений

$$n a_0 + a_1 \sum_{i=1}^{n} x_i + a_2 \sum_{i=1}^{n} x_i^2 + \ldots + a_m \sum_{i=1}^{n} x_i^m = \sum_{i=1}^{n} y_i$$

$$a_0 \sum_{i=1}^{n} x_i + a_1 \sum_{i=1}^{n} x_i^2 + a_2 \sum_{i=1}^{n} x_i^3 + \ldots + a_m \sum_{i=1}^{n} x_i^{m+1} = \sum_{i=1}^{n} x_i y_i$$

- -

$$a_0 \sum_{i=1}^{n} x_i^m + a_1 \sum_{i=1}^{n} x_i^{m+1} + a_2 \sum_{i=1}^{n} x_i^{m+2} + \ldots + a_m \sum_{i=1}^{n} x^{2m} = \sum_{i=1}^{n} x_i^m y_i \qquad (1)$$

В стандартной процедуре (PO70.4) отыскания коэффициентов полинома, к которой обращается программа, есть прием, позволяющий в некоторых случаях упростить систему. Это возможно, если значения x являются точными и даны с постоянным шагом.

Введем вместо x новый аргумент u. Если число наблюдений нечетное $n = 2k + 1$, то полагаем $u = \frac{x - x'_{k+1}}{h}$ или $x - x'_{k+1} = uh$. Когда x последовательно принимает значения $x_1, x_2, \ldots, x_{2k+1}$, то u будет принимать целочисленные значения $-k, -k+1, \ldots, -1, 0, 1, \ldots, k-1, k$. Если искать y как функцию от u, т.е. коэффициенты полинома

$$y = a_0 + a_1 u + \ldots + a_m u^m, \tag{2}$$

то система (I) для определения $a_0, a_1, \ldots, a_m$ значительно упростится, так как благодаря выбору значений суммы нечетных степеней также будут равны нулю:

$$\sum_{i=1}^{2k+1} u_i = \sum_{i=1}^{2k+1} u_i^3 = \ldots = 0 \tag{3}$$

Для случая четного числа наблюдений $n = 2k$ переменная u вводится равенством

$$u = \frac{2(x - x_k)}{h} - 1 \quad \text{или} \quad x = \frac{(u+1)h}{2} + x_k$$

Тогда при изменении индекса у x от I до $2k$ величина u последовательно будет равна

$$-2k+1, \ldots, -3, -1, 1, 3, \ldots, 2k-1$$

и снова сумма нечетных степеней u будет равна нулю.

Вот как выглядит система (I) в простейших случаях многочленов первой и второй степени.

Пусть $y = a_0 + a_1 u$, тогда, учитывая равенство (3), получим (здесь и в дальнейшем суммирование ведется в пределах от I до i и индексы суммирования для краткости опускаются).

$$n a_0 = \sum y_i \; ; \qquad a_1 \sum u_i^2 = \sum u_i y_i$$

откуда $a_0 = \frac{1}{n} \sum y_i$, $\qquad a_1 = \frac{\sum u_i y_i}{\sum u_i}$

Для случая многочлена второй степени $y = a_0 + a_1 u + a_2 u^2$ система (I) запишется в виде:

$$n a_0 + a_2 \sum u_i^2 = \sum y_i$$
$$a_1 \sum u_i^2 = \sum u_i y_i$$
$$a_0 \sum u_i^2 + a_2 \sum u_i^4 = \sum u_i^2 y_i$$

Решая полученную систему, найдем

$$a_0 = \frac{\sum y_i \sum u_i^4 - \sum u_i^2 y_i \sum u_i^2}{n \sum u_i^4 - (\sum u_i^2)^2}$$

$$a_1 = \frac{\sum u_i y_i}{\sum u_i^2} \; ; \qquad a_2 = \frac{n \sum u_i^2 y_i - \sum y_i \sum u_i^2}{n \sum u_i^4 - (\sum u_i^2)^2}$$

После того, как найдены путем обращения к стандартной процедуре (PO 704) коэффициенты полинома заданной степени находим радиус кривизны:

$$R = \frac{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{3}{2}}}{\frac{d^2y}{dx^2}}$$

Координаты центра мгновенных радиусов кривизны для каждой пары чисел:

$$x_c = x - \frac{\frac{dy}{dx}\left[1 + \left(\frac{dy}{dx}\right)^2\right]}{\frac{d^2y}{dx^2}} \; ;$$

$$y_c = y + \frac{1 + \left(\frac{dy}{dx}\right)^2}{\frac{d^2y}{dx^2}}$$

Средний радиус кривизны:

$$R^* = \frac{\sum_1^n R_n}{n}$$

Находим из найденной формулы:

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_m \cdot x^m.$$

ИНСТРУКЦИЯ К ПОЛЬЗОВАНИЮ ПРОГРАММОЙ

Исходные данные должны быть заготовлены в следующем виде:

I) карта числа исходных данных (M) ;
2) карта степени аппроксимирующего полинома (NN) ;
3) начальное значение абсциссы (x_0) (абсцисса крайней левой точки кривой);
4) шаг по оси абсцисс (Δx) ;
5) массив значений ординат (B) последовательный, начиная с крайней левой точки;

Для каждой кривой комплектуется пакет исходных данных и, в соответствии с последовательностью решения кривых, комплектуется общий пакет исходных данных. Ввод – бесконечный.

Для того, чтобы не происходило остановок решения из-за переполнения разрядной сетки, вычисление радиусов кривизны производится с предварительной проверкой порядка результатов вычислений.

Выдача производится в следующем порядке:

I) число точек кривой (m);
2) степень аппроксимирующего полинома (N);
3) начальное значение x_0 ;
4) шаг по x – Δx ;
5) абсциссы кривой x ;
6) исходные ординаты кривой [y] ;
7) коэффициенты полинома $y = B_0 + B_1 \cdot x + \ldots + B_N \cdot x^N$ $\div (a_0 ; a_2)$
8) ординаты аппроксимирующей кривой [M] ;
9) таблица первых производных [y'] ;
I0) таблица вторых производных [y''] ;
II) абсциссы центров кривизны [x_c] ;
I2) ординаты центров кривизны [y_c] ;
I3) радиусы кривизны [R] ;
I4) средний радиус кривизны R^* .

КОНТРОЛЬНЫЙ ПРИМЕР

$(m = 13 ;\ n = 2)$

m	I	2	3	4	5	6	7	8	9	I0	II	I2	I3
N	2												
x_o	−2,5												
Δ_x	+0,25												
X	−I,5	−I,25	−I,0	−0,75	−0,5	−0,25	0	+0,25	+0,5	+0,75	+I,0	+I,25	+I,5
y	−0,024	−0,022	−0,0I6	−0,0I2	−0,08	−0,06	0	−0,006	−0,008	−0,0I2	−0,0I6	−0,020	−0,022

$a_o = 0{,}008$ $a_2 = 0{,}005$

m	I	2	3	4	5	6	7	8	9	I0	II	I2	I3
M	−0,023	−0,0I9	−0,0I5	−0,0I2	−0,0I0	−0,0087	−0,008	−0,0085	−0,009	−0,0I0	−0,0I2	−0,0I5	−0,0I9
y'	+0,0I9	+0,0I6	+0,0I3	+0,0I0	+0,007	+0,006	+0,00I	−0,002	−0,004	−0,007	−0,0I0	−0,0I3	−0,0I6
y''	−0,0II	−0,0II	−0,0II	−0,0II	−0,0II	−0,0II	−0,0II	−0,0II	−0,0II	−0,0II	−0,0II	−0,0II	−0,0II
x_c	+I2,[illegible]	+I2,28	+I2,20	+I2,2[illegible]	+I2,24	+I2,24	+I2,24	+I2,24	+I2,24	+I2,24	+I2,23	+I2,22	+I2,2I
R	−85,77	−85,I5	−85,I4	−85,73	−85,78	−85,72	−85,72	−85,72	−85,72	−85,73	−85,73	−85,74	−85,75
y_c	−85,78	−85,77	−85,75	−85,74	−85,73	−85,[illegible]	−85,72	−85,73	−85,73	−85,74	−85,75	−85,76	−85,77
R^*	85,76												

Л и т е р а т у р а

I. Айнберг В.Д. Основы программирования на алгоритмическом языке АЛГОЛ-60. М., "Машиностроение", 1973, 150 с.

2. Бут Э.Д. Численные методы. М., Физматгиз, 1959, 239 с.

3. Прокопенко А.И. Количественное выражение дизъюнктивных дислокаций седиментационных бассейнов в связи с их нефтегазоносностью (на примере Ферганской межгорной впадины). -В кн.: Методология и методика геологических и геофизических исследований в Сибири. Новосибирск, 1975, 138 с.

4. Прокопенко А.И. Анализ цикличности строения моласс при прогнозировании структурных планов глубоких горизонтов локальных поднятий. - В кн.: Цикличность осадконакопления и закономерности размещения горючих полезных ископаемых. Новосибирск, 1975, с.234-236.

ТЕКСТ ПРОГРАММЫ

```
BEGIN INTEGER N,NN,M,KK ;
ВВОД: P0042(M) ;
      P0165(1,'=10ЧИСЛОТОЧЕККРИВОЙ  ') ;
      P1041(M) ;
      P0042(NN) ;
      P0165(1,'=10СТЕПЕНЬАППРОКСИМИРУЮЩЕГОПОЛИНОМА  ') ;
      P1041(NN) ;
      N:=M-1 ; KK:=2×(N+1)+7×NN+7 ;
      BEGIN INTEGER I,J ; REAL H,L,DT,D,D1,D2,D3,DH,X,ROP ;
        ARRAY RR[0:KK],XOHO[1:2],A,B,B1,P,P1,P2,XC,YC,
          R[0:N],B2,B3[0:NN] ;
        P0042(X) ;
        P0042(H) ;
        P0165(1,'=10НАЧАЛЬНОЕЗНАЧЕНИЕX  ') ;
        P1041(X) ;
        P0165(1,'=10ШАГПОX  ') ;
        P1041(H) ;
        P0042(B) ;
        FOR I:=0 STEP 1 UNTIL N DO A[I]:=X+I×H ;
        P0165(1,'=10АБСЦИССЫКРИВОЙ  ') ;
        P1041(A) ;
        P0165(1,'=10ИСХОДНЫЕОРДИНАТЫКРИВОЙ  ') ;
        P1041(B) ;
        XOHO[1]:=A[0] ; XOHO[2]:=H ; DT:=18 ; L:=LN(10) ;
        P0704(XOHO[1],RR[0],NN,B[0],M) ;
        FOR I:=0 STEP 1 UNTIL NN DO
```

```
BEGIN B2[I]:=RR[I] ;
END ;
P0165(1,
 '=10КОЭФФИЦИЕНТЫПОЛИНОМАy=B0+B1×X+...+BN×X↑N  ') ;
P1041(B2) ;
FOR I:=0 STEP 1 UNTIL N DO
  BEGIN P[0]:=B2[0] ;
    FOR J:=1 STEP 1 UNTIL NN DO
      P[J]:=P[J-1]+B2[J]×A[I]↑J ;
    B1[I]:=P[NN] ;
  END ;
P0165(1,'=10ОРДИНАТЫАППРОКСИМИРУЮЩЕЙКРИВОЙ  ') ;
P1041(B1) ;
P[1]:=B2[1] ;
FOR J:=0 STEP 1 UNTIL N DO
  BEGIN FOR I:=2 STEP 1 UNTIL NN DO
      P[I]:=P[I-1]+I×B2[I]×A[J]↑(I-1) ;
    P1[J]:=P[NN] ;
  END ;
P0165(1,'=10ТАБЛИЦАПЕРВЫХПРОИЗВОДНЫХ  ') ;
P1041(P1) ;
IF NN=2 THEN
  BEGIN FOR I:=0 STEP 1 UNTIL N DO P2[I]:=B2[2]×2 ;
  END ELSE
  BEGIN P[2]:=B2[2]×2 ;
    FOR J:=0 STEP 1 UNTIL N DO
```

```
      BEGIN FOR I:=3 STEP 1 UNTIL NN DO
          P[I]:=P[I-1]+B2[I]×I×(I-1)×A[J]↑(I-2) ;
        P2[J]:=P[NN] ;
      END ;
  END ;
P0165(1,'=10ТАБЛИЦАВТОРЫХПРОИЗВОДНЫХ   ') ;
P1041(P2) ;
FOR I:=0 STEP 1 UNTIL N DO P[I]:=ABS(P1[I]) ;
FOR I:=0 STEP 1 UNTIL N-1 DO
  BEGIN P[I+1]:=P[I] ;
  END ;
D:=P[N] ; D1:=LN(D)/L ;
FOR I:=0 STEP 1 UNTIL N DO
  BEGIN XC[I]:=A[I]-P1[I]×(1+P1[I]↑2)/P2[I] ;
    YC[I]:=B[I]+(1+P1[I]↑2)/P2[I] ;
    R[I]:=SQRT(1/D↑2+(P1[I]/D)↑2) ;
    D2:=LN(ABS(R[I]))/L ; D3:=LN(ABS(P2[I]))/L ;
    IF (D1+D2)×3-D3<DT THEN
      R[I]:=((D×R[I])↑3)/P2[I] ELSE
      BEGIN R[I]:=₁₀18 ;
        P1041(I) ;
      END ;
  END ;
P0165(1,'=10АБСЦИССЫЦЕНТРОВКРИВИЗНЫ   ') ;
P1041(XC) ;
P0165(1,'=10ОРДИНАТЫЦЕНТРОВКРИВИЗНЫ   ') ;
```

```
    P1041(VC) ;
    P0165(1,'=10РАДИУСЫКРИВИЗНЫ  ') ;
    P1041(R) ;
    P[0]:=R[0] ;
    FOR I:=1 STEP 1 UNTIL N DO P[I]:=P[I-1]+R[I] ;
    RCP:=P[N]/(N+1) ;
    P0165(1,'=10СРЕДНИЙРАДИУСКРИВИЗНЫ  ') ;
    P1041(RCP) ;
  END ;
  GO TO ВВОД ;
END
```

ПРОГРАММА АППРОКСИМАЦИИ КРИВЫХ СПЛАЙН-ФУНКЦИЕЙ

Программа составлена для транслятора ТА-IM, ЭВМ типа М-220, М-222 на языке АЛГОЛ-60

НАЗНАЧЕНИЕ

Анализ пликативных дислокаций, приуроченных к регионам, характеризующимся различной геологической историей развития. показывает наличие широкого диапазона искривленных структурных форм [3]. В связи с этим встает вопрос выбора из аппарата теории приближений (который по праву считается одним из основных при математическом моделировании геологической информации) такой функции, которая позволит с высокой точностью аппроксимировать все разнообразие кривых, встречающихся в реальных геологических средах [3,2,I].

В последнее время широкое развитие получила новая область численного анализа - теория сплайн функций. Одним из важных принципов, заложенных в понятие сплайна является то, что при приближениии сплайнами добиваются уменьшения погрешности за счет увеличения числа точек склейки в то время как степень склеиваемых многочленов можно брать не очень большой. Кроме этого, сплайны обладают хорошими свойствами сходимости, просто реализуются на ЭВМ [2].

С учетом вышеизложенного была составлена программа аппроксимации кривых сплайн-функцией и вычисления радиусов кривизны, кривизны, центров мгновенных радиусов кривизны. Эти параметры позволяют определению положения свода складки на глубине, прогнозированию структурных планов на больших глубинах и т.д.

КРАТКОЕ ОПИСАНИЕ АЛГОРИТМА

Кривые, аппроксимируемые сплайном, рассматриваются в прямоугольной системе координат (x , y), которая выбирается произвольно. Со структурных карт снимаются координаты структурных отметок (x_i , y_i) ($i = 0,I,\ldots,m$), получаемых пересе -

чением проекции вертикальной плоскости с изолиниями.

Характерной особенностью метода является быстрая сходимость сплайн-функции к аппроксимируемой кривой при увеличении числа точек. Поэтому, для более хорошего приближения воспользуемся дополнительными отметками, которые получаем путем аппроксимации каждых трех соседних точек

$$(x_i, y_i), \quad (x_{i+1}, y_{i+1}), \quad (x_{i+2}, y_{i+2}) \qquad (i = 0, 1, \ldots, m-2)$$

окружность:

$$(x - \bar{x}_i)^2 + (y - \bar{y}_i)^2 = R_i^2 \quad ;$$

где R_i - радиус окружности, $(\bar{x}_i, \bar{y}_i)$ - центр окружности. Эти параметры находим, решая систему уравнений:

$$\begin{cases} 2\left[(x_i - x_{i+1})(y_i - y_{i+2}) - (x_i - x_{i+2})(y_i - y_{i+1})\right]\bar{x}_i = \\ = (x_i^2 - x_{i+1}^2 + y_i^2 - y_{i+1}^2)(y_i - y_{i+2}) - (x_i^2 - x_{i+2}^2 + y_i^2 - y_{i+2}^2)(y_i - y_{i+1}). \\ 2\left[(y_i - y_{i+1})(x_i - x_{i+2}) - (y_i - y_{i+2})(x_i - x_{i+1})\right]\bar{y}_i = \\ = (x_i^2 - x_{i+1}^2 + y_i^2 - y_{i+1}^2)(x_i - x_{i+2}) - (x_i^2 - x_{i+2}^2 + y_i^2 - y_{i+2}^2)(x_i - x_{i+1}) \end{cases}$$

Зададим величину h - интервал, на который отстоят друг от друга дополнительные отметки. Их число и определяется целой частью от деления разности $x_m - x_o$ на h. Определим абсциссы дополнительных отметок

$$x_j = x_o + jh \qquad (j = 0, 1, \ldots, n),$$

а ординаты находим из уравнения окружности, взятого для соответствующего интервала.

Итак, имеем отрезок (x_o, x_n) разделенный точками x_j $(j = 1, 2, \ldots, n-1)$ на $n-1$ равных отрезков. Известны соответствующие ординаты: ищется функция $S(x)$ (сплайн), непрерывная на (x_o, x_n) вместе со своими первой и второй производными, совпадающими с кубическим полиномом на каждом отрезке

$$[x_{j-1}, x_j] \qquad (j = 1, 2, \ldots, n)$$

и удовлетворяющая условиям:

$$S(x_j)=y_j \qquad (j=0,1,\dots,n).$$

В программе сплайн определяется решением следующей системы уравнений:

$$\begin{bmatrix} 2 & \lambda_0 & 0 & \dots & 0 & 0 & 0 \\ \mu_1 & 2 & \lambda_1 & \dots & 0 & 0 & 0 \\ 0 & \mu_2 & 2 & \dots & 0 & 0 & 0 \\ - & - & - & \dots & - & - & - \\ 0 & 0 & 0 & \dots & \mu_{n-1} & 2 & \lambda_{n-1} \\ 0 & 0 & 0 & \dots & 0 & \mu_n & 2 \end{bmatrix} \cdot \begin{bmatrix} M \\ M \\ M \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d \\ d \\ d \\ \cdot \\ d_{n-1} \\ d_n \end{bmatrix},$$

где

$$M_j = S''(x_j); \qquad (j=0,1,\dots,h)$$

$$d_j = 3\,\frac{y_{j+1}+y_{j-1}-2y_j}{h^2}; \qquad \lambda_j=\frac{1}{2}; \quad \mu_j=\frac{1}{2};$$

$$j=1,2,\dots,n-1,$$

а d_0, d_n, λ_0, μ_n определяются краевыми условиями

$$2M_0+\lambda_0 M_1 = d_0, \qquad \mu_n M_{n-1}+2M_n = d_n$$

Коэффициенты полинома на отрезке $[x_j, x_{j+1}]$ $(j=0,1,\dots,n-1)$ определяются из системы уравнений:

$$S''(x_j) = M_j$$

$$S''(x_{j+1}) = M_{j+1}$$

$$S(x_j) = y_j$$

$$S(x_{j+1}) = y_{j+1}$$

Имея аналитическое выражение кривой на каждом отрезке, рассчитываем радиус кривизны R_j и координаты центра мгновенного радиуса кривизны x_c^j и y_c^j для каждой точки $x_j\,(j=1,2,\dots,n-1)$ по формулам

$$R_j = \frac{[1 + (S'(x_j))^2]^{3/2}}{M_j};$$

$$x_c^j = x_j - \frac{S'(x_j)\,[1 + (S'(x_j))^2]}{M_j};$$

$$y_c^j = y + \frac{1 + (S(x_j))^2}{M_j};$$

ИНСТРУКЦИЯ К ПОЛЬЗОВАНИЮ

Данные должны быть представлены в следующем виде:

I. Число точек экспериментальной кривой m ;
2. Массив наблюдаемых значений $x_j^{в}$;
3. Массив наблюдаемых значений $y_j^{в}$;
4. Интервал, на котором отстоят друг от друга диагональные отметки - h ;
5. Величины, определяющие краевые условия - k_a , $k_в$;
6. g - номер профиля.

Последовательность ввода исходных данных следующая:

I. g ; 2. m ; 3. h ; 4. $x_в$; 5. $y_в$; 6. R_a ; 7. $R_в$.

Выдача на печать производится в следующем порядке:

I. m - число точек кривой;
2. $x^{в}$ - массив входных данных;
3. $y^{в}$ - массив входных данных;
4. RS - средний радиус кривизны;
5. KS - средняя кривизна;
6. XcS - средняя абсцисса центра кривизны;
7. ycS - средняя ордината центра кривизны;
8. xc - массив абсцисс центра кривизны;
9. yc - массив ординат центра кривизны;
I0. K - массив кривизны;
II. R - массив радиусов кривизны;
I2. S - массив параметров сплайна отрезка $[x_0, x_n]$.

КОНТРОЛЬНЫЙ ПРИМЕР

Р е з у л ь т а т ы   с ч е т а

m	I	2	3	4	...	I6	I7	I8	I9	...	3I	32	33	34
$x^{в}$	6,24	5,98	5,7	5,4	...	2,8	3,3	3,75	4,I7	...	4,94	5,29	5,63	5,95
$y^{в}$	3,4	3,35	3,3	3,2	...	2,95	3,0	3,05	3,I	...	3,2	3,25	3,3	3,35

RS = 40,2I8 KS = 0,0248 XcS =96,606 YcS =I443,94

m	I	2	3	4	...	I6	I7	I8	I9	...	3I	32	33	34
x_c	0,32	I,2I	7,46	3,46	...	0,I6	0,I7	0,I9	0,036	...	6,57	3,28	0,I4	0,63
y_c	37,72	43,65	I6,96	8,25	...	23,I4	22,I7	25,55	I6,65	...	9I,79	64,52	37,I4	42,95
K	0,028	0,024	0,049	0,I9	...	0,049	0,05I	0,043	0,072	...	0,0II	0,0I6	0,029	0,024
R	34,94	40,94	20,29	5,23	...	20,35	I9,37	22,76	I3,86	...	89,34	6I,87	34,28	40,I4
S	0,05I	0,95	5,68	I4,29	...	3,5	0,0024	0,022	0,I9	...	5,06	3,73	0,0I9	0,22

Л и т е р а т у р а

I. Айнберг Б.Д. Основы программирования на алгоритмическом языке АЛГОЛ-60. М., "Машиностроение", I973, I50 с.

2. Дж.Альберг, Э.Нильсон, Дж.Уолш. Теория сплайнов и ее приложения. М.,"Мир",I972,3I6с.

3. Прокопенко А.И. Об эффективности геолого-геофизических методов при выявлении антиклинальных структур и подготовке их к глубокому бурению.- В кн.: Методология и методика геологических и геофизических исследований в Сибири. Новосибирск,I975,с.I38.

ТЕКСТ ПРОГРАММЫ

```
BEGIN INTEGER T ;   P0042(T);
BEGIN INTEGER M,L ; REAL H ; INTEGER Q ; FOR L:=1 STEP 1 UNTIL
T DO  BEGIN P0042(Q) ;
        P0042(M,H) ;
        BEGIN REAL ARRAY XB,YB[0:M],XR,YR,RR[0:M-1] ;
            REAL KA,KB,KC ; REAL KD,KE,KF ; INTEGER N,I,J ;
          P0042(XB,YB) ;
          P0165(1,
            '-G  ,-  ,-  ,-  ,-  ,-
          
            HOMEPPA3PE3A  ') ;
          P1041(Q) ;
          P0165(1,'ПЕЧАТЬВХОДНЫХДАННЫХ:XB,YB,H') ;
          P1041(XB) ;
          P1041(YB) ;
          P1041(H) ;
          N:=ENTIER((XB[M]-XB[0])/H) ;
          BEGIN REAL ARRAY X,Y[0:N] ;
            FOR I:=0 STEP 1 UNTIL M-2 DO
BEGIN XX:KA:=(XB[I]-XB[I+1])*(YB[I]-YB[I+2])-
          (XB[I]-XB[I+2])*(YB[I]-YB[I+1]) ;
        IF KA=0 THEN
          BEGIN P0165(1,'ЗАМЕНАXB[I]') ;
            P1041(I+2,XB[I+2]) ;
            XB[I+2]:=XB[I+2]+0.02 ;
            GO TO XX ;
          END ;
        KB:=XB[I]↑2+YB[I]↑2-XB[I+1]↑2-YB[I+1]↑2 ;
```

```
            KC:=XB[I]↑2+YB[I]↑2-XB[I+2]↑2-YB[I+2]↑2 ;
            XR[I]:=KB×(YB[I]-YB[I+2])-KC×(YB[I]-YB[I+1]) ;
            YR[I]:=KB×(XB[I]-XB[I+2])-KC×(XB[I]-XB[I+1]) ;
            XR[I]:=XR[I]/(2×KA) ; YR[I]:=-YR[I]/(2×KA) ;
            RR[I]:=(XR[I]-XB[I])↑2+(YR[I]-YB[I])↑2 ;
            RR[I]:=SQRT(RR[I]) ;
        END ;
XR[M-1]:=XR[M-2] ; YR[M-1]:=YR[M-2] ; RR[M-1]:=RR[M-2] ;
KA:=XB[0]-H ; KB:=0 ;
IF XB[0]+N×H=XB[M] THEN
        BEGIN KC:=N-1 ; X[N]:=XB[M] ; Y[N]:=YB[M] ;
        END ;
KC:=N ;
FOR I:=0 STEP 1 UNTIL KC DO
        BEGIN KA:=KA+H ; MM:
          IF KA>XB[KB+1] THEN
            BEGIN KB:=KB+1 ;
              GO TO MM
            END ;
          X[I]:=KA ; KD:=SQRT(RR[KB]↑2-(XR[KB]-KA)↑2) ;
          KE:=KD+YR[KB] ; KF:=YR[KB]-KD ;
          IF ABS(YB[KB]-KE)>ABS(YB[KB]-KF) THEN Y[I]:=KF ELSE
            Y[I]:=KE ;
        END ;
P0165(1,'ПЕЧАТЬХИУ') ;
P1041(X,Y)
```

```
BEGIN REAL ARRAY S[0:N-1,0:3],D,M[0:N] ; REAL LA,MU,A ;
      A:=X[0] ;
      P0042(KA,KB) ;
      D[0]:=KA ; D[N]:=KB ; LA:=0.5 ; MU:=0.5 ;
      FOR I:=1 STEP 1 UNTIL N-1 DO
        D[I]:=3×(Y[I+1]+Y[I-1]-2×Y[I])/(H↑2) ;
      BEGIN REAL ARRAY PP,QQ,UU[1:N+1] ;
        PP[1]:=2 ; QQ[1]:=-LA/2 ; UU[1]:=D[0]/2 ;
        FOR I:=2 STEP 1 UNTIL N DO
          BEGIN PP[I]:=QQ[I-1]/2+2 ; QQ[I]:=-1/(2×PP[I]) ;
            UU[I]:=(D[I-1]-UU[I-1]/2)/PP[I] ;
          END ;
        PP[N+1]:=MU×QQ[N]+2 ;
        UU[N+1]:=(D[N]-MU×UU[N])/PP[N+1] ; M[N]:=UU[N+1] ;
        FOR I:=N-1 STEP -1 UNTIL 0 DO
          M[I]:=QQ[I+1]×M[I+1]+UU[I+1] ;
      END ;
      FOR J:=0 STEP 1 UNTIL N-1 DO
        BEGIN S[J,0]:=(M[J+1]-M[J])/(6×H) ;
          S[J,1]:=M[J]/2-3×S[J,0]×(A+J×H) ; KA:=A+J×H ;
          KB:=Y[J]-S[J,1]×KA↑2-S[J,0]×KA↑3 ; KA:=KA+H ;
          KC:=Y[J+1]-S[J,1]×KA↑2-S[J,0]×KA↑3 ;
          S[J,2]:=(KC-KB)/H ; S[J,3]:=KC-S[J,2]×KA ;
        END ;
      BEGIN REAL ARRAY R,K,XC,YC[4:N-4] ;
        REAL X,P1,RS,KS,XCS,YCS ;
```

```
KS:=XCS:=YCS:=0 ;
FOR J:=4 STEP 1 UNTIL N-4 DO
  BEGIN X:=A+J×H ;
    P1:=3×S[J,0]×X↑2+2×S[J,1]×X+S[J,2] ;
    R[J]:=(1+P1↑2)↑1.5/M[J] ; K[J]:=1/R[J] ;
    KS:=KS+K[J] ; XC[J]:=X-P1×(1+P1↑2)/M[J] ;
    XCS:=XCS+XC[J] ; YC[J]:=Y[J]+(1+P1↑2)/M[J] ;
    YCS:=YCS+YC[J] ;
  END ;
KS:=KS/(N-1) ; XCS:=XCS/(N-1) ; YCS:=YCS/(N-1) ;
RS:=1/KS ;
P0165(1,
  ' - G    , -  , -  , -  , +  , -
  =15ПЕЧАТЬСРЕДНЕГОРАДИУСАRS,СРЕДНЕЙКРИВИЗНЫKS,
    ') ;
P0165(1,'=15СРЕДНИХКООРДИНАТЦЕНТРАКРИВИЗНЫXCS,YCS,
    ') ;
P0165(1,'=15КООРДИНАТЦЕНТРОВXC,YC,
  КРИВИЗНКИРАДИУСОВR') ;
P1041(RS) ;
P1041(KS) ;
P1041(XCS,YCS) ;
P1041(XC) ;
P1041(YC) ;
P1041(K) ;
P1041(R) ;
END ;
PO 165(1) (ПЕЧАТЬ S'); P1041(S); P0165(1,Ж,EQV,90); END
END; END; END; END; END
```

ПРОГРАММА ВЫЧИСЛЕНИЯ ТРЕУГОЛЬНОЙ СЕТКИ ПО ПРОИЗВОЛЬНО РАСПОЛОЖЕННЫМ ТОЧКАМ НА ПЛОСКОСТИ

ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

ИДЕНТИФИКАТОР ПРОГРАММЫ - ТРЕУГОЛЬНАЯ СЕТКА.

КЛЮЧЕВЫЕ СЛОВА - произвольная система наблюдения, М-222, М-220, БЭСМ-4, МБ.

ПРОЦЕССОР (тип ЭВМ) - программа разрабатывалась для ЭВМ М-222 с объемом памяти на магнитном барабане 76000 слов. Может быть использована на машинах М-220 и БЭСМ-4 с объемом МБ не менее 76000 слов.

СИСТЕМА ПРОГРАММИРОВАНИЯ:

- язык программирования АЛЬФА. Язык является конкретной реализацией алгоритмического языка АЛГОЛ-60 [2]. Все специфические особенности АЛЬФА-языка описаны в [4];
- транслятор - АЛЬФА для ЭВМ М-20, М-220, БЭСМ-4, М-222. Транслятор описан в [4];
- система архива - не используется;
- система редактирования - собственная система транслятора [4];
- система загрузки - собственная система транслятора [4];
- система ассемблирования - не используется;
- система графического отображения - не используется.

РЕДАКЦИЯ - I в 1974 г.

НАЗНАЧЕНИЕ ПРОГРАММЫ - построение системы треугольников, вершинами которых являются заданные точки наблюдения. Система строится таким образом, чтобы треугольники были по возможности близки к равносторонним.

СОДЕРЖАТЕЛЬНАЯ ПОСТАНОВКА ЗАДАЧИ - Один из наиболее простых и эффективных методов построения карт изолиний при значениях функции в произвольно заданных точках сводится к решению следующих задач:

а) построение треугольной сетки, в которой узлами являются заданные точки;

б) вычисление последовательностей точек линии одного уровня по треугольной сетке;

в) вычерчивание изолиний на устройстве с использованием алгоритма сглаживания ломаных или без него.

Эти задачи и метод предложены и решены рядом исследователей [3]. Ниже предлагается алгоритм построения треугольной сетки, при котором получающаяся сеть содержит треугольники, наиболее близкие к равносторонним и поэтому сама сеть не зависит от начала её построения. Для построения сети треугольников не требуется задание границы области определения точек. Процесс построения начинается с любой точки и по мере вычисления следующих треугольников происходит как бы кристаллизация, рост площади треугольной сетки, пока она не достигнет максимальных размеров, которые и дают границу определения точек.

Построение рельефа точек местности в виде многогранника из треугольников используют, например, Г.А.Лусков, В.Я.Никитенко и при обработке снимков наземной стереофотограмметрии. В [1,5] с использованием треугольной сетки предлагается метод вычисления объемом угольных складов.

Для получения предварительных результатов о поведении функции двух переменных, заданной в произвольно расположенных точках, можно использовать аппроксимацию её плоскими треугольниками. Во многих случаях, несмотря на погрешности при такой аппроксимации, можно получить необходимые сведения о поведении функции.

ПРОТОТИП - неизвестен.

ОБОРУДОВАНИЕ - Базовый комплекс ЭВМ, с памятью на магнитном барабане не меньше 76000.

ПОДПРОГРАММЫ - ABS , ARC COS , SQRT - стандартные процедуры.

ПАМЯТЬ - программа работает с ИС-2 и находится в нулевом кубе МОЗУ. На М-222 необходима также память для диспетчера ДМ-222.

МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА

МЕТОД РЕШЕНИЯ ЗАДАЧИ. Предлагается алгоритм построения треугольной сетки на заданных точках при следующих условиях:

вершинами треугольников могут быть только заданные точки; внутри построенных треугольников не должно быть заданных точек; каждый треугольник, если он не на границе области, имеет три смежных треугольника (смежные треугольники имеют общую сторону); при данном расположении точек треугольники должны быть наиболее близкие к равносторонним.

Построение базисного отрезка и выбор области для поиска точек

Пусть на плоскости заданы координаты точек x_i, y_i; $i = 1, \ldots, N$. Построение начинается с любой точки; возьмем точку $x_1 y_1$. Из всех остальных точек найдем ближайшую к первой. Это будет точка r_i, у которой рассеяние до первой точки $r_i = \sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2}$ будет меньше, чем у других.

Эти две точки будут составлять первый отрезок, с которого начинается построение треугольника. Назовем этот отрезок базисным и обозначим координаты его концов (x_a, y_a) и ($x_в$, $y_в$). Для этого базисного отрезка начнем поиск третьей точки для построения треугольника. Для ускорения поиска будем для треугольника просматривать не все точки, а только те, которые попадают в некоторую область вблизи базисного отрезка. Точки, попавшие в эту область, подвергаются последнему анализу. Определим область как сектор, образованный двумя расходящимися лучами, дугой и базисным отрезком. Лучи проведены из концов базисного отрезка и образуют с ним каждый угол в $3/4\pi$. Дуга образуется частью окружности между двумя лучами с центрами в точке "в" базисного отрезка и радиусом в 10 его длин. На рис. 1 лучи показаны пунктиром и область между ними заштрихована. Для компактности рисунка, радиус дуги взят меньше, чем требуется по условию.

В процессе проверки точек на попадание в описанную область возможны три случая:

1) в область не попадает ни одной точки;

2) попадает одна точка;

3) попадает несколько точек.

Первый случай означает, что либо базисный отрезок оказался на границе области задания точек, либо он короткий. Для устранения граничного эффекта достаточно произвести нумерацию точек "а" и "в". Это переводит сектор поиска точек в другую полуплоскость. Если опять не попало ни одной точки, то требуется в качестве базисного отрезка взять другие точки.

Во втором случае эта единственная точка берется для построения первого треугольника.

Если точек несколько, то проводится анализ на лучший треугольник.

Анализ точек для построения треугольника

Из точек, попавших в область поиска, берутся первые две. Пусть это будут точки с номерами i и j. Из этих точек нужно определить, какая из них даст с базисным отрезком лучший треугольник. Для этого на точках a, b, j, i строится четырехугольник. Для выяснения будет ли это четырехугольник $abji$ или $abij$, вычисляется угол $\angle abi$ и $\angle abj$ (рис. I). Пусть будет $\angle abj > \angle abi$, тогда строится четырехугольник $abji$. При невыполнении этого условия переиндексацией точек i и j добъемся его выполнения.

По условию внутри треугольников не должно быть ни одной заданной точки, поэтому следующей проверкой выясняется - не лежит ли точка i в треугольнике abj. Для этого сравнивается угол $\angle bai$ и $\angle baj$, и если $\angle bai \leq baj$, то из двух точек j и i точка i будет давать лучший треугольник с базисным отрезком, чем j, и она остается для сравнения со следующей точкой. Если $\angle bai > \angle baj$, то проверяется, какой диагональю разбить четырехугольник $abji$. Диагональ четырехугольника bi дает два одних треугольника, а диагональ aj - два других. Для выяснения нужной диагонали требуется произвести проверку каждой пары получающихся треугольников. Для этого найдем отклонения треугольников от равносторонних для каждой пары.

За отклонение треугольника от равностороннего возьмем сумму отклонений каждого из углов от угла равностороннего треугольника. А за отклонение угла α от равностороннего берем величину $(\alpha - \frac{\pi}{3})$.

В нашем случае (рис. I), в результате деления четырехугольника получаются либо пара треугольников $\triangle abi$ и $\triangle bji$ либо $\triangle abj$ и $\triangle aji$. По условию построения берем ту пару треугольников, для которой отклонение от равносторонних меньше.

При разбиении четырехугольника диагональю bi отклонения треугольника будут:

$$V_{\Delta abi} = (b_1 - \frac{\pi}{3})^2 + (i_2 - \frac{\pi}{3})^2 + (a_1 + a_2 - \frac{\pi}{3})^2$$
$$V_{\Delta bji} = (b_2 - \frac{\pi}{3})^2 + (i_1 - \frac{\pi}{3})^2 + (j_1 + j_2 - \frac{\pi}{3})^2 \qquad (I)$$

где угол b_1 при вершине "в" обозначен на рис. I; остальные углы не требуют пояснения. При разбиении диагональю aj отклонения треугольников следующие:

$$V_{\Delta abj} = (a_2 - \frac{\pi}{3})^2 + (j_2 - \frac{\pi}{3})^2 + (b_1 + b_2 - \frac{\pi}{3})^2$$
$$V_{\Delta aji} = (b_2 - \frac{\pi}{3})^2 + (i_1 - \frac{\pi}{3})^2 + (j_1 + j_2 - \frac{\pi}{3})^2 \qquad (2)$$

Нам следует вычислить разность:

$$S = (V_{\Delta abi} + V_{\Delta bji}) - (V_{\Delta abj} + V_{\Delta aji}) \qquad (3)$$

Подставляя (I), (2) в (3), получим

$$S = 2(a_1 a_2 + j_1 j_2 - b_1 b_2 - i_1 i_2) \qquad (4)$$

Если $S > 0$, то диагональ aj дает треугольники с меньшими отклонениями от равносторонних и точка j остается для сравнения её с другими точками. В противном случае остается точка i. Из точек, попавших в сектор, берется третья и сравнивается с точкой оставленной от предыдущих сравнений. В результате сравнения этих точек опять выбирается лучшая для последующего сравнения. Такой процесс длится, пока не проверены все точки, попавшие в сектор. В результате останется одна точка, которая прошла эти сравнения и явилась лучшей в описанном смысле для построения треугольника на базисном отрезке. Этот треугольник будет удовлетворять поставленным условиям построения. Запишем условия попадания точки в сектор и формулы для вычисления углов. Пусть координаты точки "а" (x_a, y_a), точки "в" (x_b, y_b), точки "j" (x_j, y_j), точки "i" (x_i, y_i). Как мы уже говорили, условия попадания точки „a" в сектор требуют, чтобы её расстояние до точки "в" было бы не больше десяти длин отрезка "ав". Это запишется неравенством:

$$\sqrt{(x_i - x_b)^2 + (y_i - y_b)^2} < 10\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (5)$$

Два других условия – это чтобы углы $\angle abi$ и $\angle bai$ были меньше $3/4\,\pi$ (135°). Угол $\angle bai$ равен углу между двумя векторами: один из них проведен из точки "в" в точку "а" и имеет координаты $(x_a - x_b;\ y_a - y_b)$ и второй – из точки "в" в точку "i" и имеет координаты $(x_i - x_b;\ y_i - y_b)$. Как известно из векторного исчисления, угол между двумя векторами можно вычислить из скалярного и векторного произведений этих векторов: $\angle abi = \angle b_1$

$$\sin b_1 = \frac{(x_a - x_b)(y_i - y_b) - (x_i - x_b)(y_a - y_b)}{\sqrt{[(x_a - x_b)^2 + (y_a - y_b)^2][(x_i - x_b)^2 + (y_i - y_b)^2]}}$$

$$\cos b_1 = \frac{(x_a - x_b)(x_i - x_b) + (y_a - y_b)(y_i - y_b)}{\sqrt{[(x_a - x_b)^2 + (y_a - y_b)^2][(x_i - x_b)^2 + (y_i - y_b)^2]}} \quad (6)$$

$$b_1 = \begin{cases} \arccos(\cos b_1) & \text{если } \sin b_1 \geq 0 \\ 2\pi - \arccos(\cos b_1) & \text{если } \sin b_1 < 0 \end{cases}$$

Аналогичным образом вычисляется угол $\angle iab = a_1 + a_2$.
Остальные условия попадания в сектор:

$$b_1 < \frac{3}{4}\pi\ ; \quad a_1 + a_2 < \frac{3}{4}\pi \quad (7)$$

Точки, удовлетворяющие неравенствам (5), (7), попадают в сектор. Если точка, попавшая в сектор, единственная, то на ней строится треугольник. Если их несколько, то среди них выбирается лучшая. Для этого по описанному выше способу строится четырехугольник на точке, оставшейся от предыдущего анализа и еще одной, попавшей в сектор при переборе. У построенного треугольника вычисляются все углы, необходимые для вычисления S по формуле (4). Углы вычисляются подобно углу в формуле (6).

b_2 – угол между векторами $(x_i - x_b;\ y_i - y_b)$; $(x_j - x_b;\ y_j - y_b)$
j_1 – угол между векторами $(x_b - x_j;\ y_b - y_j)$; $(x_a - x_j;\ y_a - y_j)$
j_2 – угол между векторами $(x_a - x_j;\ y_a - y_b)$; $(x_i - x_j;\ y_i - y_j)$
i_1 – угол между векторами $(x_j - x_i;\ y_j - y_i)$; $(x_b - x_i;\ y_b - y_i)$

$$a_1 = \pi - j_1 - b_1 - b_2 \; ; \qquad i_2 = \pi - b_1 - a$$

a_1 – угол между векторами $(x_i - x_a; y_i - y_a)$; $(x_j - x_a; y_j - y_a)$

Определив все эти углы и вычислив по (4) знак S, мы выясним, какую из двух точек оставить для последующего анализа. Перебрав все точки, найдем точку для построения треугольника на базисном отрезке.

Построение начальной границы

Построенный первый треугольник делит область заданных точек на две. В одной треугольники еще не построены, в другой – построены. Границей этих двух областей вначале являются стороны первого треугольника. Будем задавать границу в виде отрезков. Эти отрезки ориентируем так, чтобы направление от начала отрезка к его концу совпадало с направлением обхода границы против часовой стрелки.

Построение треугольника при наличии границы

Последующее построение треугольников начинается с того, что из границы берется отрезок, и он становится базисным. Начало этого отрезка обозначается через точку "а", а конец через точку "в". На сектор, который берется для предварительного анализа, накладывается еще одно требование, так как у базисного отрезка в границе могут быть смежные отрезки*). Если такие отрезки обнаруживаются, то сектор поиска сужается таким образом, чтобы смежные отрезки (а значит и обработанные уже точки) не входили в область дальнейшего поиска. Этим устраняется наложение треугольников.

Пусть к отрезку "ав" найдено два смежных отрезка "вс" и "da".

$\angle abc$ – угол между векторами $(x_a - x_b; y_a - y_b)$; $(x_c - x_b; y_c - y_b)$

$\angle dab$ – угол между векторами $(x_d - x_a; y_d - y_a)$; $(x_b - x_a; y_b - y_a)$.

Вместо неравенства (7) попадание точки "b" в сектор дается неравенствами

---

*) Смежными называются отрезки, имеющие общую точку.

$$b_1 < \min\left\{\frac{3}{4}\pi;\ \angle abc\right\};\ a_1 + a_2 < \min\left\{\frac{3}{4}\pi;\ \angle dab\right\} \quad (8)$$

Неравенствами (8) отбрасываются точки, на которых уже построены треугольники. Все точки последовательно проверяются на неравенства (5) и (8) и, удовлетворяющие этим неравенствам, подвергаются дополнительному анализу на четырехугольник, описанному при построении первого треугольника. Кроме изменения неравенства (7), построение треугольника на базисном отрезке идентично построению первого треугольника. В результате этих проверок точка для треугольника либо найдена, либо такой точки не оказывается.

Изменение границы

В процессе построения треугольников граница, определяющая область, где треугольники уже построены, от области, где они еще не найдены, постоянно изменяется, пока не дойдет до внешней границы рассматриваемой области. Это будет конечная граница. В процессе построения будем различать отрезки,которые уже достигли границы области задания точек. Такие отрезки мы будем выделять в отрезки конечной границы и в дальнейшем не будем брать их в качестве базисных отрезков. Из отрезков, которые не входят в конечную границу и которые составляют переменную часть раздела двух областей, будем брать базисные отрезки и для них искать третью точку для треугольника. Если для базисного отрезка не нашлась точка для треугольника, то это означает, что он вышел на конечную границу, куда он и переводится.

После построения треугольника граница меняется. Если точка треугольника не нашлась, то базисный отрезок из отрезков границы переводится в отрезки конечной границы. Конечная граница является границей области определения точек.

Если точка для треугольника нашлась, то граница раздела построенных и непостроенных треугольников должна сдвинуться, учитывая вновь появившийся треугольник. Для этого каждая сторона вновь построенного треугольника проверяется, имеется ли в границе отрезок, совпадающий с этой стороной? Если такой отрезок имеется в границе, то он исключается. Если его нет, то ищется такой отрезок в конечной границе. Если такой отрезок отыскивается там, то он исключается из конечной границы. Если отрезка, совпадающего со стороной треугольника нет в обеих границах, то он включается в границу. По условию построения треугольника, если она не на конечной границе, принадлежит не более, чем двум треугольникам. Каждая сторона треугольника должна включаться в границу ориентированной, чтобы сохранить направление обхода. Если на базисном отрезке "ав" построен треугольник Δabj , то изменение границы происходит в следующей последовательности:

I) отрезок "ав" исключается из границы, так как он там имеется;

2) происходит проверка наличия отрезка "aj" в границе и в конечной границе, если где-то он имеется, то оттуда исключается. Если нет, то включается в границу с ориентацией: точка "а" - начало, точка "j" - конец отрезка;

3) такой же проверке подвергается отрезок "jb" и если включается в границу, то с ориентацией: точка "j" - начало, точка "в" - конец отрезка.

Таким образом, либо треугольник построен и граница передвинута, либо треугольник не найден, базисный отрезок из границы переводится в конечную границу.

Для построения следующего треугольника из границы берется следующий отрезок, по описанному алгоритму отыскивается треугольник и меняется граница.

Построение треугольников длится до тех пор, пока в границе остается хоть один базисный отрезок. Отсутствие отрезков означает, что все треугольники построены.

Треугольная сетка получается в виде номеров заданных точек: каждый треугольник определяется номерами точек его вершин.

Построение треугольной сетки можно видоизменить, если конечная граница задается. В этом случае не требуется построения первого треугольника. Если конечную границу задать отрезками, с ориентацией обхода области по часовой стрелке, используя тот же алгоритм и выбирая базисные отрезки из отрезков конечной границы, можно также построить треугольники.

КРИТЕРИИ ОЦЕНКИ РЕЗУЛЬТАТОВ ВЫЧИСЛЕНИЙ - не предлагаются

ТАБЛИЦА СООТВЕТСТВИЯ

Идентификатор	Тип	Комментарий
M	целый	адрес записи на МБ
ЛО	целый	количество точек наблюдения
K		
K, I	целый	рабочие ячейки
J	целый	начальный индекс краевой границы
M3		
N	целый	верхний индекс некраевой границы
L15	логическ	треугольник построен?
XC УС	веществен	рабочие ячейки вычисления расстояний
ЭП	веществен	малая величина
-X	веществен массив	x - координаты точек наблюдения
-У	веществен массив	у - координаты точек наблюдения
Д, Д1, Д2, Д3	веществен	рабочие ячейки
-K	целый массив	буфер построенных треугольников
-X1	целый массив	массив границ
ПТР	процедура	процедура построения треугольника
УГОЛ	процедура	процедура расчета угла между двумя векторами
Т‾0	процедура	процедура обработки массива границы
L	целый	счетчик заполнения буфера треугольников

Описание блок-схемы программы
ТРЕУГОЛЬНАЯ СЕТКА

Программа начинается с ввода количества точек наблюдения ЛО. Затем вводятся массивы координат по X и по У. Дальше происходит поиск первого отрезка для первого треугольника. Для этого вычисляется расстояние между первой и второй точками. Расстояние вычисляется по формуле $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Затем проводится засылка констант, К-присвоение, Е присваивается значение равное 10^{-8} и М (начало записи на барабан) - 25003. Потом зануляется массив границы: XI = 0 и присваиваются первые значения переменным К, КI и L. Далее перебором всех точек введенного массива отыскивается точка ближайшая к первой. Для этого организуется цикл начиная от 3-х до ЛО, и вычисляются расстояния от всех этих точек до первой точки. Это расстояние сравнивается с накопленным расстоянием, т.е. всегда условным оператором проверяется, какие расстояния меньше, накопленные или только что вычисленные. Затем производится первоначальное присвоение отрезкам ДI и Д3 величины минимального расстояния между точками.

Блоком 6 производится построение треугольника. Построение треугольника производится с помощью процедуры ПТР. Процедура ПТР может либо найти точку для треугольника, либо не найти. Найдена или не найдена точка - это показывает логическая величина L I5. Если L I5 = истина, то тогда точка найдена и следующий оператор - блок 9. Если L I5 = ложь, то точка не найдена, производится перенумерация первого отрезка и переход на поиск точки для треугольника. Если точка найдена, то производится построение границы. Массив границы состоит из номеров составляющих его отрезков.

В первоначальном случае, когда имеется треугольник из номеров точек К, КI и J производится построение первичной границы. Затем производится первоначальное присвоение констант М3 и N. М3 присваивается значение 300 и N присваивается значение 4. После этого алгоритм работает следующим образом; начиная от метки Я3 (блок I0) производится выбор первого отрезка из границы. Как уже сказано раньше, отрезками границы являются номера составляющих его точек. Затем проверяется равен ли этот отрезок нулю или нет. Если отрезок нулевой, то это означает, что отрезки границы все кончились и все треугольники построены. Если все треугольники построены, то программа выходит на метку ЯО (блок 24). Если же отрезки еще имеются в границе, то производится построение точки треугольника, работает процедура ПТР.

После работы процедуры ПТР снова производится проверка логической величины L I5. Проверяется, была ли найдена точка для треугольника или нет. Если точка не была найдена, т.е. L I5 = <u>ложь</u>, тогда работает процедура ГО, которая в данном случае выбрасывает первый отрезок с номерами К и КI из массива границы. Если точка для треугольника была, то производится сдвиг границы процедурой ГО. Процедура ГО, как сказано выше, работает с границей, её работа будет пояснена дальше.

Если треугольник найден, тогда три раза работает процедура ГО. Она отрабатывает отрезки КIК, KJ и JKI. Кроме этого найденные точки засылаются в массив треугольников К с первым индексом L . Первое значение индекса (L) указывает номер треугольника, с которым идет работа, а второе значение - индекс I, 2,3, это заполнение буферов треугольников. Дальше идет проверка заполнения буфера и, если количество треугольников в буфере К равно I0, то производится запись массива треугольников на барабан. Запись производится без контрольной суммы. При этом М равняется адресу на барабане. После завершения работы с буфером накопленных треугольников переходим к обработке следующего участка границы. Опять выбирается первый отрезок из границы, проверяется равен он нулю или нет, ищется точка для треугольника, после чего работает процедура ПТР и снова производится работа с границей.

Работа с границей состоит в том, что если отрезок находится в массиве границы, то он из него исключается; если же отрезка в границе нет, то он туда заносится.

Начиная с метки ЯО (блок 24) производятся всевозможные преобразования, связанные с окончанием работы. Во-первых, записывается остаток массива треугольников на барабан; затем записываются Х-овые координаты точек наблюдения на барабан; затем У-овые координаты точек; выводится количество построенных треугольников; также записывается на барабан число точек и число построенных треугольников, причем предварительно вычисляются и записываются на барабан минимальные и максимальные значения координат точек по Х и по У. На этом программа заканчивает работу.

Описание блок-схемы подпрограммы ПТР (поиск точки для треугольника)

Процедура "Поиск точки для треугольника" работает с отрезком ККI. Для этого отрезка производится поиск третьей точки для треугольника. Первоначально для отрезка ККI ищутся смежные отрезки. Для этого просматривается массив построенной границы. Отрезок, смежный с отрезком ККI, может быть со стороны точки КI, т.е. у них общая точка КI, либо со стороны точки К, у них общая точка К. Таким образом производится поиск отрезков, смежных с отрезком ККI со стороны точки КI и со стороны точки К. Если такие отрезки находятся, то вычисляется угол между каждым из смежных отрезков и отрезком ККI. Тем самым определяется область, в которой требуется отыскать точки для треугольника. Далее производится поиск точки для треугольника. Вначале вычисляется расстояние между точками К и КI. Затем вычисляются расстояния ДI и Д2. Эти величины необходимы для того, чтобы вычислить в процессе построения треугольной сетки среднее расстояние отрезка треугольной сетки.

Далее все точки, начиная с первой и до последней ЛО, проходят проверку на возможность их использования для построения треугольника. Для этого вначале проверяется не равна ли эта точка К или КI. Если равна, то она проверку не проходит. Затем вычисляются расстояния между текущей точкой i и точкой КI. Если это расстояние меньше, чем Р3, вычисленное раньше, то тогда оно проходит дальнейшую проверку.

Дальнейшая проверка заключается в том, что вычисляется расстояние от текущей точки i до точки К. Это расстояние сравнивается с расстоянием Р3. Расстояние Р3 и углы, которые вычислялись раньше, дают сектор поиска точек. Сектор поиска определяется таким образом, что расстояние его границы от отрезка ККI не больше, чем Р3, а углы вычисляются путем поиска смежных отрезков. Если смежные отрезки не найдутся, то угол равен $3/4\pi$. Если же смежные отрезки находятся и их угол меньше, чем $3/4\pi$, тогда ограничиваются этим углом. Проверив расстояние от текущей точки i до точки КI и до точки К, и выяснив, что это расстояние меньше, чем РЗ, делаем дальнейшую проверку на возможность построения. Для этого вычисляется угол между отрезком iK и отрезком ККI. Если вычисленный угол лежит в заданных пределах, то производится дальнейшая проверка, была ли найдена до этой точки еще хотя бы одна точка в заданном секторе. Если она была найдена, то производится построение четырехугольника и проверка его углов. Здесь производится выбор такой точки, диагональ через которую разбивает четырехугольник на треугольники ближе к равносторонним.

Таким образом, процедура построения точки треугольника, перебирая все исходные точки, производит следующую проверку. Во-первых, вначале проверяется попадает ли эта точка в сектор. Затем, если точка попала в сектор, проверяется была ли там еще хоть одна точка. Если такая точка была, то производится проверка на четырехугольник, т.е. начальный отрезок ККI и две попавшие в сектор точки составляют четырехугольник и проверяется какой диагональю выгоднее разбить этот четырехугольник, т.е. какая диагональ дает отклонение двух треугольников от равносторонних меньше. В соответствии с этим разбиением выбирается точка, дающая два треугольника близких к равносторонним.

Описание блок-схемы подпрограммы обработки границы ГО

Процедура ГО работает с границей. Она имеет формальный параметр - отрезок ККI. Процедура работает следующим образом. Она проверяет, имеется ли отрезок в границе, сходный с отрезком ККI Этот отрезок ищется в обеих частях массива границы. Отдельно проверяется массив отрезков границы от I до N -I и массив отрезков границы от МЗ до конца. От I до N -I в массиве границы находятся отрезки, которые не выходили на конечную границу, а начиная от номера МЗ и дальше находятся отрезки, которые составляют конечную границу.

Процедура ГО проверяет, находится ли отрезок ККI в массиве границы от I до N -I. Если находится, то он исключается из массива границы. Если же в массиве границы от I до N -I отрезка нет, то производится проверка от МЗ до конца. Если там тоже нет отрезка, тон он включается в границу под номером N и затем N сдвигается на единицу дальше. Если же отрезок находится или от I до N -I, или от МЗ до конца (т.е. до 300), то этот отрезок исключается, и проверка считается законченной.

Таблица распределения памяти

	начало	конец	длина	примечания
программа	0020	II70	II5I	
числовые константы	II7I	I224	0034	
формируемые константы	I225	I250	0024	
скаляры	I25I	I325	0055	
рабочие ячейки	I326	I336	00II	
массивы				
I	I337	36I6	2260	
2	36I7	6076	2260	
3	6077	7226	II30	
4	7227	7264	0036	
5	XXXX	XXXX	XXXX	не распределен
6	XXXX	XXXX	XXXX	совмещен с массивом 3 со сдвигом 0000

рабочее поле ИС	начало	конец	длина
	7265	7500	02I3

адреса программных остановов

II70

ВРЕМЯ СЧЕТА - работа программы для 300 точек потребовала семь минут, для 24 точек - 3 минуты.

ОПЕРАЦИОННАЯ СИСТЕМА. На машинах М-220 и БЭСМ-4 программа может работать автономно. На ЭВМ М-222 программа работает с диспетчером ДМ-222.

ОГРАНИЧЕНИЯ - максимальное количество обрабатываемых точек I200.

К ограничениям алгоритма можно отнести невозможность описания поверхности, выходящей за пределы выпуклой оболочки двумерного пространстра, образуемого координатами точек, а также детерминистическую постановку задачи, когда признаки в точках предполагаются точно известными. И кроме того существуют трудности в оценке точности восстановленной функции.

ТОЧНОСТЬ метода повышается с увеличением плотности исходных точек и зависит от точности, с которой определяется значение признаков каждой точки.

ИНСТРУКЦИИ К ПРОГРАММЕ

ИНСТРУКЦИЯ ПРОГРАММИСТУ. Хорошо изучить АЛЬФА-язык и разобраться в программе, прежде чем что-либо переделывать.

Рабочая программа может быть введена с перфокарт или вызвана с магнитной ленты (в зависимости от организованного режима эксплуатации). Кроме этого существует возможность трансляции исходного модуля программы с последующим счетом.

ИНСТРУКЦИЯ ПО ПОДГОТОВКЕ ИСХОДНОЙ ИНФОРМАЦИИ ДЛЯ ВВОДА В ЭВМ. Исходная информация готовится на перфокартах и состоит из трех массивов перфокарт. За каждым из массивов должна следовать перфокарта контрольной суммы массива.

Первый массив состоит из одного целого числа ЛО, определяющего количество точек наблюдения.

Второй массив - ЛО вещественных чисел, определяющих X - координаты точек наблюдения.

Третий массив по строению и количеству чисел аналогичен второму. Он определяет У - координаты точек наблюдения.

Все числа перфорируются в формате числа М-20. При этом каждый новый массив должен начинаться с новой перфокарты.

Образец перфорации одного числа:

+++ 02 523 где

первый + - признак числа
второй + - знак числа
третий + - знак порядка
02 - порядок числа
523 - мантисса

ИНСТРУКЦИЯ ОПЕРАТОРУ. Сообщения оператору в программе не предусмотрены.

Программный останов в ячейке 1170_8 - конец задачи.

ПОЯСНЕНИЯ К РЕЗУЛЬТАТАМ РЕШЕНИЯ

На широкую печать выводится только одно число - количество построенных треугольников, остальные результаты записываются на магнитный барабан.

в адрес 24996(IO) - минимальное значение координаты по X-XMIN ;

в адрес 24997 - минимальное значение координаты массива точек по У;

в адрес 24998 - максимальное значение координаты по X-XMAX;

в адрес 24999 - максимальное значение координаты по У-УMAX;

в адрес 25000 - число точек;

в адрес 2500I - количество построенных треугольников;

в адрес 25002 - средняя длина стороны треугольника;

с адреса 25003 и далее - номера точек, вершин треугольников.

Если число точек обозначить N , а число треугольников - - M и координаты точек в массивах пронумеровать от I до N , то треугольник из номеров 5, 8, 2 означает, что его вершинами являются точки, координаты которых отыскиваются по соответствующим номерам.

Начиная с адреса 25003 + M x 3 расположен массив координат точек по X;

Начиная с адреса 25003 + M x 3 + N расположен массив координат точек по У.

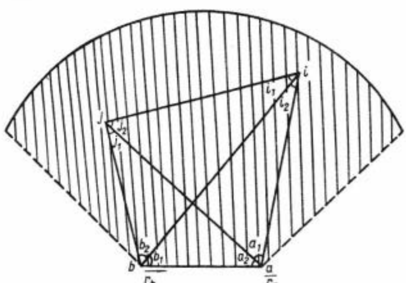


Рис. 1

КОНТРОЛЬНЫЙ ПРИМЕР

Таблица входных данных для проверки работы программ

N =24

N	I	2	3	4	5	6	7	8	9	I0	II	I2
X	I	2	3	I	2	3	3	2	5	4	5	4
У	6	6	6	8	I0	I0	I2	I4	I6	I4	I2	I0
N	I3	I4	I5	I6	I7	I8	I9	20	2I	22	23	24
X	4	3	2	4	3	I0	3	I	2	4	5	5
У	8	8	I5	I6	I6	I4	I4	I2	I2	I2	8	I0

В результате обработки этого материала на печать выдается одно число - количество построенных треугольников:

$$+ \quad 360000000_{I0} \quad + 02$$

Вся остальная информация после работы программы остается на МБ и может быть проверена включением программы ПОСТРОЕНИЕ ТРЕУГОЛЬНОЙ СЕТКИ НА ГРАФОПОСТРОИТЕЛЕ. Эта программа выполняет разрисовку системы треугольников (построенных программой ТРЕУГОЛЬНАЯ СЕТКА) и может быть использована в качестве "жесткого" теста программы ТРЕУГОЛЬНАЯ СЕТКА.

СПИСОК ЛИТЕРАТУРЫ

I. А.С.ЗАБЫШНЫЙ "Об одной модели поверхности земли для трассирования дорог на ЭВМ", Изв. ВУЗов, Геодезия и аэрофотосъемка, № 6, I970, стр. 48-5I.

2. С.С.ЛАВРОВ "Универсальный язык программирования АЛГОЛ-60, М., I964.

3. В.И.СТАРОСТЕНКО, Р.Г.БАС, Г.С.БУТАКОВ, В.А.ДЯДЮРА. Автоматизированная система оперативной обработки данных гравиметрии и магнитометрии. "Наукова Думка", Киев, I972.

4. Руководство к использованию системой АЛЬФА, НГУ, Новосибирск, I973.

5. (Франция), Compagnie Internationale pour l'Informatique.Louveciennes.

ТЕКСТ ПРОГРАММЫ

```
начало целый M, Л0, K,K1,I,J,L;
целый M3, N;
вещ эп, хс, ус; логич L15;
массив ¯X,¯Y [1:1200] ;
вещ D, D1, D2, D3;
целый массив ¯K [1:10,1 :3] ,¯X1 [1:300,1:2] ;
массив * дельта * [1:8192] ; барабан * дельта * ;
процедура Птр ; начало вещ A,A1,A2,B1,B2,I1,I2,J1,
J2,пи ; целый K2;
процедура ¯Угол (K,K1,K2,A) ; начало вещ XK, YK, ZK, UK, VK, AK;
XK:=¯X [K] - ¯X [K1] ; YK:=¯Y [K] -¯Y [K1] ;
ZK:=¯X [K2] -¯X [K1] ; UK:=¯Y [K2] -¯Y [K1] ;
A:= SQRT ((XK↑2+YK↑2) * (ZK↑2+UK↑2)) ;
VK:= (XK*UK-ZK*YK)/A; AK:= (XK*ZK+YK*UK)/A ;
если ABS(ABS(AK)-1) < эп то AK:= OKP(AK) ;
если VK > 0 то A:=ARCCOS(AK) иначе A:=2*ПU-ARCCOS(AK) конец ;
```

```
ПU:=3,141593 ; L15:= ложь ; B1:= 3×ПU/ 4 ;
K2:=J :=0; XC:=YC:=B1 ; для I:=1,...,N,M3,...,300 цикл начало
если ˉX1[I,1] =K1 and ˉX1[I,2]≠K то начало K2:=ˉX1 [1,2] ;
ˉУгол (K,K1,K2,D) ; если D< XC то XC:=D конец ;
если ˉX1 [I,2]=K1 and ˉX1[I,1]≠ K то начало K2:=ˉX1[I,1] ;
ˉУгол (K,K1,K2,D) ; если D<XC то XC:= D конец ;
если ˉX1[I,2]=K and ˉX1[I,1]≠ K1 то начало J:=ˉX1[I,1] ;
ˉУгол (J,K,K1,D) ; если D<YC то YC:= D конец ;
если ˉX1[I,1]=K and ˉX1[I,2] ≠ K1 то начало J:=ˉX1[I,2] ;
ˉУгол (J,K,K1,D) ; если D<YC то YC:=D конец ;
конец ;
D:=SQRT ((ˉX [K] -ˉX [K1]↑ 2+(ˉY [K] -ˉY [K1])↑ 2) ; J:=0 ;
D1:=D1+D; D2:=D2+1 ; если D3<D1/D2×3 то D3:= D1 / D2×3 ;
если D3>D×4 то D3:=D×4 ;
для I:=1,...,10 цикл начало если I=K or I=K1 то на ˉS2 ;
B1:=SQRT ((ˉX[I]-ˉX[K1] )↑2+(ˉY[I]-ˉY[K1] )↑2) ; если B1< D3
то начало
D:=SQRT ((ˉX [I] -ˉX [K] )↑2+(ˉY [I] -ˉY [K] )↑ 2) ;
```

```
если D>D3 то на ˉS2;ˉУгол (K,K1, I, B1);если (B1-XC) < эп то начало
ˉУгол (I,K,K1,A) ; если (A-YC) < эп то начало если L15 то начало
если ABS(B1-I1) < эп то началоь если A2>A то начало J:=I ;
A2:=A ; конец ; на ˉS2 конец ;
если ABS(A-A2) < эп то начало если B1<I1то начало J:=I ;
I1:=B1 конец ; на ˉS2 конец ;
если B1 > I1 то начало K2:=I ; I:=J ; J:=K2 ;
J2:= A; A:= A2; A2:= J2; J2:=I1; I1:=B1; B1:=J2;
конец ; B2:=I1 - B1 ;
если A<A2 то начало A2:=A; I1:=B1; K2:=J; J:=I; I:=K2 ; на ˉS2 конец ; A1:= A-A2 ;
J1:=пи -A2-I1;I2:= пи -B1 -A;
ˉУгол(K,J,I,J2) ;ˉУгол (J, I,K1,I1) ;
если (A1×A2+J1×J2-B1×B2-I1×I2 )<0 то начало K2:=I; I:=J; J:=K2; I1:=B1; A2:=A2+A1конец
иначе I1:=B1+B2конец иначе начало I1:=B1; A2:=A; J:=I; L15:= истина конец
конец конец конец ;ˉS2:еслиI< J то I:=J конец конец
процедура ˉГˉО(K,K1); начало логич L11 ;
```

```
L11:= ложь для I:=1,...,N-1 цикл начало
если ˉX1[I,1] = Kand ˉX1[ ,2]=K10R ˉX1 [I,1]=K1andX1 [I,2]=KTO
L11:= истина если L11 то начало ˉX1[I,1]:=X1[I+1 ,1];
ˉX1[I,2]:=ˉX1[I+1,2] конец конец ;
если L11 то начало N:=N-1; на ˉL2 конец ;
для I:=300,I-1 пока I>M3 цикл начало
если ˉX1[I,1]= Kand ˉX1[I,2]=K1OR ˉX1[I,1]= K1and -X1[I,2]=K то
L11:= истина; если L11 то начало ˉY1[I,1]:=ˉX1[I-1 ,1];
ˉX1[I,2]=ˉX1[I-1,2] конец ;если L11 то начало если M3<300 то M3:=M3+1; на ˉL2 конец ;
ˉY1[N,I]:=K; ˉX1[N,2]:=K1; N:=N+1;
ˉL2: если NOTL15 то начало ˉX1[M3,1]:=K; ˉX1[M3,2]:=K1 ;
M3:=M3-1 конец конец ;
ввод (Л0) ;
ˉC ˉ 0176 (1040,ˉX[1],ˉX[Л0], 0,0 ) ;
ˉC ˉ 0176 (1040,ˉY[1],ˉY[Л0], 0,0 ) ;
XC:= SQRT ((ˉX[1]-ˉX[2])↑ 2+(ˉY[1]-ˉY[2])↑2 ) ;
K:=1; K1:=L:=2 ; ЭП :=10-8; M:=25003; ˉX1[,]:=0 ;
```

```
для I:=3,..., ЛО цикл начало УС:=SQRT ((¯X[1]-X[I])↑2 +
(¯Y[1]-Y[I])↑2 ) ; если YC<XC  то начало K1:=I; XC:=YC  конец конец ;
M3:=300; N:=4 ;
D1:=D3:=XC+  эп ; D2:=1 ;¯S:
¯ПТР  если NOTL15  то начало  I:=K; K:=K1; K1:=I;  на ¯S  конец  ;
¯X1 [1,1]:=¯X1[3,2]:=¯K[1,2]:=K1 ;
¯X1[1,2]:=¯X1[2,1]:=¯K [1,1]:=K  ;
¯X1[2,2]:=¯X1 [3,1]:=¯K [1,3]:=J  ;
¯Я3:K:=¯X1[1,1]; K1:=¯X1[1,2];        если  K=0 то на ¯Я0 ;
¯ПТР; если NOT L15то ¯Г¯O (K1,K)  иначе начало  ¯Г¯O (K1,K) ;¯Г¯O (K,J) ;
¯Г¯O(J,K1); ¯K[L,1]:=K ;¯K[L,2]:= K1;¯K[L,3]:=J;
если L= 10  то начало ¯С¯П0177(1037 , ¯K,[,]¯K[,], 0, *дельта*[M] ) ;
M:=M+3
0; L:=1  конец иначе  L:=L+1  конец  ; на ¯Я3  ;
¯Я0 : L:=L-1; если L>0 то начало
¯С¯П 0177(1037, ¯K [,],¯K[L,3] ,0,*дельта*[M] ) конец  ;
M:=M+L*3; L:=(M-25003)/3;
```

```
¯С¯П 0177(1037,¯X[ ],¯X[ЛО],0, *дельта *[M]) ;
M:=M + ЛО ;
¯С¯П 0177(1037,¯Y[ ],¯Y [ЛО],0, *дельта *[M]);
M:=L ;
вывод (L) ; L:= 25000 ;
¯С¯П 0177(1037,ЛО ,ЛО ,0,*дельта * [L]) ;
¯С¯П 0177(1037,M,M,0, *дельта * [L + 1] ) ;
начало массив U [1:4] ;
U[1]:=U[3]:=¯X[1] ; U[2]:= U[4]:=¯Y[1] ;
для I:=2,... , ЛО цикл начало если U[1]>¯X[I] то U[1]:=¯X[I] ;
если U[3]<¯X[I] то U[3]:=¯X[I] ;
если U[2]>¯Y [I] то U[2]:=¯Y[I];
если U[4]<¯Y[I] то U[4] :=¯Y[I] конец ;
¯С¯П 0177 (1037, U[ ], U[ ],0,дельта *[L-4]) ; Д:=Д1/Д2 ;
¯С¯П 0177 (1037, D,D,0,дельта *[L + 2]) конец ;
конец
ЕОП)
```

У. КОМПЛЕКС СТАТИСТИЧЕСКИХ ПРОГРАММ

ПРОГРАММА "НАТУРАЛЬНЫЙ ЛОГАРИФМ" (ПI)

(Язык альфа, ЭВМ М-220, М-222)

НАЗНАЧЕНИЕ

Программа производит вычисление натуральных логарифмов с выдачей результатов логарифмирования на перфорацию (на перфокарты).

ИНСТРУКЦИЯ К ПОЛЬЗОВАНИЮ

Программа рассчитана на обработку массива информации объемом $q = m \times N \leq 1500$ знаков (в десятичной системе), где m - число признаков, N - число наблюдений.

Подготовка данных

Массив информации перфорируется по строкам (строка за строкой) в десятичной системе. Дополнительно к массиву информации пробиваются две информационные карты. На первой информационной карте пробивается число признаков в десятичной системе с правильной $\kappa\Sigma$ для этой карты. На второй информационной карте пробивается число наблюдений в десятичной системе с правильной $\kappa\Sigma$.

Порядок постановки перфокарт для счета

1. Программа с контрольной суммой.
2. Первая информационная карта.
3. Вторая информационная карта.
4. Массив информации с контрольной суммой.

На печать выдаются следующие величины: число признаков, число наблюдений, исходный массив данных и массив прологарифмированных данных.

На перфорацию выдаются результаты логарифмирования, причем перфорация строк идет подряд одна за другой.

Необходимо напомнить для пользователей, что логарифм числа, меньшего единицы – отрицательное число.

КОНТРОЛЬНЫЙ ПРИМЕР

Исходная таблица (Т)

№	x_1	x_2	x_3	x_4
I	7,17	30,30	36,98	2,78
2	18,09	10,86	32,95	12,17
3	32,84	10,27	29,49	13,85
4	1,09	0,59	0,50	0,42
5	0,42	0,59	0,42	0,42
6	0,50	0,67	0,50	0,33

где N – число наблюдений, x_1, x_2, x_3, x_4 – признаки.

Исходная информация, подготовленная к счету

Информационная карта № I

π	КОП	A_I	A_2	A_3		
+	0I	4000	–	–		m
+	0I	4000	–	–	к Σ	

Информационная карта № 2

π	КОП	A_I	A_2	A_3		
+	0I	6000	–	–		N
+	0I	6000	–	–	к Σ	

Массив исходных данных, перфорирующийся по строкам

π	КОП	A_I	A_2	A_3
+	0I	7I70	-	-
+	02	3030	-	-
+	02	3698	-	-
+	0I	2780	-	-
+	02	I809	-	-
:	:	:	:	:
+	00	5000	-	-
+	00	3300	-	-

T

РЕЗУЛЬТАТЫ СЧЕТА

На печать:

I. Число признаков - m
2. Число наблюдений - N
3. Исходная таблица - T
4. Таблица прологарифмированных данных - T′

№	x_1	x_2	x_3	x_4
I	I,969	3,4II	3,6I0	I,022
2	2,895	2,385	3,495	2,499
3	3,492	2,329	3,384	2,628
4	0,086	-0,527	-0,693	-0,867
5	-0,867	-0,527	-0,867	-0,867
6	-0,693	-0,400	-0,693	-I,I08

На перфорацию: таблица прологарифмированных данных (T′) на перфокартах.

```
НАЧАЛО
ЦЕЛЫЙ ¯N,M; ВВОД(¯N,M); НАЧАЛО
ВЕЩ B;
ЦЕЛЫЙ I,J,K;
МАССИВ ¯B,¯A[1:¯N,1:M],¯C[1:¯N×M];
ВВОД(¯A);
ВЫВОД(¯N,M,¯A);
ДЛЯ I:=1 ШАГ 1 ДО ¯N ЦИКЛ
НАЧАЛО ДЛЯ J:=1 ШАГ 1 ДО M ЦИКЛ НАЧАЛО B:=LN(¯A[I,J]); ¯B[I,J]:=B КОНЕЦ КОНЕЦ ; ВЫВОД(¯B);

K:=1; ДЛЯ I:=1 ШАГ 1 ДО ¯N ЦИКЛ
НАЧАЛО ДЛЯ J:=1 ШАГ 1 ДО M ЦИКЛ НАЧАЛО ¯C[K]:=¯B[I,J]; K:=K+1 КОНЕЦ КОНЕЦ ;
¯C¯П0176(1281,¯C[1],¯C[¯N×M],0,0);
СТОП ;
КОНЕЦ ;
КОНЕЦ *
<-15'Ш
```

ПРОГРАММА "ПЕРЕВОД ДАННЫХ" - П2.

Составлена на языке АЛЬФА для ЭВМ типа М-220, М-222

НАЗНАЧЕНИЕ

Программа предназначена для перевода массива информации с перфорацией по строкам в массив с перфорацией по столбцам или наоборот с выдачей результата на перфорацию (на перфокарты).

ИНСТРУКЦИЯ К ПОЛЬЗОВАНИЮ

Программа рассчитана на обработку массива информации объемом $q = m \times N \leq 12000$ знаков (в десятичной системе), где m - число признаков, N - число наблюдений.

Подготовка данных

Массив информации может быть отперфорирован по строкам или по столбцам в десятичной системе. Дополнительно к массиву информации пробиваются две информационные карты. Если необходимо перевести массив информации с перфорацией по строкам в массив с перфорацией по столбцам, то на первой информационной карте пробивается число наблюдений в десятичной системе с правильной $\kappa\Sigma$ для этой карты. На второй информационной карте пробивается число признаков в десятичной системе с правильной $\kappa\Sigma$ для этой карты. Если же необходим перевод массива информации с перфорацией по столбцам в массив с перфорацией по строкам, то информационные карты меняются местами.

Порядок постановки перфокарт для счета

1) Программа с контрольной суммой.
2) Первая информационная карта.
3) Вторая информационная карта.
4) Массив информации с контрольной суммой.

На печать выдаются величины: число признаков, число наблюдений. Результат перевода выдается на перфорацию. Причем столбцы или строки выдаются на перфорацию подряд.

КОНТРОЛЬНЫЙ ПРИМЕР

Исходная таблица (Т)

№	x_1	x_2	x_3	x_4
I	7,I7	30,30	36,98	2,78
2	I8,09	I0,86	32,95	I2,I7
3	32,84	I0,27	29,49	I3,85
4	I,09	0,59	0,50	0,42
5	0,42	0,59	0,42	0,42
6	0,50	0,67	0,50	0,33

где N – число наблюдений, x_1, x_2, x_3, x_4 – признаки.

Исходная информация, подготовленная к счету. Рассмотрим случай перевода строчечной перфорации в перфорацию по столбцам.

Информационная карта № I

π	КОП	A_I	A_2	A_3
+	0I	6000	–	–
+	0I	6000	–	– к Σ

N

Информационная карта № 2

π	КОП	A_I	A_2	A_3
+	0I	4000	–	–
+	0I	4000	–	– к Σ

m

Массив исходных данных, перфорирующийся по строкам

π	КОП	A_I	A_2	A_3
+	0I	7I70	–	–
+	02	3030	–	–
+	02	3698	–	–
+	0I	2780	–	–
+	00	5000	–	–
+	00	3300	–	–

T

РЕЗУЛЬТАТЫ СЧЕТА

На печать:

1. Число наблюдений – N
2. Число признаков – m

На перфорацию: Производная таблица (T')

π	КОП	A_I	A_2	A_3
+	0I	7I70	–	–
+	02	I809	–	–
+	02	3284	–	–
+	0I	I090	–	–
.	.	.	.	.
+	02	I385	–	–
+	00	4200	–	–
+	00	4200	–	–
+	00	3300	–	–

T

```
 НАЧАЛО  ЦЕЛЫЙ ГРАН,I;  ВЕЩ CFIN;
 ЦЕЛЫЙ ЧЗ,ЧСЗ,ГРАНК;
 ВЕЩ  МАССИВ -Б-А-Р[1:12,1:1024];  БАРАБАН -Б-А-Р;
 ВЕЩ  МАССИВ D[1:1024];
 ЦЕЛЫЙ М,ЛО; ВВОД(М,ЛО); ВЫВОД(М,ЛО);
ЧСЗ:=ENTIER(1023/ЛО); ЧЗ:=ENTIER((M-0.5)/ЧСЗ)+1;
ГРАНК:=(М-ЧСЗ×(ЧЗ-1))×ЛО+1;
ГРАН:=N×ENTIER(1023/N); I:=0;
-МЕТЗАП:I:=I+1; ЕСЛИ I>12 ТО СТОП; D[I]:=0;
 ЕСЛИ I=ЧЗ ТО ГРАН:=ГРАНК;
-С-П0176(0016,D[1],D[ГРАН],0,0);
CFIN:=D[ГРАН];
-С-П0177(1037,D[1],D[1024],0,-Б-А-Р[I,1]);
-М1:-К-О-Д(777,7777,7777,7777);
```

```
¯К¯О¯Д(015,СFIN,¯M1,0);
¯К¯О¯Д(076,0,¯МЕТЗАП,0);
НАЧАЛО ЦЕЛЫЙ *НЮ*,L,I,¯N¯3;
ЦЕЛЫЙ ГРАН;
ЦЕЛЫЙ *ЛАМБДА*;
ЦЕЛЫЙ Ю; ЧС3:=ENTIER(1023/M); Ю:=1;
ГРАН:=¯N×ENTIER(1023/¯N);
L:=1; ¯ML: ДЛЯ I:=1,...,M ЦИКЛ НАЧАЛО
*ЛАМБДА*:=L+(I-1)×¯N; ¯N¯3:=ENTIER((*ЛАМБДА*-0.5)/(ГРАН-1))+1;
*НЮ*:=*ЛАМБДА*-(¯N¯3-1)×(ГРАН-1);
D[(Ю-1)×M+I]:=¯Б¯А¯Р[¯N¯3,*НЮ*] КОНЕЦ ;
ЕСЛИ 0<ЧС3-0.5ANDL<¯N-0.5 ТО НАЧАЛО Ю:=Ю+1; L:=L+1; НА ¯ML КОНЕЦ ;
Ю:=Ю×M; ¯С¯П0176(0257,D[1],D[Ю],0,0);
ЕСЛИ L<¯N-0.5 ТО НАЧАЛО Ю:=1; L:=L+1; НА ¯ML КОНЕЦ ;
КОНЕЦ ;
КОНЕЦ *
```

Программа "Вычисление некоторых статистических параметров и коэффициентов корреляции"

Составлена: I вариант – на языке Альфа для ЭВМ типа М-220, М-222 (П3) и II вариант – на языке Алгол для ЭВМ типа БЭСМ-6 (П4)

Назначение.

Первая часть программы предназначена для вычисления параметров, связанных с проверкой гипотезы о нормальной модели распределения методом асимметрии и эксцесса, для чего вычисляются следующие величины: среднее арифметическое ($\bar{x}$), средне-квадратическое (стандартное) отклонение (S), выборочный коэффициет асимметрии (q), оценка нормированного среднего абсолютного отклонения (d), оценка коэффициента эксцесса ($\bar{\gamma}_2$), стандартные отклонения оценок коэффициентов асимметрии (σ_q) и эксцесса ($\sigma_{\bar{\gamma}_2}$) отношения оценок асимметрии (a) и эксцесса ($б$) к их стандартным отклонениям. Кроме того, первая часть программы вычисляет матрицу парных коэффициентов корреляции [1,2,3]. Грубая проверка гипотезы о нормальной модели распределения осуществляется с помощью отношений (а) и (б). Если оба этих отношения по абсолютной величине меньше трех, то гипотеза о нормальной модели распределения принимается. Наиболее точная оценка модели производится с помощью величин (q) и (d) (см. таблица № I).

Вторая часть программы вычисляет матрицы частных и множественных (сводных) коэффициентов корреляции первого порядка.

Краткая математическая справка.

Вычисление параметров и коэффициентов производится по формулам, приведенным в работе В.Н.Бондаренко (1970 г.). Отметим, что коэффициент парной корреляции вычисляется по формуле:

$$r = \frac{\sum (x-\bar{x})\cdot(y-\bar{y})}{(N-1)\cdot(S_x\cdot S_y)}$$

где x и y – признаки, $\bar{x}$ и $\bar{y}$ – среднее арифметическое признаков x и y, N – число наблюдений, S_x и S_y – средне-квадратическое отклонение признаков x и y. Для проверки значимости коэффициентов корреляции приведена таблица № 2.

Таблица № I

Критерии отклонения распределения от нормального
(по Л.Н.Большеву, Н.В.Смирнову, 1968 г.)

N	$q_\alpha = 0.05$ $\alpha = 5\%$	$q_\alpha = 0.01$ $\alpha = 1\%$	$d_\alpha = 0.05$ $\alpha = 5\%$	$d_\alpha = 0.01$ $\alpha = 1\%$
25	0,7II	I,06I	0,736 - 0,868	0,704 - 0,890
30	0,66I	0,982	0,739 - 0,863	0,7I0 - 0,884
35	0,62I	0,92I	0,743 - 0,859	0,7I6 - 0,878
40	0,587	0,869	0,746 - 0,855	0,72I - 0,873
45	0,558	0,825	0,749 - 0,852	0,725 - 0,869
50	0,533	0,787	0,752 - 0,849	0,729 - 0,866
60	0,492	0,723	0,755 - 0,844	0,734 - 0,859
70	0,459	0,673	0,758 - 0,840	0,739 - 0,855
80	0,432	0,63I	0,76I - 0,838	0,743 - 0,852
90	0,409	0,596	0,763 - 0,835	0,746 - 0,848
I00	0,389	0,567	0,764 - 0,834	0,749 - 0,846
I25	0,350	0,508	0,767 - 0,830	0,758 - 0,842
I50	0,32I	0,464	0,770 - 0,827	0,758 - 0,837
I75	0,298	0,430	0,772 - 0,825	0,760 - 0,834
200	0,280	0,403	0,774 - 0,823	0,763 - 0,832
250	0,25I	0,360	0,776 - 0,820	0,766 - 0,829
300	0,230	0,329	0,778 - 0,8I8	0,769 - 0,826
350	0,2I3	0,305	0,779 - 0,8I7	0,77I - 0,824
400	0,200	0,285	0,78I - 0,8I6	0,773 - 0,822
450	0,I88	0,269	0,78I - 0,8I5	0,774 - 0,82I
500	0,I79	0,255	0,782 - 0,8I4	0,776 - 0,820
550	0,I7I	0,243	0,783 - 0,8I3	0,777 - 0,8I9
600	0,I63	0,233	0,784 - 0,8I2	0,778 - 0,8I8
700	0,I5I	0,2I5	0,785 - 0,8II	0,779 - 0,8I6
800	0,I42	0,202	0,786 - 0,8I0	0,780 - 0,8I5
900	0,I34	0,I90	0,786 - 0,809	0,78I - 0,8I4
I000	0,I27	0,I80	0,787 - 0,809	0,782 - 0,8I3

Модель о нормальном распределении: принимается, если коэффициент асимметрии $q \le q_\alpha$ =0,05 и d в пределах d_α=0,05; отвергается, если $q > q_\alpha$ =0,0I или d вне пределов d_α =0,0I, где q - выборочный коэффициент асимметрии, d - оценка нормированного среднего абсолютного отклонения (статистика d).

Таблица № 2.

Критические значения выборочного коэффициент линейной корреляции r_α (по В.Ю.Урбаху, 1964 г.)

№	α =5%	α =1%	№	α =5%	α =1%
4	0,950	0,990	26	0,388	0,496
5	0,878	0,959	27	0,381	0,487
6	0,811	0,917	28	0,374	0,478
7	0,754	0,874	29	0,367	0,470
8	0,707	0,834	30	0,361	0,463
9	0,666	0,798	35	0,332	0,435
10	0,632	0,765	40	0,310	0,407
11	0,602	0,735	45	0,292	0,384
12	0,576	0,708	50	0,277	0,364
13	0,553	0,684	60	0,253	0,333
14	0,532	0,661	70	0,234	0,308
15	0,514	0,641	80	0,219	0,288
16	0,497	0,623	90	0,206	0,272
17	0,482	0,606	100	0,196	0,256
18	0,468	0,590	125	0,175	0,230
19	0,456	0,575	150	0,160	0,210
20	0,444	0,561	200	0,138	0,182
21	0,433	0,549	250	0,124	0,163
22	0,423	0,537	300	0,113	0,148
23	0,413	0,526	400	0,098	0,128
24	0,404	0,515	500	0,088	0,115
25	0,396	0,505	1000	0,062	0,081

r не значим при $r \leq r_\alpha = 0{,}05$ и

r значим при $r > r_\alpha = 0{,}01$

Инструкция к пользованию.

Программа (П3) рассчитана на обработку массива информации объемом $q_{10}=m \cdot N \leq 4000$ знаков (в десятичной системе), где m – число признаков, N – число наблюдений.

Программа (П4) рассчитана на обработку массива информации объемом $q_{10}=m \cdot N \leq 150000$.

Подготовка данных. Массив информации перфорируется по строкам (строка за строкой в десятичной системе). Для программы (П4) числа набиваются через запятую, а в конце массива ставится точка с запятой. Дополнительно к массиву информации пробиваются три информационные карты. На первой информационной карте пробивается число признаков – m в десятичной системе, на второй – число наблюдений – N , на третьей – логический параметр – ЛП, если ЛП=0, то вторая часть программы не работает, если ЛП=I, то работают обе части программы. Для программы (П3) все три информационные карты пробиваются с правильной суммой. Для программы (П4) информационные карты представляют собой три числа, пробитые один за другим через точку с запятой.

Порядок постановки перфокарт для счета.

I. Программа с контрольной суммой.
2. Первая информационная карта.
3. Вторая информационная карта.
4. Третья информационная карта.
5. Массив исходных данных с контрольной суммой.

Для программы (П4) информационные карты и массив исходных данных вставляются в программу между картами № 92 и № 93 .

На печать выдаются следующие величины:

Первая часть программы:

I. Исходная таблица.
2. Среднее арифметическое.
3. Среднеквадратическое отклонение.
4. Оценка коэффициента асимметрии.
5. Оценка нормированного среднего абсолютного отклонения.
6. Оценка коэффициента эксцесса.
7. Стандартные отклонения оценок коэффициентов асимметрии и эксцесса.
8. Отношения оценок асимметрии и эксцесса к их стандартным отклонениям.

9. Полная матрица коэффициентов парной корреляции.

Вторая часть программы.

I0. Коэффициенты множественной корреляции.

II. Коэффициенты частной корреляции.

Контрольный пример.

Исходная таблица (Т).

№	x_1	x_2	x_3	x_4	№	x_1	x_2	x_3	x_4
I	I	I	8	8	I4	3	5	4	6
2	2	3	6	7	I5	4	7	2	5
3	3	5	4	6	I6	5	2	I	4
4	4	7	2	5	I7	6	4	2	2
5	5	2	I	4	I8	5	2	I	4
6	6	4	2	2	I9	2	3	6	7
7	7	6	4	0	20	3	5	4	6
8	4	7	2	5	2I	2	3	6	7
9	5	2	I	4	22	3	5	4	6
I0	6	4	2	I	23	4	7	2	5
II	4	7	2	5	24	5	2	I	4
I2	6	4	2	2	25	4	7	2	5
I3	5	2	I	4	26	4	7	2	5

где N – число наблюдений, x_1, x_2, x_3, x_4 – признаки.

Исходная информация, подготовленная к счету для программы (ПЗ).

Информационная карта № I.

π	КОП	A_I	A_2	A_3
+	0I	4000	–	–
+	0I	4000	–	– к Σ

m

Информационная карта № 2

π	КОП	A_I	A_2	A_3
+	02	2600	–	–
+	02	2600	–	– к Σ

N

Информационная карта № 3

Пробивается, как сказано ранее, в двух вариантах (либо ЛП=0, либо ЛП=I).

π	КОП	A_I	A_2	A_3	
+	00	0000	–	–	ЛП=0
+	00	0000	–	– кΣ	

π	КОП	A_I	A_2	A_3	
+	0I	I000	–	–	ЛП=I
+	0I	I000	–	– кΣ	

Массив исходных данных, перфорирующийся по строкам

π	КОП	A_I	A_2	A_3	
+	0I	I000	–	–	
+	0I	I000	–	–	
+	0I	8000	–	–	
+	0I	8000	–	–	Т
+	0I	2000	–	–	
•	•	•	•	•	
+	0I	2000	–	–	
+	0I	5000	–	–	

Исходная информация, подготовленная к счету для программы (П4).

I) 4;

2) 26;

3) либо 0; либо I;

4) I,I,8,8,2,3,6,...,4,7,2,5;

Результаты счета для программы (П3) и (П4)

I. Исходная матрица данных (Т)

2. Среднее арифметическое ($\bar{x}$):

$\bar{x}_1$ =4,I54; $\bar{x}_2$ =4,346; $\bar{x}_3$ =22,846; $\bar{x}_4$ =4,577

3. Стандартное отклонение (s):

s_1 =I,488, s_2=2,038, s_3=I,9I2, s_4 =I,943

4. Выборочный коэффициент асимметрии (q):

q_1 =-0,I826, q_2= 0,0637, q_3= I,069, q_4=-0,52I.

5. Статистика (d):

$d_1 = 0{,}791$, $d_2 = 0{,}862$, $d_3 = 0{,}820$, $d_4 = 0{,}766$

6. Оценка коэффициента эксцесса ($\bar{\gamma}_4$):

$\bar{\gamma}_4(x_1) = -0{,}76$; $\bar{\gamma}_4(x_2) = -1{,}502$, $\bar{\gamma}_4(x_3) = 0{,}128$, $\bar{\gamma}_4(x_4) = -0{,}319$.

7. Стандартные отклонения оценок коэффициентов асимметрии (σ_q) и эксцесса ($\sigma_{\bar{\gamma}_4}$): $\sigma_q = 0{,}480$, $\sigma_{\bar{\gamma}_4} = 0{,}961$

8. Отношения оценок асимметрии (а) и эксцесса (б) к их стандартным отклонениям:

$a1 = 0{,}380$, $a2 = 0{,}1327$, $a3 = 2{,}226$, $a4 = 1{,}084$

$б1 = 0{,}7917$, $б2 = 1{,}563$, $б3 = 0{,}1328$, $б4 = 0{,}3318$

Судя по коэффициентам (а) и (б) - они все меньше трех по абсолютной величине - модель о нормальном распределении принимается для всех признаков. Но если провести оценку по более точному критерию, т.е. по величинам (q) и (d), то для третьего признака ($q_3 > q_{\alpha = 0.01}$) модель о нормальном распределении отвергается.

9. Полная матрица коэффициентов корреляции (r):

	x_1	x_2	x_3	x_4
x_1	1,000	0,0477	-0,7504	-0,9726
x_2	0,0477	1,000	-0,1397	-0,0524
x_3	-0,7504	-0,1397	1,000	0,5956
x_4	-0,9726	-0,0524	0,5956	1,000

Распечатка коэффициентов корреляции производится поочередно для каждого признака, т.е. сначала для первого признака, затем для второго и т.д.

10. Матрица сводных коэффициентов ($R_{k.l_i}$):

$R_{1.23} = 0{,}7526$, $R_{1.24} = 0{,}9726$, $R_{1.34} = 0{,}9957$

$R_{2.13} = 0{,}1643$, $R_{2.14} = 0{,}5439$, $R_{2.34} = 0{,}1449$

$R_{3.12} = 0{,}7576$, $R_{3.14} = 0{,}9469$, $R_{3.24} = 0{,}6055$

$R_{4.12} = 0{,}9726$, $R_{4.13} = 0{,}9936$, $R_{4.23} = 0{,}5964$

11. Матрица частных коэффициентов корреляции ($r_{kl.i}$):

$r_{32.1} = -0{,}1575$ $r_{42.1} = -0{,}0262$, $r_{43.1} = -0{,}8737$

$r_{31.2} = -0{,}752$, $r_{41.2} = -0{,}9725$, $r_{43.2} = 0{,}595$

$r_{21.3} = -0{,}08738$, $r_{41.3} = -0{,}990$, $r_{42.3.} = 0{,}0387$

$r_{21.4} = -0{,}0144$, $r_{31.4} = -0{,}9164$, $r_{32.4} = -0{,}1353$

Литература

1. Л.Н.Большев, Н.В.Смирнов. "Таблицы математической статистики", М., 1968, 474 с.
2. В.Н.Бондаренко. "Статистическое решение некоторых задач геологии". М., "Недра", 1969, 246 с.
3. В.Ю.Урбах. "Биометрические методы". М., "Наука", 1964, 415 с.

ТЕКСТ ПРОГРАММЫ П 3

```
НАЧАЛО
ЦЕЛЫЙ M,N; ВВОД(M,N);  НАЧАЛО
МАССИВ X[1:N,1:M],X1,S,S1,ГА,D,БЕ,А,ДЕ[1:M],
R[1:M,1:M];
ВЕЩ ЛП;
ЦЕЛЫЙ I,J,L;
ЦЕЛЫЙ K;  ВЕЩ Б1,Б2;  ЦЕЛЫЙ N1;
ВЕЩ СИГ,СИБ;
ВВОД(ЛП,X); N1:=10;
ВЫВОД(X); ВЫВОД(ЛОЖЬ);
ДЛЯ J:=1 ШАГ 1 ДО M ЦИКЛ  НАЧАЛО X1[J]:=0;  ДЛЯ I:=1 ШАГ 1 ДО N ЦИКЛ X1[J]:=X1[J]+X[I,J];

X1[J]:=X1[J]/N;  КОНЕЦ ;
ВЫВОД(X1); ВЫВОД(ЛОЖЬ);
ДЛЯ J:=1 ШАГ 1 ДО M ЦИКЛ  НАЧАЛО S[J]:=0;  ДЛЯ I:=1 ШАГ 1 ДО N ЦИКЛ S[J]:=S[J]+
(X[I,J]-X1[J])↑2;
S[J]:=S[J]/(N-1);
S1[J]:=SQRT(S[J]) КОНЕЦ ;
ВЫВОД(S1); ВЫВОД(ЛОЖЬ);
```

```
ДЛЯ J:=1 ШАГ 1 ДО M ЦИКЛ НАЧАЛО D[J]:=0;
ГА[J]:=0; ДЛЯ I:=1 ШАГ 1 ДО N ЦИКЛ НАЧАЛО
ГА[J]:=ГА[J]+(X[I,J]-X1[J])↑3;
D[J]:=D[J]+ABS(X[I,J]-X1[J]) КОНЕЦ;
ГА[J]:=ГА[J]/(N×S1[J]↑3); D[J]:=D[J]/(N×S1[J]) КОНЕЦ; ВЫВОД(ГА); ВЫВОД(ЛОЖЬ);
ВЫВОД(D); ВЫВОД(ЛОЖЬ);
ДЛЯ J:=1 ШАГ 1 ДО M ЦИКЛ НАЧАЛО БЕ[J]:=0; ДЛЯ I:=1 ШАГ 1 ДО N ЦИКЛ
БЕ[J]:=БЕ[J]+(X[I,J]-X1[J])↑4;
БЕ[J]:=БЕ[J]/(N×S1[J]↑4)-3 КОНЕЦ;
ВЫВОД(БЕ); ВЫВОД(ЛОЖЬ);
СИГ:=SQRT(6/N); СИБ:=SQRT(24/N);
ВЫВОД(СИГ,СИБ); ВЫВОД(ЛОЖЬ);
ДЛЯ J:=1 ШАГ 1 ДО M ЦИКЛ НАЧАЛО A[J]:=ABS(ГА[J]/СИГ); ДЕ[J]:=ABS(БЕ[J]/СИБ);
КОНЕЦ;
ВЫВОД(A,ДЕ); ВЫВОД(ЛОЖЬ);
ДЛЯ I:=1 ШАГ 1 ДО M ЦИКЛ НАЧАЛО ДЛЯ J:=1 ШАГ 1 ДО M ЦИКЛ НАЧАЛО R[I,J]:=0; ДЛЯ L:=1 ШАГ
1 ДО N ЦИКЛ
R[I,J]:=R[I,J]+(X[L,I]-X1[I])×(X[L,J]-X1[J]);
R[I,J]:=R[I,J]/((N-1)×S1[I]×S1[J]);
```

```
КОНЕЦ  КОНЕЦ ;
ВЫВОД(п  ;  ВЫВОД(ЛОЖЬ);
ЕСЛИ ЛП=0 ТО  НА М4;  ДЛЯ К:=1 ШАГ 1 ДО М ЦИКЛ  НАЧАЛО  ДЛЯ L:=1 ШАГ 1 ДО М ЦИКЛ
НАЧАЛО  ДЛЯ I:=1 ШАГ 1 ДО М ЦИКЛ  НАЧАЛО  ЕСЛИ К≠LANDК≠IANDL<I
ТО  НАЧАЛО Б1:=SQRT((R[K,L]↑2+R[K,I]↑2-2*R[K,L]*
R[K,I]*R[L,I])/(1-R[L,I]↑2));
ВЫВОД(K,L,I,Б1);
КОНЕЦ  КОНЕЦ  КОНЕЦ ; ВЫВОД(ЛОЖЬ);  КОНЕЦ ;  ДЛЯ I:=1 ШАГ 1 ДО М ЦИКЛ  НАЧАЛО  ДЛЯ L:=1 ШАГ
1 ДО М ЦИКЛ  НАЧАЛО  ДЛЯ К:=1 ШАГ 1 ДО М ЦИКЛ  НАЧАЛО  ЕСЛИ К>LANDК≠IANDL≠I ТО  НАЧАЛО
Б2:=(R[K,L]-R[K,I]*R[L,I])/SQRT((1-R[K,I]↑2)*
(1-R[L,I]↑2));
ВЫВОД(K,L,I,Б2);
КОНЕЦ  КОНЕЦ  КОНЕЦ ; ВЫВОД(ЛОЖЬ);  КОНЕЦ ; М4:
КОНЕЦ
КОНЕЦ •
```

ТЕКСТ ПРОГРАММЫ П4

```
_BEGIN
_INTEGER M,N;
INPUT(M,N);
_BEGIN
_ARRAY X[1:N,1:M],X1,S,S1,ГА,D,БЕ,A,ДЕ[1:M],
R[1:M,1:M];
_REAL ЛП;
_INTEGER I,J,L;
_INTEGER K;_REAL Б1,Б2;
_INTEGER N1;_REAL _ARRAY МП[1:10];_INTEGER KL;
_REAL СИГ,СИБ;
INPUT(ЛП,X);N1:=10;
OUTPUT('1/','T','ПЕЧАТЬ ДАННЫХ X');
KL:=1;
МЕ1:_FOR I:=1_STEP 1_UNTIL N_DO
_BEGIN _FOR J:=N1×(KL-1)+1_STEP 1_UNTIL N1×KL_DO
_BEGIN _IF J≤M_THEN МП[J-N1×(KL-1)]:=X[I,J]_END ;
OUTPUT('1/','Z+4D.3D',МП);
_END ;
OUTPUT('3/');
_IF N1×KL<M_THEN _BEGIN KL:=KL+1;_GOTO МЕ1_END ;
_FOR J:=1_STEP 1_UNTIL M_DO _BEGIN
X1[J]:=0;_FOR I:=1_STEP 1_UNTIL N_DO X1[J]:=X1[J]+X[I,J];
X1[J]:=X1[J]/N;_END ;
OUTPUT('1/','T','СРЕДНЕЕ АРИФМЕТИЧЕСКОЕ X1');
OUTPUT('1/','Z+4D.3D',X1);
_FOR J:=1_STEP 1_UNTIL M_DO _BEGIN
S[J]:=0;_FOR I:=1_STEP 1_UNTIL N_DO S[J]:=S[J]+
(X[I,J]-X1[J])↑2;
S[J]:=S[J]/(N-1);
S1[J]:=SQRT(S[J])_END ;
OUTPUT('1/','T','СТАНДАРТНОЕ ОТКЛОНЕНИЕ');
OUTPUT('1/','Z+4D.3D',S1);
_FOR J:=1_STEP 1_UNTIL M_DO _BEGIN D[J]:=0;
ГА[J]:=0;_FOR I:=1_STEP 1_UNTIL N_DO _BEGIN
ГА[J]:=ГА[J]+(X[I,J]-X1[J])↑3;
D[J]:=D[J]+ABS(X[I,J]-X1[J])_END ;
ГА[J]:=ГА[J]/(N×S1[J]↑3);D[J]:=D[J]/(N×S1[J])_END ;
```

```
OUTPUT('1/','T','КОЭФФИЦИЕНТ АССИМЕТРИИ');
OUTPUT('1/','Z+4D.3D',ГА);
OUTPUT('1/','T','ВСПОМОГАТЕЛЬНЫЙ ПОКАЗАТЕЛЬ
ЭКСЦЕССА');
OUTPUT('1/','E',D);
_FOR J:=1_STEP 1_UNTIL M_DO _BEGIN
БЕ[J]:=0;_FOR I:=1_STEP 1_UNTIL N_DO
БЕ[J]:=БЕ[J]+(X[I,J]-X1[J])↑4;
БЕ[J]:=БЕ[J]/(N×S1[J]↑4)-3_END ;
OUTPUT('1/','T','БЕТА');
OUTPUT('1/','Z+4D.3D',,БЕ);
СИГ:=SQRT(6/N);СИБ:=SQRT(24/N);
OUTPUT('1/','E',СИГ,СИБ);
_FOR J:=1_STEP 1_UNTIL M_DO _BEGIN
A[J]:=ABS(ГА[J]/СИГ);ДЕ[J]:=ABS(БЕ[J]/СИБ);_END ;
OUTPUT('1/','T','A=','1/','E',A,'1/','T','ДЕ=','1/','E',ДЕ);
_FOR I:=1_STEP 1_UNTIL M_DO _BEGIN
_FOR J:=1_STEP 1_UNTIL M_DO _BEGIN
R[I,J]:=0;_FOR L:=1_STEP 1_UNTIL N_DO
R[I,J]:=R[I,J]+(X[L,I]-X1[I])×(X[L,J]-X1[J]);
R[I,J]:=R[I,J]/((N-1)×S1[I]×S1[J]);
_END _END ;
OUTPUT('1/','T','КОЭФФИЦИЕНТЫ КОРРЕЛЯЦИИ');
KL:=1;
ME2:_FOR I:=1_STEP 1_UNTIL M_DO
_BEGIN _FOR J:=N1×(KL-1)+1_STEP 1_UNTIL N1×KL_DO
_BEGIN _IF J≤M_THEN МП[J-N1×(KL-1)]:=R[I,J]_END ;
OUTPUT('1/','Z+3D.4D',,МП);
_END ;
OUTPUT('3/');
_IF N1×KL<M_THEN _BEGIN KL:=KL+1;_GOTO ME2_END ;
_IF ЛП=0_THEN _GOTO M4;
OUTPUT('1/','T','МНОЖЕСТВЕННЫЙ КОЭФФИЦИЕНТ КОРРЕЛЯЦИИ','1/');
_FOR K:=1_STEP 1_UNTIL M_DO _BEGIN _FOR L:=1_STEP 1_UNTIL M_DO
_BEGIN _FOR I:=1_STEP 1_UNTIL M_DO _BEGIN _IF K≠L∧K≠I∧L<I
_THEN _BEGIN Б1:=SQRT((R[K,L]↑2+R[K,I]↑2-2×R[K,L]×
R[K,I]×R[L,I])/(1-R[L,I]↑2));
OUTPUT('T','P','Z+2D',K,'-','.','Z+2D',L,I,'T','=','Z+4D.3D',Б1,'4B');
```

```
_END ;_END _END _END ;
OUTPUT('1/','T','ЧАСТНЫЙ КОЭФФИЦИЕНТ КОРРЕЛЯЦИИ','1/');
_FOR I:=1_STEP 1_UNTIL M_DO _BEGIN
_FOR L:=1_STEP 1_UNTIL M_DO _BEGIN
_FOR K:=1_STEP 1_UNTIL M_DO _BEGIN
_IF K>L∧K≠I∧L≠I_THEN _BEGIN
Б2:=(R[K,L]-R[K,I]×R[L,I])/SQRT((1-R[K,I]↑2)×
(1-R[L,I]↑2));
OUTPUT('T','R4','Z+2D',K,L,'T',',','Z+2D',I,'T','=','Z+4D.3D',Б2,'4B');
_END _END _END _END ;
M4:
OUTPUT('1/','T','ПРОСЧЕТ ОКОНЧЕН');
_END
_END
```

ПРОГРАММА "КРИТЕРИЙ ОДНОРОДНОСТИ РОДИОНОВА-2"

Программа написана на α -языке для машин, имеющих α -транслятор ТА-IМ.

Сущность критерия Родионова [I] состоит в нахождении границ между двумя однородными совокупностями объектов по результатам измерений одного или нескольких признаков объектов.

Математическое выражение критерия таково:

$$\upsilon(\tau_0^2) = \frac{n-1}{n(n-k)k} \sum_j \frac{\left[(n-k)\sum_{t\in A} x_{tj} - k\sum_{t\in\bar{A}} x_{tj}\right]^2}{\sum_{t\in T} x_{tj}^2 - \frac{1}{n}\left(\sum_{t\in T} x_{tj}\right)^2}$$

где $A \cup \bar{A} = T$

n - количество объектов;

К - принимает значение от I $\div$ n -I;

x_{tj} - результат измерений признаков объектов;

m - количество признаков.

При $\upsilon(\tau_0^2)$ большем табличного при заданном уровне значимости, отвергается гипотеза об однородности пространства, следовательно, граница неоднородности скорее всего соответствует максимальному значению $\upsilon(\tau_0^2)$. Выбрав максимальное значение, мы, тем самым, разбиваем совокупность (пространство Т) на подпространства ξ^{T_α}. Последовательное дробление выделенных подпространств проводится до тех пор, пока все выделенные по максимальным значениям критерия $\upsilon(\tau_0^2)$ пространства ξ^{T_α} не окажутся однородными.

Программа "Критерия Родионова" и составлена для реализации этого алгоритма, то есть для последовательного разбиения совокупности на однородные участки. Однако, число таких участков обычно больше, чем имеется на самом деле. Дальнейшая обработка заключается в устранении ложных границ последовательным попарным сравнением смежных участков по программе Критерий однородности Родионова-I [4].

Для удобства программирования выражение (I) сведено к ряду простых формул.

I. $K = \dfrac{(N-1)}{N\cdot M(N-M)}$;

2. $F_1 = (N-M)\sum_{i=1}^{M} x_{ij}$;

3. $F_2 = M\sum_{i=M+1}^{N} x_{ij}$;

4. $S_1 = \sum_{i=1}^{N} x_{ij}^2$

5. $S_2 = \dfrac{1}{N}\left(\sum_{i=1}^{N} x_{ij}\right)^2$

6. $F = \dfrac{(F_1 - F_2)^2}{S_1 - S_2}$;

7. $$v_m = K\cdot\sum_{j=1}^{J} F;$$

Программа работает следующим образом: величина $v_m\,max$ делит совокупность на две части, для каждой из которых снова находится v_{mi} и максимальные из них, эти совокупности снова разбиваются на две части каждая. Число частей, на которые разбивается вся совокупность в программе ограничено и задается самим программистом, исходя из объема памяти машины. Для этого в программе есть три сменных перфокарты, на которых пробивается число разрешенных разбиений (2^n). Например, 64 или I28, 256, 5I2 и т.д.

Среднее время решения задач $t = 1' \sim 1{,}5'$ для машин типа М-220.

Запись исходной информации

Для выполнения счета по программе необходимо отперфорировать в десятичном коде:

v_1 - массив наблюдений, заданный в виде таблицы с N_1 - число строк и J - числом столбцов. В каждой строке записаны значения признаков для одного объекта, в каждом столбце - значение некоторого свойства для всех объектов.

Перфорация выполняется построчно.

J - - число признаков в объекте,

N1 - - количество объектов.

Постановка перфокарт на ЧУ

Массив перфокарт вводится в следующем порядке:

I) α - схема, контрольная сумма;

2) J - число признаков в объекте, контрольная сумма;

3) N1 - число объектов, контрольная сумма;

4) x - массив наблюдений, контрольная сумма.

Выдача на печать

На печать выдаются следующие значения:

I) $V_{m\,max}$;

2) V_{m_i} - для всех признаков;

3) порядковый номер $p[t_i]\,V_{m\,max}$;

4) число разбиений данного шага.

Программа самовосстанавливается, т.е. имеет (циклический ввод).

Контрольный пример см. стр.

Л и т е р а т у р а

I. Родионов Д.А. Статистические методы разграничения геологических объектов по комплексу признаков. Москва, "Недра", I968.

2. Добрецов Н.Л., Маковская Н.С. Применение вероятностно - статистических методов в геологии (Курс лекций для студентов НГУ), НГУ, Новосибирск, I967.

3. Лакин Г.Ф. "Биометрия", Высшая школа, Москва, I973.

4. Программы к ЭВМ для статистической обработки геологической информации, НГУ, Новосибирск, I972.

КОНТРОЛЬНЫЙ ТЕСТОВЫЙ ПРИМЕР

№№ п/п	SO_2	TiO_2	AI_2O_3	Fe_2O_3	FeO	MnO	MgO	CaO	Na_2O_3	K_2O	P_2O_5
I	74,57	0,24	I4,02	0,96	0,66	0,03	0,70	2,04	4,57	2,0I	0,I0
2	7I,5I	0,37	I5,74	I,06	I,46	0,09	I,02	3,35	3,78	I,67	0,I6
3	72,00	0,27	I4,35	0,66	3,50	0,09	0,85	2,6I	3,80	I,70	0,00
4	70,87	0,37	I5,47	I,06	I,46	0,09	I,02	3,35	3,78	I,67	0,I6
5	70,89	0,36	I4,65	I,04	I,98	0,07	I,05	2,03	3,98	3,42	0,07
6	69,48	0,37	I5,00	I,2I	I,93	0,09	I,22	2,68	4,I6	2,90	0,I0
7	68,24	0,36	I5,I4	I,59	I,59	0,06	I,84	3,3I	3,69	3,23	0,09
8	67,73	0,38	I6,28	I,33	I,60	0,03	I,06	2,20	4,40	3,20	0,I2
9	69,I2	0,38	I4,50	I,48	I,84	0,08	I,49	3,48	3,9I	3,08	0,I8
I0	68,07	0,02	I5,57	0,43	0,95	0,02	0,38	2,04	6,88	2,90	0,03
II	69,94	0,08	I6,07	0,57	0,90	0,03	0,55	2,I6	5,40	3,40	0,05
I2	73,07	0,I5	I3,92	0,94	I,I4	0,06	0,6I	I,66	3,99	4,64	0,07
I3	74,75	0,I3	I3,26	0,49	I,60	0,05	0,50	0,7I	3,97	4,20	0,03
I4	70,75	0,40	I4,I9	I,20	I,78	0,07	I,I7	2,54	3,84	3,05	0,05
I5	75,57	0,05	I2,76	0,20	I,I5	0,03	0,57	0,6I	3,85	4,84	0,06
I6	75,66	0,II	I2,62	0,57	0,98	0,04	0,55	0,79	3,64	4,72	0,02
I7	76,3I	0,26	II,97	0,70	I,64	0,02	0,27	0,55	2,87	5,02	0,02
I8	75,80	0,I8	I2,65	0,64	0,54	0,05	0,23	0,99	3,83	4,33	0,I0
I9	73,I4	0,26	I3,32	0,66	I,75	0,06	0,90	I,44	4,05	3,6I	0,04

Выборка из I9 анализов проверялась на однородность. В результате она была разбита на I2 участков, условно однородных. (Для установления действительных границ результат должен быть подвергнут дополнительному анализу). В результате счета получено:

-.--------	-0-	-.--------	-0-	+.460638090	-00
+.I90000000	+02	+.II0000000	+02	+.442766282	+0I
-.--------	-0-	-.--------	-0-	+.220829562	+0I
+.7I6II6698	+02	+.567378009	+0I	+.836390I02	+00
-.--------	-0-	+.2448I748I	+0I	+.269230769	+0I
+.900000000	+0I	+.679364262	+0I	+.I275823I9	+0I
-.--------	-0-	+.650826446	+00	+.23I838532	+0I
+.760672246	+0I	+.I4074452I	+0I	+.I06I356II	+0I
+.540777627	+0I	+.259430604	+0I	+.I2I742695	+0I
+.750630540	+0I	+.3656I2070	-00	+.454634745	-00
+.876642920	+0I	+.270409545	+0I	-.--------	-0-
+.3263I7283	+0I	+.73II36930	+0I	+.400000000	+0I
+.55I8598I3	+0I	+.3536243I4	+0I	-.--------	-0-
+.260077557	+0I	+.373456790	-00	+.I9362I93I	+02
+.I0II37220	+02	-.--------	-0-	-.--------	-0-
+.95I576746	-00	+.200000000	+0I	+.300000000	+0I
+.845527344	+0I	-.--------	-0-	-.--------	-0-
+.302I3I766	+0I	+.24260724I	+02	+.I362I9248	+0I
-.--------	-0-	-.--------	-0-	+.200000000	+0I
+.I00000000	+0I	+.400000000	+0I	+.200000000	+0I
-.--------	-0-	-.--------	-0-	+.200000000	+0I
+.247585370	+02	+.42986I5I5	+0I	+.200000000	+0I
-.--------	-0-	+.I59380069	+0I	+.000000000	-00
+.I00000000	+0I	+.I407I5238	-02	+.200000000	+0I
-.--------	-0-	+.355687205	+0I	+.200000000	+0I
+.2I79I9905	+0I	+.5I774594I	+00	+.200000I07	+0I
+.258298755	+0I	+.236250000	+0I	+.I9999995I	+0I
+.2I8237496	+0I	+.246669893	+0I	+.200000000	+0I
+.329349743	+00	+.758472822	+00	-.--------	-0-
+.248003I63	+0I	+.I93I58I38	+0I	+.I48358765	+02
+.288000000	+0I	+.675759682	+0I	-.--------	-0-
+.I844485I6	+0I	+.I54269972	-0I	+.600000000	+0I
-.I7II25258	+0I	-.--------	-0-	-.--------	-0-
+.378628257	+0I	+.20032I078	+02	+.26703I680	+0I
+.5532965I8	+00	-.--------	-0-	+.26703I680	+0I

+.292772I86 -0I		+.I40000000 +02	Уш.	+.833333324 +00
-.--------- -0-		-.--------- -0-		+.566574404 +00
+.338589520 +02		+.3079I872I +0I		+.297I0I449 +0I
+.276923840 +0I		+.3I8339547 +0I		+.I60000000 +02
+.I23270440 +0I		+.7I428557I -0I		-.--------- -0-
+.II522520I +0I		-.--------- -0-		+.I7I8I8458 +02
+.II82I5895 +0I	IУш.	+.800000000 +0I		-.--------- -0-
+.82I573289 -0I		-.--------- -0-		+.I60000000 +02
+.324499726 +0I		+.II2786772 +02		-.--------- -0-
+.I37288I35 +0I		-.-------- -0I		+.I975I5949 +0I
-.--------- -0-		+.800000000 +0I		+.I846I5384 +0I
+.I68I74343 +02		-.--------- -0-		+.I94485085 +0I
-.--------- -0-		+.I73697I37 +0I		+.9829I2332 +00
+.I30000000 +02		+.499999994 -00		+.I87695I46 +0I
-.--------- -0-		+.I20067247 +0I		+.I50000000 +0I
+.I650I8509 +0I		+.I56555773 -0I		+.I99289099 +0I
+.I99II6347 +0I		+.I997504I5 +0I		+.96I53846I +00
+.I04849278 +0I		+.I28947368 +0I		+.I9I7I779I +0I
+.I2I53I903 +0I		+.698842543 -02		+.I6842I049 +0I
+.I02876377 +0I		+.725I5003I +00		+.499999999 -00
+.I50000000 +0I		+.9I9757760 -0I		-.--------- -0-
+.I953I37I0 +0I		+.I92857I49 +0I	УIш.	+.320000000 +02
+.I46I27979 +0I		+.I7857I428 +0I		-.--------- -0-
+.I96984922 +0I		-.--------- -0-		
+.I85638692 +0I		+.I3099003I +02		
+.II42857I4 +0I		-.--------- -0-		
-.--------- -0-		+.I70000000 +02		
+.2I0355740 +02		-.--------- -0-		
-.--------- -0-		+.I49527I54 -0I		
+.I80000000 +02		+.I4634I463 -00		
-.--------- -0-		+.2335236I3 -00		
+.378644975 +0I		+.3342I3566 -00		
+.II2952I58 +0I		+.I86368584 +0I		
+.233037305 +0I		+.I80000000 +0I		
+.3394I5I76 -00		+.I26I58590 +0I		
+.I47392756 +0I		+.2206I0687 +0I		
+.200000000 +0I		+.499667384 -00		
+.26749454I +0I		+.2466I9843 +0I		
+.3085I9553 +0I		+.227272727 +0I		
+.96092I892 +00		-.--------- -0-		

Текст программы (Критерий Родионова-2)

```
{целые T1,T,t1, i, n,t,n1, N, N1 ,M,к, J , j ,e ; вещественные K, v2,У,У1; M:ввод ( J , N1 );{целый
массив L1, L[1 :2,1: 2^n], p[1 : 2^n] ; массив x, F3 [1 : J ,1: N1], v1 [1 : N1, 1 : J], v [1 : N1] ,
S01, S02, S1 ,S2, S0, S , F1 ,F 2, F [1: J]; ввод (v1); вывод ( N1); T1 :=T:= 1; L[1,1] := 1 ; L[2,
1]:=N1;α1: для t1 := 1,..., T1 цикл{ N := L[2, t1] - L[1, t1]+ 1 ; если N≤ 2 то { p [t1]:=0; на α};
для n1:= L[1, t1] ,...,L[2,t1] цикл для j := 1,..., J цикл x [ j, n1 - L[1, t1] + 1] := v1[n1, j ] ;
для к:= 1 ,...,N - 1 цикл {M:=к;K:=(N - 1 )/(N x M x (N - M)); v2:=0; для j := 1 ,..., J цикл{S01[j]
:=0; S02 [j] :=0; s1[j] :=0; S 2 [j] :=0; для n := 1,..., M цикл S01 [j]:= S01[j] +x [j , n];
F1[j]:= S01 [j]× (N -M); для n :=M+ 1 ,...,N цикл S02 [j] := S02 [j] +x [j , n] ; F2 [j] := S02[j] xM;
S[j] :=(F1[j] - F2[j] )↑2; для n := 1,...,N цикл S1 [j] := S1 [j] +x[j , n]↑2 S2[j] := S2[j]
+x [j , n] } ; S0[j] := S1 [j]- S2 [j]↑2/N ; если S0[j] =0 то F[j]:=0 иначе F [j] := S[j]/S0[j] ; v2
:= v2+ F[j]; F3 [j , к] := F [j]× K } ; v [ к]:=K ×v2 } ; y:=0; для к:= 1 ,..., N - 1 цикл {y1 :=v[к] ;
если {y1>y то {y:=y1 ; p[t1]:=к+ L [1,t1] - 1; для j := 1 ,..., J цикл F[j]:= F3 [j , к ]}} ; e:=p
[t1]; вывод (y,e,F );α:} ; T:=T1× 2; если T >2^n то на α3; t1:=1 ; для t := 1 шаг 2 до T - 1 цикл
{L1[1,t ] := L[1, t1]; L1[2,t] :=p [t1]; если L[2, t1] =p [t1] то L1[2,t] :=p [t1]- 1 ; если p [t1]=0
то {L1[1, t]:=0; L1[2,t] :=0} ; t1 := t1+ 1}; вывод (T1) ; t1:= 1; для t :=2 шаг 2 до T цикл
{L1[1 , t] :=p [t1] + 1 ; L1[2,t] := L [2, t1] ; если p[t1]=0 то { L1[1 ,t] :=0 ; L1[2,t]
:=0} ; t1 := t1+ 1} ; T1 :=T ; для i := 1,..., 2 цикл для t1:= 1 ,..., T1 цикл L[i, t1] := L1[i,
t1] ; для t1 := 1 ,..., T1 цикл{N := L[2,t1] - L[1, t1] + 1 ; если N > 2 то на α1;α3; }} ; на M}*
```

КОМПЛЕКС АЛГОРИТМОВ ДЛЯ ПРОГРАММ РАСПОЗНАВАНИЯ ПО Т-СВОЙСТВАМ

В работе [I] для произвольной бинарной таблицы Т ($m \times n$) определено понятие Т-свойства. Если фиксирован класс Т-свойств $\mathcal{U}$, то среди его членов выделяются тупиковые относительно $\mathcal{U}$ Т-свойства, составляющие подкласс $\tau(\mathcal{U})$ класса $\mathcal{U}$. Подкласс $\tau(\mathcal{U})$ порождает заданную на n-мерном единичном кубе E_n числовую меру $B(S)$, принимающую значения на отрезке [0,I] и равную единице для всех строк таблицы Т. На основе $B(S)$ может быть сформулирована следующая процедура распознавания [I]. Пусть $T = T_I$ - таблица эталонов первого класса. Для строк S таблицы эталонов T_2 второго класса вычислим $\max B(S) = a$. Предположим, что $a < 1$ и $\varepsilon_1, \varepsilon_2 > 0$ таковы, что $\varepsilon_1 + \varepsilon_2 \leq 1 - a$. Тогда для пробы S, подлежащей распознаванию

I) если $B(S) \geq 1 - \varepsilon_1$, то S относится к первому классу;
2) если $B(S) \leq a + \varepsilon_2$, то S относится ко второму классу;
3) если $a + \varepsilon_2 < B(S) < 1 - \varepsilon_1$, то S не распознается.

В данной работе (для понимания которой не требуется знакомства с [I]) представлены алгоритмы, дающие частные реализации этой схемы распознавания для фиксированных классов Т-свойств. Для этих алгоритмов ведется разработка комплекса программ. В работе дается описание получаемых с помощью этих программ результатов. Теоретические основы подхода см. в [I], кроме того, в ближайшее время должна выйти еще одна публикация на эту тему. И если в настоящей статье упор сделан на процесс вычисления величин, необходимых для распознавания, то там основное внимание уделяется непосредственно процессу распознавания.

I. ОСНОВНЫЕ ПОНЯТИЯ

Определение I. Набор столбцов $t = (x_{j_1}, \ldots, x_{j_k})$ таблицы Т называется Т-набором (узловой системой), если таблица

T^t, полученная из T удалением всех столбцов, кроме x_{j_1}, ..., x_{j_k}, не содержит в качестве строки по крайней мере одну вершину k-мерного единичного куба E_k.

Каждой узловой системе можно сопоставить некоторую булеву функцию переменных $x_{j_1}, \ldots, x_{j_k}$ $z(x_{j_1}, \ldots, x_{j_k})$, называемую узким Т-свойством, по следующему правилу: $z(\tilde{\sigma}) = 1$, где $\tilde{\sigma} = (\sigma_1, \ldots, \sigma_k)$, в том и только в том случае, когда существует строка S_i таблицы Т, $S_i = (t_{i1}, t_{i2}, \ldots, t_{in})$, такая, что $t_{ij_1} = \sigma_1, \ldots, t_{ij_k} = \sigma_k$. Всякое узкое Т-свойство является Т-свойством в смысле [I]. Каждому узкому Т-свойству отвечает единственный Т-набор и наоборот.

Для набора столбцов t обозначим через T^t таблицу, полученную удалением из T всех столбцов, не входящих в t.

<u>Определение 2</u>. Т-набор $t = (x_{j_1}, \ldots, x_{j_k})$ называется тупиковым, если существует вершина $\tilde{\sigma}$ куба E_k, не входящая в число строк T^t, такая, что t является тупиковым тестором [2] для T^t и $\tilde{\sigma}$.

<u>Утверждение I.</u> Т-набор t является тупиковым в том и только том случае, когда отвечающее ему узкое Т-свойство является тупиковым (в смысле [I]) в классе всех узких Т-свойств.

Доказательство этого утверждения будет опубликовано отдельно. В дальнейшем ограничимся рассмотрением только узких Т-свойств, причем, учитывая сказанное выше и соображения удобства, изложение будем вести в терминах Т-наборов.

<u>Пример I</u>.

x_1	x_2	x_3	x_4
I	0	0	I
0	I	I	0
I	I	0	0
0	0	I	0

Здесь (x_1, x_3), (x_2, x_3, x_4) – тупиковые Т-наборы, (x_1, x_2, x_3) – не тупиковый

<u>Определение 3</u>. Узловая система t называется узлом, если никакой поднабор t не является узловой системой.

Ясно, что каждый узел является тупиковым Т-набором. Кроме того, для длины узла $\ell(t)$ справедлива оценка

$$1 \le \ell(t) \le [\log_2 m] + 1$$

В дальнейшем предполагается, что заданы:

I) таблица $T = T_I$ $(m_1 \times n)$ эталонов первого класса;
2) таблица T_2, состоящая из трех подтаблиц : $T_2^{II}(m_1 \times n)$, $T_2^{экз}$ $(m_3 \times n)$, $T_2^{пр}(m_4 \times n)$, где T_2^{II} - таблица эталонов 2-го класса, $T_2^{экз}$ - таблица экзамена и $T_2^{пр}$ - таблица проб, подлежащих распознаванию; $m_2 + m_3 + m_4 = \bar{m}$.

<u>Определение 4</u>. Т-набор t голосует за строку $S = (t_1, \ldots, t_n)$ (вершину куба E_n), если $S^t = (t_{j_1}, \ldots, t_{j_k})$ совпадает с одной из строк T^t.

<u>Определение 5</u>. Т-набор t называется сквозным, если он голосует за все строки T_2^{II}.

<u>Определение 6.</u> Несквозной набор t называется эффективным, если существует строка S подтаблицы T_2^{II} такая, что t не голосует за S и любой подна-бор $t' \subset t$ длины k-I голосует за S.

П. АЛГОРИТМ "Т-НАБОРЫ"

<u>I. Назначение алгоритма</u>

Требуется: I) представить результаты голосования тупиковых T_I-наборов по всем строкам таблицы T_2 в виде следующей матрицы (логической матрицы):

Здесь I обозначает, что набор t голосует за строку, 0 - что не голосует.

2) Вычислить общее число K эффективных и сквозных тупиковых Т-наборов, число эффективных $K_э$ и сквозных K_c Т-наборов, $K = K_э + K_c$.
3) Для каждой строки T_2 вычислить отношение числа эффективных голосов за нее к $K_э$ (вес строки), равное $B(S)$.
4) Для каждого столбца x_j исходной таблицы $T = T_I$ вычислить частоту P_j его вхождения в эффективные тупиковые Т-наборы (важность признака).
5) Перечислить все сквозные и эффективные наборы.

Для таблиц большого объема (I) и (5) могут быть опущены.

В качестве класса U здесь выступает совокупность всех узких Т-свойств, отвечающих тупиковым эффективным Т-наборам, а в качестве $B(S)$ - отношение (3). Свойства, отвечающие сквозным наборам, образуют "критерий общности" [3] для допуска пробы к распознаванию.

Если необходимо построить все тупиковые Т-наборы*) и получить по ним результаты голосования, то следует в качестве $T_2^П$ взять любую строку T_I. Тогда все тупиковые Т-наборы окажутся сквозными. Таким образом, описанный алгоритм применим и для класса всех узких Т-свойств.

2. Процедура

Нетрудно показать, что для длины тупиковых Т-наборов $\ell(t)$ справедлива оценка $1 \le \ell(t) \le m$. Основную идею описываемой процедуры можно проиллюстрировать следующими очевидными соображениями. Если t -тупиковый Т-набор и $\ell(t) = m$, то каждый столбец t содержит либо m-I нулей, либо m-I единиц. Поэтому искать такой набор t следует только на множестве столбцов этого вида. Проверку на тупиковость можно производить так: если выделенные (т.е. встречающиеся один раз в своих столбцах) элементы лежат в разных строках, то набор t является тупиковым.

Пример I. Выделенные элементы обведены:

---

*) а не только сквозные и эффективные.

$$T^{t_1}=\begin{array}{c} \begin{array}{ccc} x_{j_1} & x_{j_2} & x_{j_3} \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \quad T^{t_2}=\begin{array}{c} \begin{array}{ccc} x_{j_1} & x_{j_2} & x_{j_3} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \quad T^{t_3}=\begin{array}{c} \begin{array}{ccc} x_{j_1} & x_{j_2} & x_{j_3} \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

Здесь t_1, t_2 – тупиковые, t_3 – не тупиковый.

Описание процедуры

I. Найти отождествляющие признаки, проголосовать по ним, разбраковать их на сквозные и эффективные. Из таблицы вывести и при поиске остальных тупиковых Т-наборов (т. Т-н) не рассматривать.

II. Поочередно в лексикографическом порядке перебираем все наборы длины $k = 2, 3, \ldots$ и т.д. вплоть до $\min(n-1, m-1)$ включительно*).

При $k = 2$ проверяем, является ли t узловой системой. Если да, то t -узел (так как отождествляющие столбцы удалены), поэтому если он не голосует за s из T_2^{II}, то он эффективный т.Т-н. Если t голосует за все строки T_2^{II}, он сквозной т.Т-н.

III. Пусть набор $t = (x_{j_1}, \ldots, x_{j_k})$, $3 \le k \le m-1$, имеет вид:

$$\begin{array}{cccc} x_{j_1} & x_{j_2} & \cdots & x_{j_k} \\ \hline t_{1j_1} & t_{1j_2} & \cdots & t_{1j_k} \\ t_{2j_1} & t_{2j_2} & \cdots & t_{2j_k} \\ \cdots & \cdots & \cdots & \cdots \\ t_{zj_1} & t_{zj_2} & \cdots & t_{zj_k} \\ \cdots & \cdots & \cdots & \cdots \\ t_{mj_1} & t_{mj_2} & \cdots & t_{mj_k} \end{array}$$

Удалим из него все повторения совпадающих строк, оставив по одному представителю. Для оставшейся таблицы $t^{\text{ост}}$ перебираем все сочетания строк длины k.

*) Можно вести перебор не обязательно до $\min(n-1, m-1)$, а до заданного числа $q \le m-1$. Следует подчеркнуть также, что сначала перебираются все наборы длины 2, затем все наборы длины 3 и т.д. При заданном ограничении длины q схема распознавания та же, что и выше.

Пусть сочетание имеет вид

$$\boxed{\begin{matrix} t_{11} & t_{12} & \dots & t_{1k} \\ t_{21} & t_{22} & \dots & t_{2k} \\ \dots & \dots & \dots & \dots \\ t_{k1} & t_{k2} & \dots & t_{kk} \end{matrix}}$$

Если в каждом его столбце либо к-I нуль, либо к-I единица, такое сочетание рассматривается. Если нет, сочетание пропускается и переходим к следующему сочетанию. Для рассматриваемого сочетания находим выделенные элементы столбцов (встречающиеся I раз). Если выделенные элементы лежат в разных строках*), то составим "реплику" сочетания - строку $\tilde{t} = (t_1, \dots, t_k)$ из невыделенных (повторяющихся) элементов. Сравним её с невошедшими в сочетание строками $t^{\text{ост}}$. Если $\tilde{t}$ совпадает с одной из строк, то переходим к следующему сочетанию. Если нет, то набор t - тупиковый. Дальше сочетания не просматриваются, переходим к голосованию и разбраковке набора на сквозной, эффективный или пустой. Сквозные и эффективные запоминаются. Если $\tilde{t}$ совпадает с одной из строк $t^{\text{ост}}$ при всех сочетаниях строк по к, то набор t - пустой и переходим к следующему набору.

Если т.Т-н не сквозной, то проверка на эффективность производится путем голосования строк T_2^{Π} по всем поднадобо - рам $t' \subset t$, $\ell(t') = k-1$

IУ. Найдем указанным выше способом все сквозные и эффективные тупиковые T_I-наборы длины m, просматривая только столбцы, имеющие m-I единиц или m-I нулей. Проведем по ним голосование. При $m \geq n$ этап IУ пропускается.

3. Пример

$$T_I = \begin{pmatrix} I & 0 & 0 & I \\ 0 & I & I & 0 \\ I & I & 0 & 0 \\ 0 & 0 & I & 0 \end{pmatrix}$$

Здесь $n = 4$, $m_1 = 4$, $m_2 = I$, $m_3 = 2$, $m_4 = I$. Обозначим строки T_2 через S_1^2, S_2^2, S_3^2, S_4^2. Через α_j обозначим двоичный набор, отвечающий набору t_j. Проведем построение по описанной выше схеме и дадим образец окончательного результата счета.

*) или, что то же самое, в каждой строке сочетания имеется выделенный элемент.

$$T_2 = \begin{array}{l} \left[\begin{array}{cccc} I & 0 & I & 0 \\ \hline I & I & 0 & I \\ 0 & 0 & 0 & 0 \\ \hline I & I & I & I \end{array}\right] \end{array} \begin{array}{l} \\ \} \text{ П кл} \\ \} \text{ I кл} \\ \\ \end{array}$$

I отождествляющих признаков нет;

П к = 2:

I) $\alpha_1 = (I\ I\ 0\ 0)$ – не узловая система (пустой набор);

2) $\alpha_2 = (I\ 0\ I\ 0)$ – узел, эффективный, не голосует за S_1^2, S_3^2, S_4^2;

3) $\alpha_3 = (I\ 0\ 0\ I)$ – узел, сквозной;

4) $\alpha_4 = (0\ I\ I\ 0)$ – пустой;

5) $\alpha_5 = (0\ I\ 0\ I)$ – узел, сквозной, не голосует за S_2^2, S_4^2;

6) $\alpha_6 = (0\ 0\ I\ I)$ – узел, сквозной, не голосует за S_2^2, S_4^2;

Ш. $\kappa > 2$

7) $\alpha_7 = (I\ I\ I\ 0)$:

$$T^{t_7} = \left[\begin{array}{c|ccc} I & I & 0 & 0 \\ 2 & 0 & I & I \\ 3 & I & I & 0 \\ 4 & 0 & 0 & I \end{array}\right], \quad t_7^{ост} = t_7, \quad \begin{array}{c|ccc} I & I & 0 & 0 \\ 2 & 0 & I & I \\ 3 & I & I & 0 \end{array}, \quad \begin{array}{c|ccc} I & I & 0 & 0 \\ 2 & 0 & I & I \\ 4 & 0 & 0 & I \end{array}, \quad \begin{array}{c|ccc} I & I & 0 & 0 \\ 3 & I & I & 0 \\ 4 & 0 & 0 & I \end{array},$$

$$\begin{array}{c|ccc} 2 & 0 & I & I \\ 3 & I & I & 0 \\ 4 & 0 & 0 & I \end{array}$$

– набор пустой;

8) $\alpha_8 = (I\ I\ 0\ I)$:

$$T^{t_8} = \left[\begin{array}{c|ccc} I & I & 0 & I \\ 2 & 0 & I & 0 \\ 3 & I & I & 0 \\ 4 & 0 & 0 & 0 \end{array}\right], \quad t_8 = t_8^{ост},$$

$$\begin{array}{c|ccc} I & I & 0 & I \\ 2 & 0 & I & 0 \\ 3 & I & I & 0 \end{array} \qquad \begin{array}{c|ccc} I & I & 0 & I \\ 2 & 0 & I & 0 \\ 4 & 0 & 0 & 0 \end{array} \qquad \begin{array}{c|ccc} I & I & 0 & I \\ 3 & I & I & 0 \\ 4 & 0 & 0 & 0 \end{array} \quad \tilde{t} = (I,0,0)$$

не входит в $t^{ост}$, t^8 – т.Т-н, эффективный, не голосует за S_1^2, S_2^2, S_4^2;

9) $\alpha_9 = (I\ 0\ I\ I)$:

$$T^{t_9} = \left[\begin{array}{c|ccc} I & I & 0 & I \\ 2 & 0 & I & 0 \\ 3 & I & 0 & 0 \\ 4 & 0 & I & 0 \end{array}\right], \quad t_9^{ост} = \left[\begin{array}{c|ccc} I & I & 0 & I \\ 2 & 0 & I & 0 \\ 3 & I & 0 & 0 \end{array}\right]$$

– набор пустой;

I0) $\alpha_{10} = (0\ I\ I\ I)$: $T^{t_{10}} = \left[\begin{array}{c|ccc} I & 0 & 0 & I \\ 2 & I & I & 0 \\ 3 & I & 0 & 0 \\ 4 & 0 & I & 0 \end{array}\right]$, $t_{10}^{ост} = t_{10}$,

$\begin{array}{c|ccc} I & 0 & 0 & I \\ 2 & I & I & 0 \\ 3 & I & 0 & 0 \end{array}$, $\begin{array}{c|ccc} I & 0 & 0 & I \\ 2 & I & I & 0 \\ 4 & 0 & I & 0 \end{array}$, $\begin{array}{c|ccc} I & 0 & 0 & I \\ 3 & I & 0 & 0 \\ 4 & 0 & I & 0 \end{array}$, $\tilde{t} = (0,0,0)$

не входит в $t_{10}^{ост}$. t_{10} – т.Т-н, сквозной, не голосует за S_2^2, S_3^2, S_4^2.

IУ. Этап пропускается ($m \geq n$).

Логическая матрица

	2	3	4	6	I	5
эталон П кл	I	I	I	I	0	0
экзамен П кл	I	0	0	0	I	0
экзамен I кл	I	I	I	0	0	I
проба	I	0	0	0	0	0

веса строк 0, I/2, I/2, 0;
важность исходных признаков I , I/2, I/2, I/2;
$K_c = 4$, $K_э = 2$, $K = 4 + 2 = 6$;
сквозные наборы: (x_1, x_4), (x_2, x_4), (x_3, x_4), (x_2, x_3, x_4);
эффективные наборы (x_1, x_3), (x_1, x_2, x_4);
общий список построенных наборов I) (x_1, x_3); 2) (x_1, x_4); 3) (x_2, x_4); 4) (x_3, x_4); 5) (x_1, x_2, x_4); 6) (x_2, x_3, x_4).

Ш. АЛГОРИТМ "УЗЛЫ"

I. Назначение алгоритма

То же, что у алгоритма "Т-наборы", но применительно к совокупности узких Т-свойств, отвечающих узлам бинарной таблицы T_I.

2. Процедура

При её разработке использованы некоторые особенности алгоритма Мацака для построения тупиковых тестов [4].

Описание процедуры

I. Вычисляется $\sigma = [\log_2 m] + 1$, где $m = m_1$.

2. Построение узлов. Рассматриваются всевозможные n -разрядные двоичные числа $\alpha = \alpha_1, \ldots, \alpha_n, \alpha_i = 0,1$ такие, что $\alpha_1 + \alpha_2 + \ldots + \alpha_n \leq \sigma$. Такие числа назовем допустимыми. Каждому числу сопоставляется набор столбцов, составленный из тех столбцов, номера которых совпадают с номерами разрядов числа $\alpha = \alpha_1 \ldots \alpha_n$, равных I.

Очевидно, что если к числу $00\ldots0I$ последовательно прибавлять по единице, пока не дойдем до числа $\overbrace{I\ldots I}^{\sigma}0\ldots0$, то можно получить все n -разрядные двоичные числа $\alpha = \alpha_1 \ldots \alpha_n$, для которых $\alpha_1 + \alpha_2 + \ldots + \alpha_n \leq \sigma$ и соответствующие им наборы столбцов имеют длину не большую σ.

Пусть $t = (t_{k_1}, \ldots, t_{k_\ell})$ – допустимый набор. Если $\ell = \sigma$, то t заведомо является узловой системой. Если $\ell < \sigma$, то проверяем, является ли t узловой системой.

<u>Случай (а)</u>. t является узловой системой. Ему отвечает двоичное число, у которого внизу указаны номера разрядов числа α:

$$\alpha = 0\ldots0\underset{k_\ell}{I}0\ldots0\underset{k_2}{I}0\ldots0\underset{k_1}{I}0\ldots0$$

Проверяется, является ли t узлом. Для этого поочередно удаляется каждый столбец и проверяется, будет ли узловой полученная система*). Если t -узел, он запоминается и по нему проводится голосование строк T_2. Независимо от того, окажется t узлом или нет, для получения следующего набора t' прибавим к α единицу сразу в k_1 -разряде. Легко видеть, что полученный набор t' также является допустимым. Если α имеет вид $\overbrace{I\,I\ldots I}^{\ell}0\ldots0$, то построение закончено.

<u>Случай (б)</u>. t не является узловой системой. Переходим к следующему набору t'. Для этого прибавим к α единицу, $\alpha' = \alpha + 1$. В качестве t' берем набор, отвечающий α'. t' также будет допустимым набором, так как в случае (б) $\alpha < \sigma$ и, значит, $\ell' \leq \ell + 1 \leq \sigma$. Если α имеет вид

$\overbrace{I\,I\ldots I}^{\sigma}0\ldots0$, то построение закончено.

---

*) При $\ell = I$ в случае (а) t очевидным образом является узлом.

Очевидно, что при таком способе перебора будут найдены все узлы. При этом просматриваются <u>только допустимые</u> наборы.

3. Пример

$$T_I = \begin{bmatrix} I & 0 & I \\ 0 & I & 0 \\ 0 & 0 & 0 \\ I & I & 0 \end{bmatrix} \qquad T_2 = \begin{bmatrix} 0 & 0 & I \\ 0 & I & I \\ I & 0 & 0 \\ I & I & I \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}0\\0\end{matrix}}\right\} \text{II кл} \\ \left.\vphantom{\begin{matrix}0\\0\end{matrix}}\right\} \text{I кл} \end{matrix}$$

Здесь $m_2 = I$, $m_3 = 2$, $m_4 = I$

Логическая матрица

2	I
I	0
0	0
I	I
0	I

веса строк : 0, 0, I, I;
важность исходных признаков I, 0, I ;
$K_c = I$, $K_3 = I$, $K = 2$;
сквозные наборы: (x_2, x_3) ;
эффективные наборы: (x_1, x_3) ;
общий список построенных наборов I) (x_1, x_3), 2) (x_2, x_3).

IУ. ПЕРЕХОД К МНОГОЗНАЧНЫМ ПРИЗНАКАМ

Описываемые алгоритмы без каких-либо существенных изменений могут применяться и для анализа данных, представленных k-значными признаками при $k \geq 3$. Полное изложение данного вопроса потребовало бы привлечения аппарата функций k-значной логики [5] и теории измерений [6], что выходит за рамки настоящей статьи. Оставляя это на будущее, ограничимся рассмотрением только процедурных вопросов.

Пусть признаки $x_1, \ldots, x_j, \ldots, x_n$ принимают значения на множествах $D_1, \ldots, D_j, \ldots, D_n$ соответственно, где $D_j = \{0, 1, \ldots, R_j - 1\}$ и $R_j = 2, 3, 4\ldots$. С каждым набором столбцов $t = (x_{j_1}, \ldots, x_{j_e})$ свяжем класс функций k-значной логики $\Omega(j_1,\dots,j_e)$, заданных на декартовом произведении $D(t) = D_{j_1} \times D_{j_2} \times \dots \times D_{j_e}$ и принимающих значения на множестве $\{0,I\}$ *). Объединение всех построенных таким образом множеств обозначим через Ω.

Заменяя в определении Т-свойства [I], в определении узкого Т-свойства и определениях I–6 термин "булева функция" на "функция класса Ω", k-мерный единичный куб E_k – на $D(t)$, приходим к соответствующим определениям для многозначных признаков. Столь же естественным образом на многозначный случай распространяются способ задания меры $B(S)$ и приведенная в начале статьи общая схема распознавания.

Рассмотрение обоих описанных выше алгоритмов с этой точки зрения показывает, что они (с учетом некоторых несущественных изменений) могут быть применены для обработки многозначных таблиц. Отправной точкой такого рассмотрения может послужить следующий очевидный факт.

Утверждение 2. Пусть квадратная матрица (таблица) $C = (C_{ij})_{e\times e}$ и строка $\tilde{\sigma} = (a_1,\dots,a_e)$, составленные из натуральных чисел, таковы, что для пары таблиц

$$\begin{bmatrix} C_{11} & C_{12} & \dots & C_{1e} \\ C_{21} & C_{22} & \dots & C_{2e} \\ \dots & \dots & \dots & \dots \\ C_{e1} & C_{e2} & \dots & C_{ee} \end{bmatrix}$$
$$\begin{bmatrix} a_1 & a_2 & \dots & a_e \end{bmatrix}$$

набор столбцов ($x_1, x_2, \dots, x_e$) образует тупиковый тестор. Тогда перестановкой строк матрица C может быть приведена к следующему виду

$$\begin{bmatrix} b_1 & a_2 & a_3 & \dots & a_{e-1} & a_e \\ a_1 & b_2 & a_3 & \dots & a_{e-1} & a_e \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_1 & a_2 & a_3 & \dots & b_{e-1} & a_e \\ a_1 & a_2 & a_3 & \dots & a_{e-1} & b_e \end{bmatrix}$$

---

*) Подразумевается также, что функции из $\Omega(j_1,\dots,j_e)$ не равны тождественно нулю или единице.

где $b_j \neq a_j$, $j = 1, \ldots, \ell$.

Таким образом, как и в бинарном случае, в каждом столбце обнаруживаются повторяющийся и выделенный элементы. Исходя из этого, легко внести соответствующие изменения в описания процедур. В частности, при $\ell = 2$ возникает два варианта расстановки повторяющегося и выделенного элементов, располагающих выделенные элементы в разных строках.

Пример 4.

$$T_I = \begin{bmatrix} 0 & I & 2 \\ I & 2 & 2 \\ 2 & 0 & I \end{bmatrix}, \quad T_2 = \begin{bmatrix} 3 & 2 & I \\ 2 & I & 0 \end{bmatrix},$$

Логическая матрица

I	2	3	4
0	I	0	0
I	0	0	0

Здесь $m_1 = 3$, $m_2 = 2$, $m_3 = m_4 = 0$. Эффективными тупиковыми T-наборами являются (x_1), (x_3), (x_1, x_2), (x_2, x_3), сквозных т.Т-н нет. $D_1 = \{0,1,2,3\}$, $D_2 = \{0,1,2\}$, $D_3 = \{0,1,2\}$;веса строк $B(s_1^2) = 0{,}25$, $B(s_2^2) = 0{,}25$; важность исходных признаков 0,5; 0,5; 0,5.

Литература

I. В.О.Красавчиков. Модификация тестового подхода к анализу таблиц описаний на основе понятия пакета. - В кн.: Дискретный анализ. Вып.26. Новосибирск, I974, с.36-60.

2. А.Н.Дмитриев, Ю.И.Журавлев, Ф.П.Кренделев. О математических принципах классификации предметов и явлений. - В кн.: Дискретный анализ. Вып.7. Новосибирск, I966, с.3-I5.

3. Организация и обработка геологической информации с помощью ЭВМ на основе построения тупиковых тестов. - В кн.: Логико-информационные решения геологических задач. М., "Наука", I975, с.83-I28. - Авт.: Дмитриев А.Н., Кренделев Ф.П., Бишаев А.А.

4. Алгоритмы и программы решения геологических задач на ЭВМ "Минск-2" и "БЭСМ-3М", Алма-Ата, I969, вып.2. - Авт.: Бугаец А.Н., Дворниченко Г.К., Мацак А.П. и др.

5. С.В.Яблонский. Введение в теорию функций к-значной логики. - В кн.: Дискретная математика и математические вопросы кибернетики, М., I974, с.9-65.

6. П.Супес, Дж.Зинес. Основы теории измерений. - В кн.: Психологические измерения, М., I967.

С О Д Е Р Ж А Н И Е

Ответственная за выпуск Т.И.Штатнова

Технический редактор *Л. А. Панина*

Подписано к печати 25. I. 1977г. МН 02620

Бумага 60x84/16. Печ.л. I0.25. Уч.-изд.л. 8,0.

Тираж 500. Заказ I30. Цена 70 коп.

Институт геологии и геофизики СО АН СССР

Новосибирск, 90. Ротапринт.