5. ЛЕКЦИЯ. Кластерный анализ

В настоящее время разработано достаточное количество методов и алгоритмов кластеризации, но, к сожалению, не все они могут эффективно работать с большими массивами данных, поэтому дальнейшие исследования в этом направлении связаны с преодолением этой проблемы. Одним из широко известных в аналитическом сообществе алгоритмов кластеризации, позволяющих эффективно работать с большими объемами данных, является EM-алгоритм.

Статистические алгоритмы. EM-алгоритм

Алгоритм основан на методике итеративного вычисления оценок максимального правдоподобия (МП). С помощью этого метода максимального правдоподобия производится точечная оценка неизвестных параметров априорно известного закона распределения случайной величины.

EM-алгоритм – очень общий итеративный алгоритм для МП-оценивания в задачах с неполными данными. На деле круг задач, которые можно решать с помощью EM-алгоритма, очень широк. В основе идеи EM-алгоритма лежит предположение, что исследуемое множество данных может быть смоделировано с помощью линейной комбинации многомерных нормальных распределений, а целью является оценка параметров распределения, которые максимизируют логарифмическую функцию правдоподобия, используемую в качестве меры качества модели. Иными словами, предполагается, что данные в каждом кластере подчиняются определенному закону распределения, а именно, нормальному распределению. С учетом этого предположения можно определить параметры –математическое ожидание и дисперсию, которые соответствуют закону распределения элементов в кластере, наилучшим образом «подходящему» к наблюдаемым данным. Таким образом, в EM-алгоритме формализована относительно старая идея обработки неполных данных: 1) заполнение пропусков оценками пропущенных значений; 2) оценивание параметров; 3) повторное оценивание пропущенных значений, при этом оценки параметров считаются точными; 4) повторное оценивание параметров и так далее до сходимости процесса.

Предполагается, что любое наблюдение принадлежит ко всем кластерам, но с разной вероятностью. Тогда задача будет заключаться в «подгонке» распределений смеси к данным, а затем в определении вероятностей принадлежности наблюдения к каждому кластеру. Очевидно, что наблюдение должно быть отнесено к тому кластеру, для которого данная вероятность выше.

Среди преимуществ EM-алгоритма можно выделить следующие:

- Мощная статистическая основа.
- Линейное увеличение сложности при росте объема данных.
- Устойчивость к шумам и пропускам в данных.
- Возможность построения желаемого числа кластеров.
- Быстрая сходимость при удачной инициализации.

Однако алгоритм имеет и ряд недостатков. Во-первых, предположение о нормальности всех измерений данных не всегда выполняется. Во-вторых, при неудачной инициализации сходимость алгоритма может оказаться медленной. Кроме этого, алгоритм может остановиться в локальном минимуме и дать квазиоптимальное решение (приближенное к оптимальному, но выбираемое из ограниченного количества вариантов.).

Статистические основы алгоритма

Как отмечалось, EM-алгоритм предполагает, что кластеризуемые данные подчиняются линейной комбинации (смеси) нормальных (гауссовых) распределений. Плотность вероятности нормального распределения имеет вид:

$$p(x) = \frac{1}{\sqrt{2 \times \pi \times \sigma^2}} \times exp\left\{-\frac{(x-\mu)^2}{2 \times \sigma^2}\right\}$$

где

$\mu = E(X)$ – математическое ожидание,

$\sigma^2 = E(X-\mu)^2$ – дисперсия.

Многомерное нормальное распределение для q-мерного пространства является обобщением предыдущего выражения. Многомерная нормальная плотность для q-мерного вектора $x = (x_1, x_2, \dots, x_q)$ может быть записана в виде:

$$p(x) = \frac{1}{(2 \times \pi)^{\frac{q}{2}} \times \sqrt{|\Sigma|}} \times exp\left\{-\frac{1}{2} \times (x-\mu)^T \times \Sigma^{-1} \times (x-\mu)\right\}$$

где

$\sum$ – ковариационная матрица размером $q \times q$, которая, как известно, является обобщением дисперсии для многомерной случайной величины;

μ – представляет из себя q-мерный вектор математических ожиданий;

$|\sum|$ – определитель ковариационной матрицы;

T – оператор транспонирования.

Введем в рассмотрение функцию $\delta^2 = (x-\mu)^T \times \sum^{-1} \times (x-\mu)$, называемую квадратичным расстоянием Махаланобиса.

Расстояние Махаланобиса – мера расстояния между векторами случайных величин, обобщающая понятие евклидова расстояния. С помощью расстояния Махаланобиса можно определять сходство неизвестной и известной выборки.

Оно отличается от расстояния Евклида тем, что учитывает корреляции между переменными и инвариантно к масштабу.

Алгоритм предполагает, что данные подчиняются смеси многомерных нормальных распределений для q переменных. Модель, представляющая собой смесь гауссовых распределений, задается в виде:

$$p(x) = \sum_{t=1}^{k} w_i \times p(x \mid i)$$

где

$p(x \mid i)$ – нормальное распределение для i-го кластера;

w_i – доля (вес) i-го кластера в исходной базе данных.

Существуют два подхода к решению задач кластеризации: основанный на расстоянии и основанный на плотности. Первый подход заключается в определении областей пространства признаков, внутри которых точки данных расположены ближе друг к другу, чем к точкам других областей, относительно некоторой функции расстояния (например, евклидовой). Второй – обнаруживает области, которые являются более «заселенными», чем другие. Алгоритмы кластеризации могут работать сверху вниз (иерархические) и снизу вверх (агломеративные). Агломеративные алгоритмы, как правило, являются более точными, хотя и работают медленнее.

Алгоритм EM основан на вычислении расстояний. Он может рассматриваться как обобщение кластеризации на основе анализа смеси вероятностных распределений. В процессе работы алгоритма происходит итеративное улучшение решения, а остановка осуществляется в момент, когда достигается требуемый уровень точности модели. Мерой в данном случае является монотонно увеличивающаяся статистическая величина, называемая логарифмическим правдоподобием. Целью алгоритма является оценка средних значений C, ковариаций R и весов смеси W для функции распределения вероятности, описанной выше. Параметры, оцененные алгоритмом, сохраняются в таблице вида:

Таблица 5.1.

Матрица	Размер	Содержит
C	$q \times k$	Математические ожидания, μ
R	$q \times q$	Ковариации, $\sum$
W	$k \times 1$	Веса, w_i

Следует отметить, что один из популярных алгоритмов кластеризации k-means является частным случаем EM-алгоритма, когда W и R постоянны: $w_i =$

$\frac{1}{k}$, $R = I$ (I - единичная матрица).

Алгоритм начинает работу с инициализации, т.е. некоторого приближенного решения, которое может быть выбрано случайно или задано пользователем исходя из некоторых априорных сведений об исходных данных. Наиболее общим способом инициализации является присвоение элементам матрицы математических ожиданий случайных значений $C \leftarrow \mu$ Random, начальная ковариационная матрица определяется как единчиная $r \leftarrow I$, веса кластеров задаются одинаковыми ($w_i \leftarrow \frac{1}{k}$).

Следует обратить внимание, что алгоритм может «застрять» в локальном оптимуме и дать квазиоптимальное решение при выборе неудачного начального приближения. Поэтому одним из его недостатков следует считать чувствительность к выбору начального состояния модели.

Реализация EM-алгоритма может быть проиллюстрирована с помощью следующего псевдокода:

Вход: k – число кластеров;

$Y = \{y_1, y_2, \dots, y_n\}$ – множество из n наблюдений q -мерного пространства;

ε – допустимое отклонение для логарифмического правдоподобия;

Q – максимальное число итераций.

Выход: C, R, W – матрицы, содержащие обновляемые параметры смеси.

X – матрица с вероятностями членства в кластерах.

1. Инициализация: установка начальных значений C, R, W, выбранных случайно или заданных пользователем.

2. Пока изменение логарифмического правдоподобия $\Delta llh \geq \varepsilon$ и не достигнуто максимальное число итераций Q, выполнять шаги ***E*** и ***M.***

Шаг E

$C' = 0, R' = 0, W' = 0, llh = 0$

Для i, изменяющегося от 1 до n

$sump_i = 0$

Для j, изменяющегося от 1 до k

$$\delta_{ij} = (y_i - C_j)^T \times R_j^{-1} \times (y_i - C_j)$$

$$p_{ij} = \frac{w_j}{(2 \times \pi)^{\frac{q}{2}} \times |R_j|^{\frac{1}{2}}} \times exp\{-\frac{1}{2} \times \delta_{ij}\}$$

$$sump_i = sump_i + p_{ij}$$

Конец цикла по j

$x_i = \frac{p_i}{sump_i}, llh = llh + ln(sump_i)$

$C' = C' + y_i\, x_i{}^T,\ W' = W' + x_i$

Конец цикла по i

Шаг M

Для j, изменяющегося от 1 до k

$$C_j' = \frac{C_j'}{W_j'}$$

Для i, изменяющегося от 1 до n

$$R_j' = R_j' + (y_i - C_j)x_{ij}(y_i - C_j)^T$$

Конец цикла по i

$R_j = \frac{R_j'}{n}, W = \frac{W'}{n}$

Конец цикла по j

Алгоритм содержит два шага: шаг ожидания (expectation) или *E*-шаг и шаг максимизации (maximization) или *M*-шаг. Каждый из них повторяется до тех пор, пока изменение логарифмического правдоподобия Δllh не станет меньше, чем ε, или пока не будет достигнуто максимальное число итераций.

Логарифмическое правдоподобие вычисляется как:

$$llh = \sum_{i=1}^{n} ln(sump_i)$$

Переменные δ, R, P представляют собой матрицы, хранящие расстояния Махаланобиса, ковариации и вероятности членства в кластере для каждой из n точек. C', R' и W' являются временными матрицами, используемыми только для вычислений. $||\ W\ ||= 1$, т.е. $\sum_{i=1}^{k} w_i = 1$. Обозначение вида p_i, использованное в псевдокоде, обозначает k-размерный вектор принадлежности i-го наблюдения к каждому из k кластеров. Соответственно, x_i – нормированная вероятность принадлежности к каждому из k кластеров. Столбец C_j матрицы C есть оценка математического ожидания по j-му кластеру, R – диагональная матрица, т.е. $R_{ij} = 0$ для всех $i \neq j$. Со статистической точки зрения это означает, что ковариации являются независимыми.

Диагональность является ключевым предположением, которое делает алгоритм масштабируемым. В этом случае детерминант матрицы и его обращение может быть вычислено за время $O(p)$, а алгоритм имеет сложность $O(kpn)$. В случае недиагональной матрицы сложность алгоритма составит $O(kp^2n)$, т.е. будет квадратично возрастать с увеличением размерности данных.

Важнейшим действием, выполняемым на *E*-шаге, является вычисление расстояний Махаланобиса δ_{ij}. Если матрица R является диагональной, то расстояние Махаланобиса от точки y до среднего значения кластера C, имеющего ковариацию R, будет:

$$\delta_{ij} = (y - C)^T R^{-1} (y - C) = \sum_{k=1}^{q} \frac{(y_k - C_k)^2}{R_{kk}}$$

поскольку ${R_{kk}}^{-1} = 1/R_{kk}, \overline{k = 1, q}$. Если матрица R является диагональной, то ее обращение R^{-1} легко вычисляется, т.к. R^{-1} для любых $k \neq l$. Кроме этого, ускорению вычислений способствует то, что диагональная матрица R может храниться в виде вектора ее диагональных элементов. Поскольку R не изменяется в процессе *E*-шага, ее детерминант вычисляется только единожды, что делает вычисление вероятностей p_{ij} более быстрым. На *M*-шаге диагональность матрицы R также упрощает вычисления, поскольку недиагональные элементы матрицы $(y_i - C_j)x_{ij}(y_i - C_j)^T$ равны нулю.

Для оптимизации используемого объема памяти, алгоритм может работать в двух режимах. В первом загружается только часть доступных данных и на их основе предпринимается попытка построения модели. Если она увенчалась успехом, то алгоритм завершает работу, в противном случае загружается следующая порция данных и т.д., пока не будут получены приемлемые результаты. Во втором режиме загружаются сразу все имеющиеся данные. Как правило, последний вариант обеспечивает более точную подгонку модели, но предъявляет более жесткие требования к объему доступной оперативной памяти.

Упрощенный пример.

Пусть даны две монеты A и B, которые будет подбрасываться. Равновероятностно выбирается любая монета 5 раз. Каждая выбранная монета взбрасывается 10 раз. Получаем ряд событий, состоящих из орлов и орешек.

Таблица 5.2

Выбор	Событие	монета А	монета В
В	О Р Р Р О О Р О Р О		5 - орлов и 5 - решек
А	О О О О Р О О О О О	9 - орлов и 1 - решка	
А	О Р О О О О О Р О О	8 - орлов и 2 - решки	
В	О Р О Р Р Р О О Р Р		4 - орла и 6 - решек
А	Р О О О Р О О О Р О	7 - орлов и 3 - решки	
Сумма орлов и решек для монет		24 О, 6 Р	9 О, 11 Р

Если известно, что какая монета подбрасывается, то можно найти вероятность выпадения орла для каждой монетки:

$$p(A) = \frac{24}{24+6} = 0{,}8$$
$$p(B) = \frac{9}{9+11} = 0{,}45$$

Теперь рассмотрим, события выпадения орлов и решек, при этом не известно какая монета подбрасывалась. Таким образом, выбор монеты является скрытой переменной. Выдвинем первоначальные гипотезы:

1. При 10 подбрасываний монеты *A* больше выпадают орлы;
2. При 10 подбрасываний монеты *B* больше выпадают решки.

ЕМ-алгоритм начинается с первоначального предположения о параметрах. Предполагаем, что начальные параметры:

$$p(A)^0 = 0{,}6$$
$$p(B)^0 = 0{,}5$$

Есть выборка тех же событий, но соответственно монеты неизвестны.

Таблица 5.3

№	Выбор	Событие
1	?	O P P P O O P O P O
2	?	O O O O P O O O O O
3	?	O P O O O O O P O O
4	?	O P O P P P O O P P
5	?	P O O O P O O O P O

На этапе Е-шага вычисляется распределение вероятностей по возможным завершениям с использованием текущих параметров. Будем рассматривать строку №2 таблицы 5.3. Определим функцию правдоподобия исходя из биномиального распределения.

Биномиальное распределение – дискретное распределение вероятностей случайной величины X принимающей целочисленные значения $k = 0,1,\dots,n$ с вероятностями:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

где $\binom{n}{k} = \frac{n!}{(n-k)!k!}$ – биномиальный коэффициент.

Данное распределение характеризуется двумя параметрами: целым числом $n > 0$, называемым числом испытаний, и вещественным числом $p, 0 \le p \le 1$, называемом вероятностью успеха в одном испытании. Биномиальное распределение – одно из основных распределений вероятностей, связанных с последовательностью независимых испытаний. Если проводится серия из n независимых испытаний, в каждом из которых может произойти «успех» с

вероятностью p то случайная величина, равная числу успехов во всей серии, имеет указанное распределение. Эта величина также может быть представлена в виде суммы $X = X_1 + \cdots + X_n$ независимых слагаемых, имеющих распределение Бернулли.

Тогда вероятность выпадения орлов k при 10 подбрасываний монеты $i \in \{A, B\}$

$$p_i(k) = \binom{10}{k} p^k (1-p)^{10-k}$$

Учитывая, что во второй строке 9 орлов и 1 решка, а также принимая во внимание начальные параметры $p(A)^0 = 0{,}6$ и $p(B)^0 = 0{,}5$. Биномиальный коэффициент одинаков для обеих монет, поэтому он выпадает при нормализации.

$$p_A(9) = (p(A)^0)^9 (1 - p(A)^0)^{10-9} = 0{,}6^9 \times 0{,}4 = 0{,}004$$
$$p_B(9) = (p(B)^0)^9 (1 - p(B)^0)^{10-9} = 0{,}5^9 \times 0{,}5 = 0{,}001$$

Вероятно, строка №2 относится к монете *A*, которая определена большим количеством орлов при сбрасывании. Оцениваем апостериорную вероятность:

$$P(A|X) = \frac{p_A(9)}{p_A(9) + p_B(9)} = \frac{0{,}004}{0{,}004 + 0{,}001} = 0{,}8$$
$$P(B|X) = \frac{p_B(9)}{p_A(9) + p_B(9)} = \frac{0{,}001}{0{,}004 + 0{,}001} = 0{,}2$$

Остальные строки считаются аналогично.

Далее считается математическое ожидание. Подсчеты, показанные в таблице 5.4, представляют собой ожидаемое количество орлов и решек в соответствии с этим распределением.

Таблица 5.4

№	монета A	монета B
1	5 – орлов $0{,}45 \times 5 \approx 2{,}2$ 5 – решек $0{,}45 \times 5 \approx 2{,}2$	5 – орлов $0{,}55 \times 5 \approx 2{,}8$ 5 – решек $0{,}55 \times 5 \approx 2{,}8$
2	9 – орлов $0{,}8 \times 9 = 7{,}2$ 1 – решка $0{,}8 \times 1 = 0{,}8$	9 – орлов $0{,}2 \times 9 = 1{,}8$ 1 – решка $0{,}2 \times 1 = 0{,}2$
3	8 – орлов $0{,}73 \times 8 \approx 5{,}9$ 2 – решки $0{,}73 \times 2 \approx 1{,}5$	8 – орлов $0{,}27 \times 8 \approx 2{,}1$ 2 – решки $0{,}27 \times 2 \approx 0{,}5$
4	4 – орла $0{,}35 \times 4 = 1{,}4$ 6 – решек $0{,}35 \times 6 = 2{,}1$	4 – орла $0{,}65 \times 4 = 2{,}6$ 6 – решек $0{,}65 \times 6 = 5{,}9$
5	7 – орлов $0{,}65 \times 7 \approx 4{,}5$ 3 – решки $0{,}65 \times 3 \approx 1{,}9$	7 – орлов $0{,}35 \times 7 \approx 2{,}5$ 3 – решки $0{,}35 \times 3 \approx 1{,}1$
Итог	$= 21{,}3(\text{орлы}), 8{,}6(\text{решка})$	$= 11{,}7(\text{орлы}), 8{,}4(\text{решка})$

На *M*-шаге новые параметры определяются с использованием текущих завершений.

$$p(A)^1 \approx \frac{21{,}3}{21{,}3 + 8{,}6} \approx 0{,}71$$

$$p(B)^1 \approx \frac{11{,}7}{11{,}7 + 8{,}4} \approx 0{,}58$$

После нескольких (десяти) повторений (итераций) Е-шага и М-шага алгоритм сходится.

$$p(A)^1 \approx 0{,}8$$

$$p(B)^1 \approx 0{,}52$$

Если эти два параметра оценены достаточно хорошо, вероятность будет для них высокой (приближенной к истинной вероятности).

Иерархическая кластеризация

Алгоритмы иерархической кластеризации называются также графовыми алгоритмами кластеризации, которые строят не одно разбиение выборки на непересекающиеся классы, а систему вложенных разбиений (таксономии). Результат разбиения обычно представляется в виде таксономического дерева — дендрограммы. Дендрограмма – дерево (граф без циклов), построенное по матрице мер близости. Дендрограмма позволяет изобразить взаимные связи между объектами из заданного множества. Для создания дендрограммы требуется матрица сходства, которая определяет уровень сходства между любой парой объектов.

Среди алгоритмов иерархической кластеризации различаются два основных типа:

1) Дивизимные или нисходящие алгоритмы разбивают выборку на всё более и более мелкие кластеры. Таким образом, в начале работы алгоритма все объекты принадлежат одному кластеру, который на последующих шагах делится на меньшие кластеры и в результате образуется последовательность расщепляющих групп.

2) Агломеративные или восходящие алгоритмы, в которых объекты объединяются во всё более и более крупные кластеры. Это метод слияния – проводится рекурсивное поочерёдное слияние близких кластеров. При слиянии отдельные точки-объекты также считаются кластерами. На каждом шаге выбирается для слияния пара наиболее схожих кластеров. На последующих шагах объединение продолжается до тех пор, пока все объекты не будут составлять один кластер.

Более распространены агломеративные или восходящие алгоритмы, в

которых объекты объединяются во всё более и более крупные кластеры.

Сначала каждый объект считается отдельным кластером. Для одноэлементных кластеров естественным образом определяется функция расстояния

$$R(\{x\},\{x'\}) = \rho(x,x')$$

Затем запускается процесс слияний. На каждой итерации вместо пары самых близких кластеров U и V образуется новый кластер $W = U \cup V$. Расстояние от нового кластера W до любого другого кластера S вычисляется по расстояниям $R(U,V)$, $R(U,S)$ и $R(V,S)$, которые к этому моменту уже должны быть известны:

$$R(U \cup V, S) = \alpha_U R(U,S) + \alpha_V R(V,S) + \beta R(U,V) + \gamma |R(U,S) - R(V,S)|,$$

где α_U, α_V, β, γ – числовые параметры. Эта универсальная формула обобщает практически все разумные способы определить расстояние между кластерами. Она была предложена Лансом и Уильямсом в 1967 году.

Реализация агломеративная кластеризация Ланса-Уильямса может быть проиллюстрирована с помощью следующего псевдокода:

1: инициализировать множество кластеров C_1:

$t := 1;\ C_t = \{\{x_1\},\ldots,\{x_\ell\}\}$;

2: **для всех** $t = 2,\ldots,\ell$ (t – номер итерации):

3: найти в C_{t-1} два ближайших кластера:

$(U,V) := arg\min_{U \neq V} R(U,V)$;

$Rt := R(U,V)$;

4: изъять кластеры U и V, добавить слитый кластер $W = U \cup V$:

$C_t := C_{t-1} \cup \{W\} \backslash \{U,V\}$;

5: для всех $S \in C_t$

6: вычислить расстояние $R(W,S)$ по формуле Ланса-Уильямса;

где \ – Разность множеств. $A \backslash B$ означает множество элементов, принадлежащих A, но не принадлежащих B. Например, $\{1,2,3,4\} \backslash \{3,4,5,6\} = \{1,2\}$.

Рассмотрим пример.

Необходимо провести кластеризацию пяти предприятий, каждое из которых характеризуется тремя переменными: x_1 – среднегодовая стоимость основных производственных фондов, млрд. руб.; x_2 – материальные затраты на 1 руб. произведенной продукции; x_3 – объем произведенной продукции, млрд. руб. Значения переменных приведены в табл. 5.5.

Таблица 5.5 – Финансовые показатели производственных предприятий

Номер предприятия	x_1	x_2	x_3
1	120,0	94,0	164,0
2	85,.0	75,2	92,0
3	145,0	81,0	120,0
4	78,0	76,8	86,0
5	70,0	75,9	104,0
Среднее значение, $\overline{x_k}$	99,6	80,6	113,2
Среднее квадратическое отклонение, s_k	28,4	10,9	27,9

Произведем нормировку исходных данных согласно формуле:

$$z_k^i = \frac{x_k^i - \overline{x_k}}{s_k}$$

Рассчитаем $z_1^1 = \frac{x_1^1 - \overline{x_1}}{s_1} = \frac{120,0 - 99,6}{28,4} \approx 0,718$

Таким образом, заполняем нормированную матрицу:

$$Z = \begin{pmatrix} 0,718 & 1,229 & 1,821 \\ -0,514 & -2,238 & -0,760 \\ 1,514 & 0,037 & 0,244 \\ -0,760 & -0,349 & -0,975 \\ -1,042 & -0,431 & -0,330 \end{pmatrix}$$

Классификацию проведем при помощи иерархического агломеративного метода. Для построения матрицы расстояний воспользуемся квадратом евклидового расстояния. Тогда, например, квадрат расстояния между первым и вторым объектами:

$$d^2(1,2) = (0,718 - (-0,514))^2 + (1,229 - (-2,238))^2 + (1,821 - (-0,760))^2 = 20,20$$

Квадрат расстояния между первым и третьим объектами:

$$d^2(1,3) = (0,718 - 1,514)^2 + (1,229 - 0,037)^2 + (1,821 - 0,244)^2 = 4,54$$

В результате получаем первоначальную матрицу расстояний, характеризующую расстояния между отдельными объектами, каждый из которых изначально является отдельным кластером:

$$D_0^2 = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 20,20 & 4,54 & 12,49 & 10,48 \\ 2 & & 0 & 10,30 & 3,68 & 3,73 \\ 3 & & & 0 & 6,81 & 7,08 \\ 4 & & & & 0 & \mathbf{0,50} \\ 5 & & & & & 0 \end{array}$$

Как видно по элементам матрицы D_0^2, наиболее близкими являются

объекты 4 и 5: $d^2(4,5) = 0.50$. Объединим их в один кластер и присвоим ему метку 45. Пересчитаем расстояния всех оставшихся объектов (кластеров) до кластера 45. В матрице D_1^2 расстояния между кластерами определяются по алгоритму «дальнего соседа».

В методе наиболее удаленных соседей расстояния между кластерами определяются наибольшим расстоянием между любыми двумя объектами в различных кластерах (т.е. «наиболее удаленными соседями»).

Расстояние между кластером 1 и кластером 45:

$$d^2(1,45) = max(d^2(1,4), d^2(1,5)) = max(12.49, 10.48) = 12.49.$$

Расстояние между кластером 2 и кластером 45:

$$d^2(2,45) = max(d^2(2,4), d^2(2,5)) = max(3.68, 3.73) = 3.73.$$

Расстояние между кластером 3 и кластером 45:

$$d^2(3,45) = max(d^2(3,4), d^2(3,5)) = max(6.81, 7.08) = 7.08.$$

Получим новую матрицу расстояний:

$D_1^2 =$

	1	2	3	45
1	0	20,20	4,54	12,49
2		0	10,30	**3,73**
3			0	7,08
45				0

В матрице D_1^2 1 D опять находим самые близкие кластеры. Это будут кластеры 2 и 45, $d^2(2,45) = 3.73$. Следовательно, на этом шаге объединяем кластеры 2 и 45; получим новый кластер, состоящий из кластеров 2 и 45 (объекты 2,4,5). Присвоим ему метку 245. Теперь имеем три кластера: 1, 245, 3. Пересчитываем расстояния $d^2(1,\ 245)$, $d^2(3,\ 245)$ и получаем матрицу D_2:

$$d^2(1,245) = max(d^2(1,2), d^2(1,45)) = max(20.2, 12.49) = 20.2.$$

$$d^2(3,245) = max(d^2(3,2), d^2(3,45)) = max(10.3, 7.08) = 10.3.$$

Следующая матрица выглядит таким образом:

$D_2^2 =$

	1	245	3
1	0	20,20	**4,54**
245		0	10,30
3			0

На следующем шаге объединяем кластеры 1 и 3 ($d^2(1,3) = 4.54$) в один кластер и присваиваем ему номер 13. Теперь имеем только два кластера: 13, 245.

$$d^2(13,245) = max(d^2(1,245), d^2(3,245)) = max(20.20, 10.3) = 20.20$$

И, наконец, на последнем шаге объединяем кластеры 13 и 245 на квадрате расстояния 20.20.

$$D_1^2 = \begin{matrix} & 13 & 245 \\ 13 & 0 & 20{,}20 \\ 245 & & 0 \end{matrix}$$

Представим результаты классификации в виде дендрограммы (рис. 5.1). Дендрограмма свидетельствует о том, что кластер S_2 более однороден по составу входящих объектов, так как в нем объединение происходит при меньших расстояниях, чем в кластере S_1.

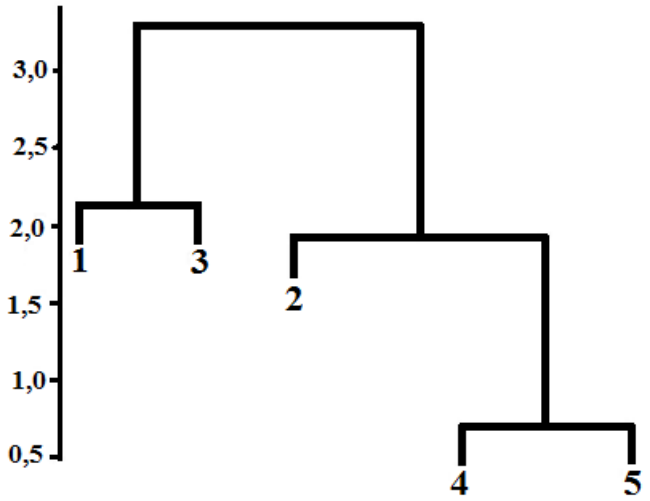

Рис. 5.1. Дендрограмма кластеризации пяти производственных предприятий

Различные алгоритмы отличаются выбором метрики измерения расстояний и критерия схожести кластеров. На практике используются следующие способы вычисления расстояний $R(W,S)$ между кластерами W и S. Для каждого из них доказано соответствие формуле Ланса-Уильямса при определённых сочетаниях параметров.

Методы измерения расстояний между кластерами.

Метод ближнего соседа. Здесь расстояние между двумя кластерами определяется расстоянием между двумя наиболее близкими объектами (ближайшими соседями) в различных кластерах. Расстояние ближнего соседа:

$$R^{б}(W,S) = \min_{w\in W, s\in S} \rho(w,s)\,; \quad \alpha_U = \alpha_V = \frac{1}{2}, \beta = 0, \gamma = -\frac{1}{2}$$

Метод наиболее удаленных соседей. Здесь расстояния между кластерами определяются наибольшим расстоянием между любыми двумя объектами в различных кластерах (т.е. «наиболее удаленными соседями»).

$$R^{д}(W,S) = \max_{w\in W, s\in S} \rho(w,s)\,; \quad \alpha_U = \alpha_V = \frac{1}{2}, \beta = 0, \gamma = -\frac{1}{2}$$

Метод невзвешенного попарного среднего. В качестве расстояния между двумя кластерами берется среднее расстояние между всеми парами объектов в них. Среднее расстояние:

$$R^{\text{с}}(W,S)=\frac{1}{|W||S|}\sum_{w\in W}\sum_{s\in S}\rho(w,s)\,;\qquad \alpha_U=\frac{|U|}{|W|},\alpha_V=\frac{|V|}{|W|},\beta=\gamma=0$$

Метод взвешенного попарного среднего. Этот метод похож на метод невзвешенного попарного среднего, разница состоит лишь в том, что здесь в качестве весового коэффициента используется размер кластера (число объектов, содержащихся в кластере).

Невзвешенный центроидный метод. В качестве расстояния между двумя кластерами в этом методе берется расстояние между их геометрическими центрами. Центром кластера S_{k} (центроидом) называется геометрический центр точек кластера k в евклидовом пространстве:

$$X_k=\frac{1}{|S_k|}\sum_{x^i\in S_k}x^i$$

где $|S_k|$ – число точек в кластере k, $k=1,2,3,\dots,K$; K – число кластеров.

Расстояние между центрами:

$$R^{\text{ц}}(W,S)=\rho^2\left(\sum_{w\in W}\frac{w}{|W|},\sum_{s\in S}\frac{s}{|S|}\right);\quad \alpha_U=\frac{|U|}{|W|},\alpha_V=\frac{|V|}{|W|},\beta=-\alpha_U\alpha_V,\gamma=0$$

Взвешенный центроидный метод. Для учета разницы между размерами кластеров (числе объектов в них) используются веса.

Метод Варда. В качестве расстояния между кластерами берется прирост суммы квадратов расстояний объектов до центров кластеров, получаемый в результате их объединения. В отличие от других методов кластерного анализа для оценки расстояний между кластерами здесь используются методы дисперсионного анализа. На каждом шаге алгоритма объединяются такие два кластера, которые приводят к минимальному увеличению целевой функции, т.е. внутригрупповой суммы квадратов. Расстояние Варда:

$$\text{R}^{\text{в}}(\text{W},\text{S})=\frac{|\text{S}||\text{W}|}{|\text{S}|+|\text{W}|}\rho^2\left(\sum_{w\in W}\frac{w}{|W|},\sum_{s\in S}\frac{s}{|S|}\right);\quad \alpha_U=\frac{|\text{S}|+|U|}{|\text{S}|+|W|},\ \alpha_V=\frac{|\text{S}|+|V|}{|\text{S}|+|W|},$$

$$\beta=\frac{-|\text{S}|}{|\text{S}|+|W|},\gamma=0$$