12.4.3. Линейные решающие функции

Рассмотрим алгоритм построения линейных решающих функций. Линейная функция имеет следующий вид: $D(X) = w_0 + \sum_{j=1}^{n} w_j \cdot x_j$. Цель алгоритма — найти коэффициенты w_j решающей функции $D(X)$ методом последовательного уточнения. Рассмотрим случай двух классов (выше было показано, какими приемами можно свести к этому случаю вариант нескольких классов). Основой для вычисления коэффициентов w_j является анализ обучающей выборки $\tilde{X}$, где известна заранее принадлежность объектов $\tilde{X}$ классу 1 или классу 2. Далее эти два класса обозначим C_1 и C_2.

Решающая функция считается построенной, если все объекты обучающей выборки $\tilde{X}$ распознаются этой функцией правильно, то есть $D(X) > 0$, если $X \in C_1$, и, соответственно, $D(X) < 0$, если $X \in C_2$. Коррекция коэффициентов решающей функции выполняется по следующему правилу: коэффициенты решающей функции увеличиваются при неправильном распознавании объекта из класса C_1, уменьшаются при неправильном распознавании объекта из класса C_2 и остаются без изменения, если распознавание идет правильно.

Если на некотором шаге произойдет корректировка коэффициентов решающей функции, счетчик правильно распознанных объектов, обозначаемый далее как $сч$, сбрасывается на ноль, поскольку мы перешли к новой функции, и теперь ее надо проверить заново на всех элементах обучающей выборки.

Алгоритм завершается, когда окажется, что построенная решающая функция $D(X)$ правильно распознает все объекты обучающего множества.

Алгоритм 12.4.

1. Получить обучающую выборку $\tilde{X} = (X_1, X_2, \ldots, X_M)$, элементы которой принадлежат непересекающимся классам C_1 или C_2.
2. Установить в ноль счетчик правильно распознанных объектов: $сч = 0$.
3. Установить номер итерации равным нулю: $k = 0$.
4. Задать начальные значения коэффициентов w_j в решающей функции (например, $w_j = 0$ для $j = 1, 2, \ldots, n$). Получим решающую функцию $D_0(X)$.

5. Выбираем класс C_1 в качестве текущего класса.

6. Переход к новой итерации: $k = k + 1$.

7. Выбрать очередной объект X_k текущего класса (класса C_1). Если класс C_1 исчерпан, объявить текущим классом класс C_2, выбрать первый объект этого класса.

8. Вычислить новые значения коэффициентов решающей функции на итерации k: $w_j^k = w_j^{k-1} + c \cdot x_{kj}$, где c — множитель, определяемый из условия

$$c = \begin{cases} 1, & \sum\limits_{j=0}^{n} w_j^{k-1} \cdot x_{kj} \leqslant 0, \quad \text{и } X_k \in C_1; \\ -1, & \sum\limits_{j=0}^{n} w_j^{k-1} \cdot x_{kj} > 0, \quad \text{и } X_k \in C_2; \\ 0 & \text{\textit{при правильном распознавании.}} \end{cases}$$

9. Если $c = 0$, $сч = сч + 1$ (увеличиваем на 1 число правильно распознанных объектов), иначе $сч = 0$.
10. Если $сч = M$ — общему числу объектов обучающей выборки $\tilde{X}$, то КОНЕЦ, иначе перейти к шагу 6.

Приведенный алгоритм обеспечивает построение решающей функции во всех случаях, когда классы являются линейно разделимыми.

Пример 12.4.

Рассмотрим пример работы алгоритма построения линейной разделяющей функции. В табл. 12.10 дана обучающая выборка — объекты, принадлежащие двум классам.

Таблица 12.10

	Точки X	Координаты точек
Класс C_1	1	$<1;\ 2>$
	2	$<1;\ 3>$
	3	$<3;\ 3>$
Класс C_2	4	$<4;\ 1>$
	5	$<5;\ 2>$
	6	$<6;\ 2>$

Каждый объект задается двумя числовыми значениями и может интерпретироваться как точка на плоскости (рис. 12.7). Цель — получить линейную разделяющую функцию, которая дает положительные значения для точек 1, 2 и 3 и принимает отрицательные значения для точек 4, 5, 6. Функция должна иметь вид $F(X) = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2$.

Выполняем *итерацию 0*. Коэффициенты $w_0 = w_1 = w_2 = 0$.

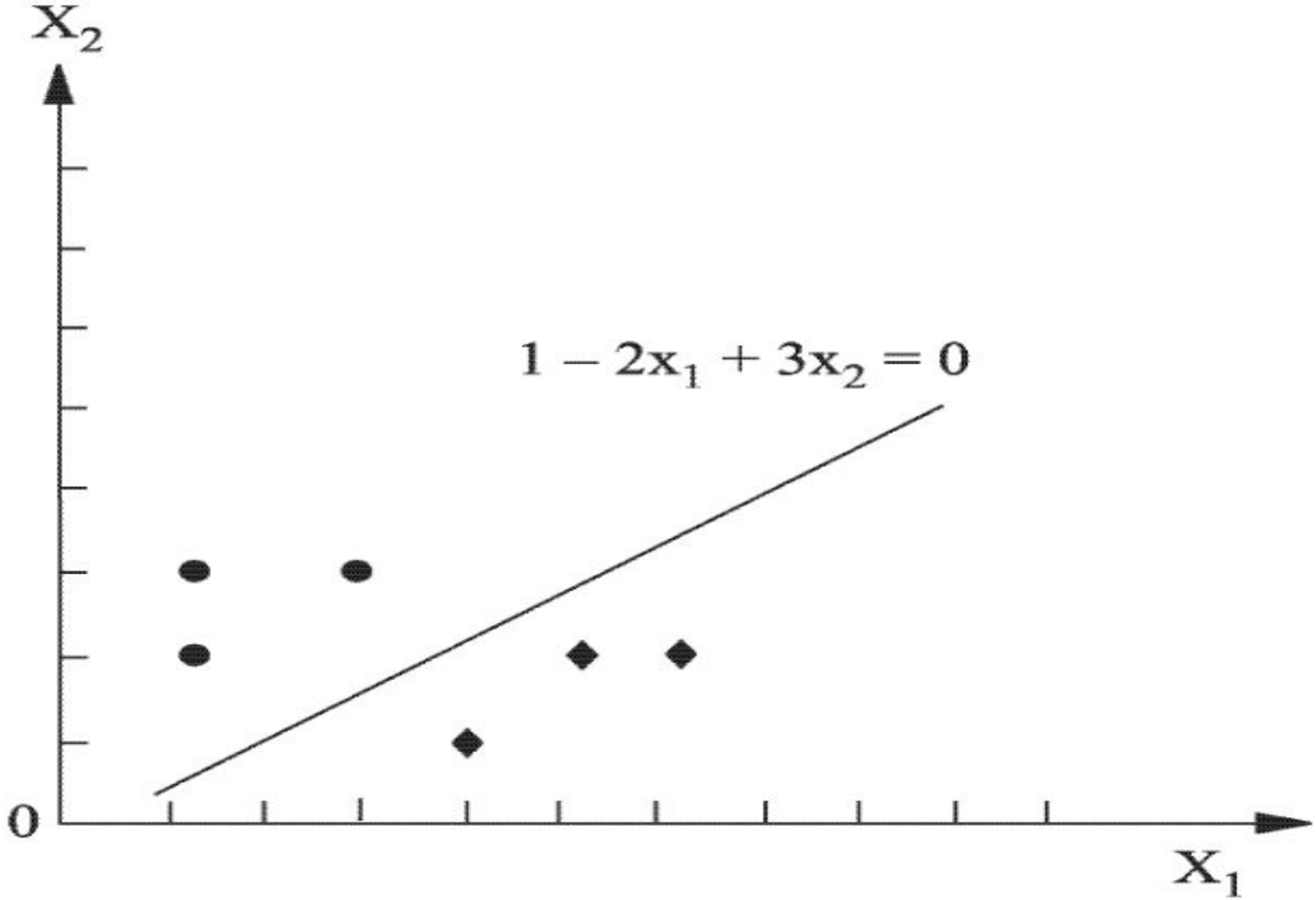

Рис. 12.7.

Выполняем *итерацию 1*. Выбираем первый объект класса C_1 – вектор $X_1 =< 1,2 >$. Значение функции $F(X_1) = 0 + 0 \cdot 1 + 0 \cdot 2 = 0$, по правилу П8 необходима коррекция коэффициентов при значении множителя $c = 1$. Вычисляем новые коэффициенты функции: $w_0^1 = w_0 + c = 0 + 1 = 1$; $w_1^1 = w_1 + c \cdot x_1 = 0 + 1 \cdot 1 = 1$; $w_2^1 = w_2 + c \cdot x_2 = 0 + 1 \cdot 2 = 2$. Получаем $F^1(X) = 1 + x_1 + 2 \cdot x_2$.

Выполняем *итерацию 2*. Вычислим последовательно значения $F^1(X)$ для элементов выборки: $F^1(<1,2>) = 1 + 1 \cdot 1 + 2 \cdot 2 = 6 > 0$; $F^1(<1,3>) = 1 + 1 \cdot 1 + 2 \cdot 3 = 7 > 0$; $F^1(<3,3>) = 13 > 0$. Все элементы класса C_1 распознаны правильно. Выбираем текущим класс C_2. $F^1(<4,1>) = 1 + 1 \cdot 4 + 2 \cdot 1 = 7 > 0$ — объект распознан неправильно. Необходима коррекция коэффициентов при значении множителя $c = -1$. $w_0^2 = w_0^1 + c = 1 - 1 = 0$; $w_1^2 = w_1^1 + c \cdot x_1 = 1 - 1 \cdot 4 = -3$; $w_2^2 = w_2^1 + c \cdot x_2 = 2 - 1 \cdot 1 = 1$. Новая функция $F^2(X) = x_2 - 3 \cdot x_1$.

Выполняем *итерацию 3*. Вычисляем значения функции для элементов выборки $F^2(<1,2>) = 2 - 3 \cdot 1 = -1 < 0$. Необходима коррекция коэффициентов с поправкой $c = 1$: $w_0^3 = w_0^2 + c = 0 + 1 = = 1$; $\quad w_1^3 = w_1^2 + c \cdot x_1 = -3 + 1 \cdot 1 = -2$; $\quad w_2^3 = w_2^2 + c \cdot x_2 = 1 + 1 \cdot 2 = 3$. Новая функция $F^3(X) = 1 - 2x_1 + 3x_2$.

Начинаем *новую итерацию* с проверки значений $F^3(X)$ на элементах выборки: $F^3(< 1,2 >) = 5 > 0$; $F^3(< 1,3 >) = 8 > 0$; $F^3(< 3,3 >) = 4 > 0$. Переходим к проверке объектов класса C_2: $F^3(< 4,1 >) = -4 < 0$; $F^3(< 5,2 >) = -3 < 0$; $F^3(< 6,2 >) = -5 < 0$. Все объекты выборки разделены правильно, таким образом получена искомая решающая функция $F(X) = 1 - 2x_1 + 3x_2$. На рис. 12.7 дана геометрическая интерпретация решения.

Начинаем *новую итерацию* с проверки значений $F^3(X)$ на элементах выборки: $F^3(< 1,2 >) = 5 > 0$; $F^3(< 1,3 >) = 8 > 0$; $F^3(< 3,3 >) = 4 > 0$. Переходим к проверке объектов класса C_2: $F^3(< 4,1 >) = -4 < 0$; $F^3(< 5,2 >) = -3 < 0$; $F^3(< 6,2 >) = -5 < 0$. Все объекты выборки разделены правильно, таким образом получена искомая решающая функция $F(X) = 1 - 2x_1 + 3x_2$. На рис. 12.7 дана геометрическая интерпретация решения.

12.4.4. Построение решающих функций методом потенциалов

Иногда точки обучающей выборки невозможно разделить, используя линейные функции. На рис. 12.8 приведен пример такой обучающей выборки. В случае, если классы невозможно разделить линейно, используется метод потенциалов, предназначенный для построения нелинейных решающих функций.

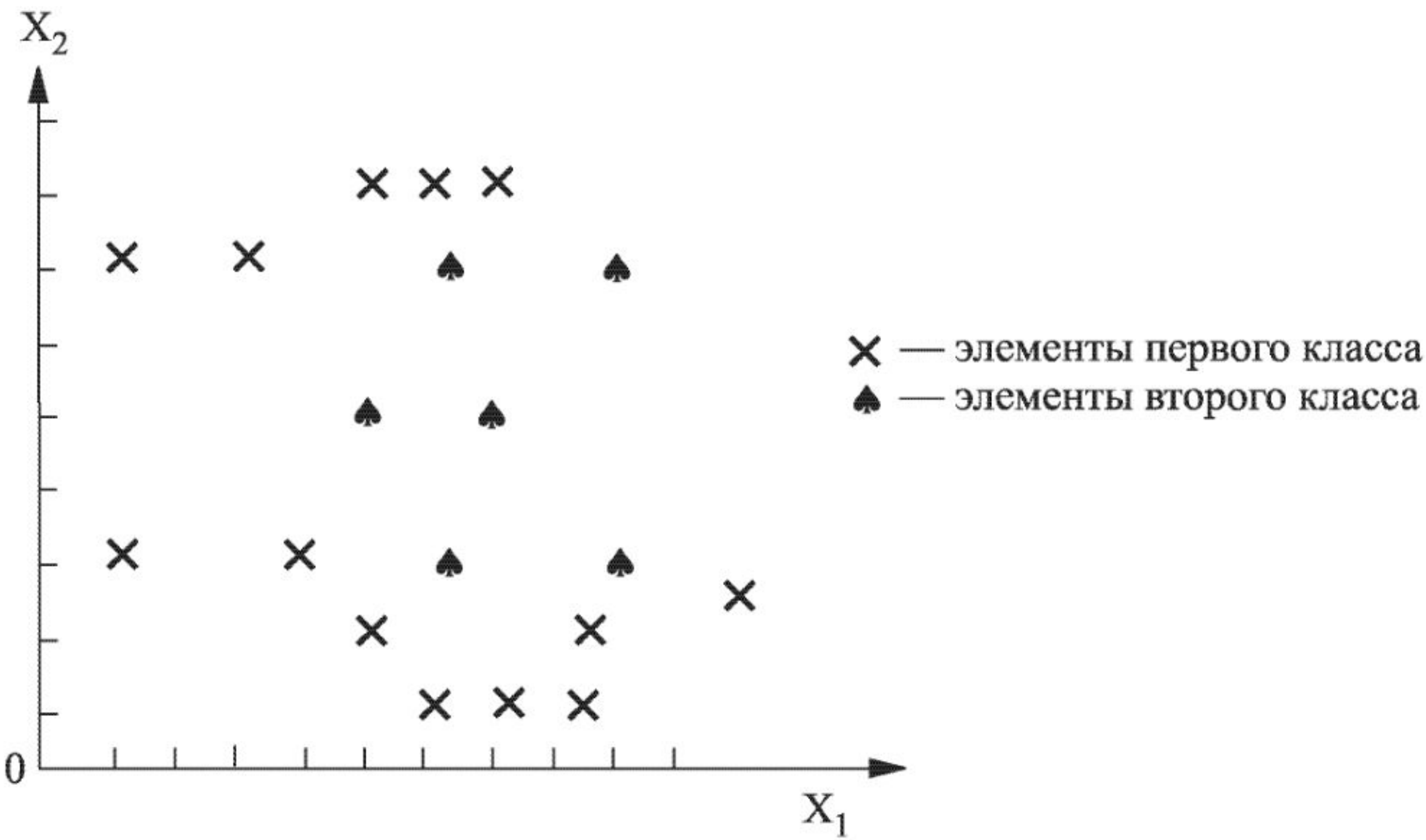

Рис. 12.8.

Само название «метод потенциалов» связано со следующей интерпретацией. Предположим, что любой точке из обучающего множества соответствует некоторый объект, обладающий зарядом (наподобие электрического). Объекты разных классов имеют заряды разных знаков: положительные и отрицательные. В любой точке n-мерного пространства признаков такие заряды создадут некоторый потенциал, являющийся суммой потенциалов отдельных зарядов. Величина потенциала прямо пропорциональна величине заряда и обратно пропорциональна расстоянию до него.

Известно, что линии, соединяющие точки с одинаковыми потенциалами, называются эквипотенциалями. В таких условиях каждый класс отделен от другого «потенциальной долиной», имеющей нулевой потенциал. Такая нулевая эквипотенциаль представляет собой границу между классами; ее уравнение и будет решающей функцией.

Функция, описывающая закон изменения потенциала, создаваемого зарядом, помещенным в точку X_c, может быть задана по-разному. Рассмотрим два варианта:

1) $F(X, X_c) = \exp(-a \cdot |X - X_c|^2)$;

2) $F(X, X_c) = \dfrac{1}{1 + a \cdot |X - X_c|^2}$, где $a > 0$ — константа (можно принять $a = 1$).

Если рассматривать расстояние по Евклиду, вторая функция примет вид

$$F(X, X_c) = \frac{1}{1 + \sum\limits_{j=1}^{n} (x_j - x_{cj})^2}, \text{ где коэффициент } a = 1.$$

Рассмотрим алгоритм построения решающей функции как разделяющей границы $D(X)$, отделяющей области положительного и отрицательного потенциалов. При описании алгоритма используем обозначения, принятые в описании алгоритма построения линейной разделяющей функции. Легко заметить, что оба алгоритма имеют много общего.

Алгоритм 12.5.

1. Получить обучающую выборку $\tilde{X} = (X_1, X_2, \ldots, X_M)$, элементы которой принадлежат классам C_1 или C_2.
2. Установить в ноль счетчик правильно распознанных объектов: $сч = 0$.
3. Установить номер итерации равным нулю: $k = 0$.
4. Начальное значение решающей функции $D^0(X) = 0$.

5. Выбрать в качестве текущего класса класс C_1.

6. Выполнение новой итерации: $k = k + 1$. Если $k > M$, примем $k = 1$.

7. Выбрать очередной объект X_k из текущего класса. Если текущий класс исчерпан, то объявить текущим класс C_2 и выбрать из него первый объект.

8. Построить новую решающую функцию:

$$D^k(X) = D^{k-1}(X) + c \cdot F(X, X_k),$$

где множитель c выбирается из условий

$$c = \begin{cases} 1, \text{ если } D^{k-1}(X_k) < 0 \text{ и } X_k \in C_1; \\ -1, \text{ если } D^{k-1}(X_k) \geqslant 0 \text{ и } X_k \in C_2; \\ 0 \textit{ при правильном распознавании.} \end{cases}$$

9. Если $c = 0$, то $сч = сч + 1$, иначе $сч = 0$.
10. Если $сч = M$, то КОНЕЦ, иначе перейти к шагу 6.

Заметим, что при выполнении пункта 8 на первой итерации всегда получим значение множителя с = 1, так как $D^0(X_1) = 0,\ X_1 \in C_1$.

Пример 12.5

Пример работы алгоритма рассмотрим для обучающей выборки, представленной в табл. 12.11.

Таблица 12.11

	Точки X	Координаты точек
Класс C_1	1	$<2; 4>$
	2	$<4; 1>$
	3	$<4; 6>$
Класс C_2	4	$<4; 2>$
	5	$<6; 2>$
	6	$<6; 5>$

Как следует из рис. 12.9, между точками классов нельзя провести линейную границу.

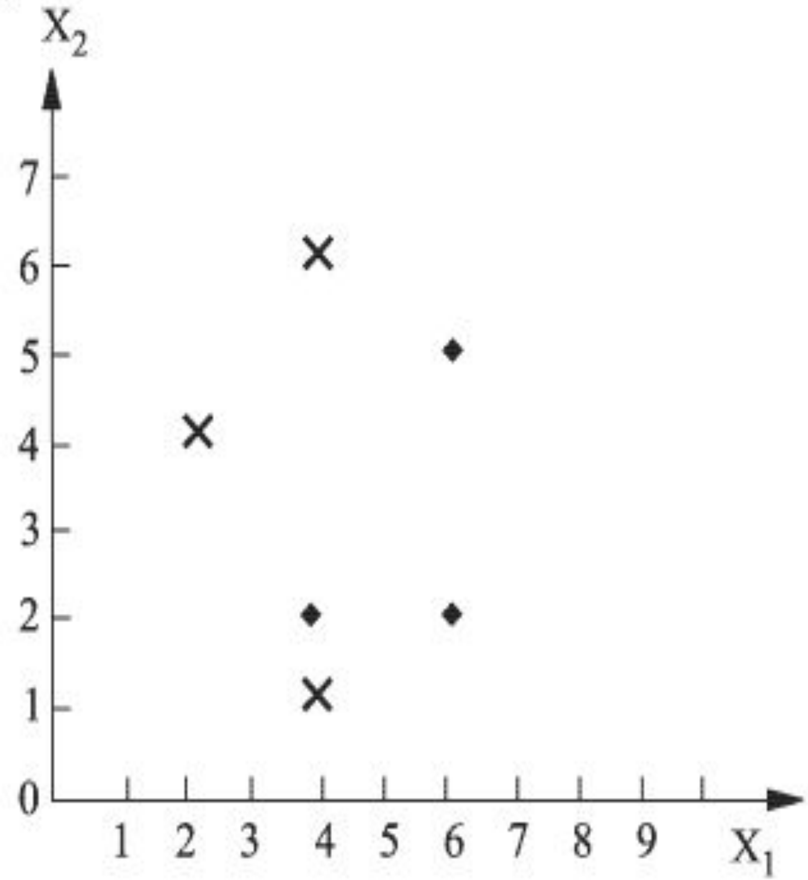

Рис. 12.9.

Воспользуемся методом потенциалов, причем для вычисления нелинейной границы предлагается использовать функцию второго типа. Поскольку каждая точка задана двумя координатами, функция примет вид

$$F(X, X_c) = \frac{1}{1 + (x_1 - x_{1c})^2 + (x_2 - x_{2c})^2}.$$

Рассмотрим выполнение алгоритма по шагам.

Итерация 0. Искомая функция $D^0(X) = 0$. Проверяем элементы обучающей выборки. $D^0(< 2, 4 >) = 0$, необходима коррекция при $c = 1$.

Итерация 1. Строим решающую функцию $D^1(X) = 1/(1 + (x_1 - 2)^2 + (x_2 - 4)^2)$. Проверим элементы обучающей выборки для класса C_1:

$$D^1(<2,4>) = 1 > 0; \quad D^1(<4,1>) = \frac{1}{1+4+9} = \frac{1}{14} > 0;$$

$$D^1(<4,6>) = \frac{1}{9} > 0.$$

Для элементов класса C_2: $D^1(<4,2>) = \dfrac{1}{9} > 0$. Распознавание ошибочно, поэтому выбираем множитель $c = -1$, и начинаем коррекцию функции.

Итерация 2.

$$D^2(X) = \frac{1}{1+(x_1-2)^2+(x_2-4)^2} - \frac{1}{1+(x_1-4)^2+(x_2-2)^2}.$$

Проверяем элементы обучающей выборки:

$$D^2(<2,4>) = 1 - \frac{1}{9} > 0; \quad D^2(<4,1>) = \frac{1}{14} - \frac{1}{2} < 0.$$

Необходима коррекция при $c = 1$.

Итерация 3. Новая функция примет вид

$$D^3(X) = \frac{1}{1+(x_1-2)^2+(x_2-4)^2} - \frac{1}{1+(x_1-4)^2+(x_2-2)^2} + \\ + \frac{1}{1+(x_1-4)^2+(x_2-1)^2}.$$

Выполним проверку для элементов класса C_1:

$$D^3(<2,4>) = 1 - \frac{1}{9} + \frac{1}{14} > 0; \quad D^3(<4,1>) = \frac{1}{14} - \frac{1}{2} + 1 > 0;$$

$$D^3(<4,6>) = \frac{1}{9} - \frac{1}{17} + \frac{1}{26} > 0.$$

Проверяем элементы класса C_2:

$$D^3(<4,2>) = \frac{1}{9} - 1 + \frac{1}{2} < 0; \quad D^3(<6,2>) = \frac{1}{21} - \frac{1}{5} + \frac{1}{6} > 0.$$

Необходима коррекция функции при значении $c = -1$.

Итерация 4. Новая функция содержит четыре компонента:

$$D^4(X) = \frac{1}{1+(x_1-2)^2+(x_2-4)^2} - \frac{1}{1+(x_1-4)^2+(x_2-2)^2} + \frac{1}{1+(x_1-4)^2+(x_2-1)^2} - \frac{1}{1+(x_1-6)^2+(x_2-2)^2}.$$

Выполним проверку для элементов класса C_1:

$$D^4(<2,4>) = 1 - \frac{1}{9} + \frac{1}{14} - \frac{1}{21} > 0; \quad D^4(<4,1>) = \frac{1}{14} - \frac{1}{2} + 1 - \frac{1}{6} > 0;$$

$$D^4(<4,6>) = \frac{1}{9} - \frac{1}{17} + \frac{1}{26} - \frac{1}{21} > 0.$$

Переходим к проверке элементов класса C_2:

$$D^4(<4,2>) = \frac{1}{9} - 1 + \frac{1}{2} - \frac{1}{5} < 0; \quad D^4(<6,2>) = \frac{1}{21} - \frac{1}{5} + \frac{1}{6} - 1 < 0;$$

$$D^4(<6,5>) = \frac{1}{18} - \frac{1}{14} + \frac{1}{21} - \frac{1}{10} < 0.$$

Все объекты обучающей выборки функция $D^4(X)$ разделила правильно, следовательно, она является искомой решающей функцией. Работа алгоритма завершена. На рис. 12.10 представлена графическая интерпретация решения.

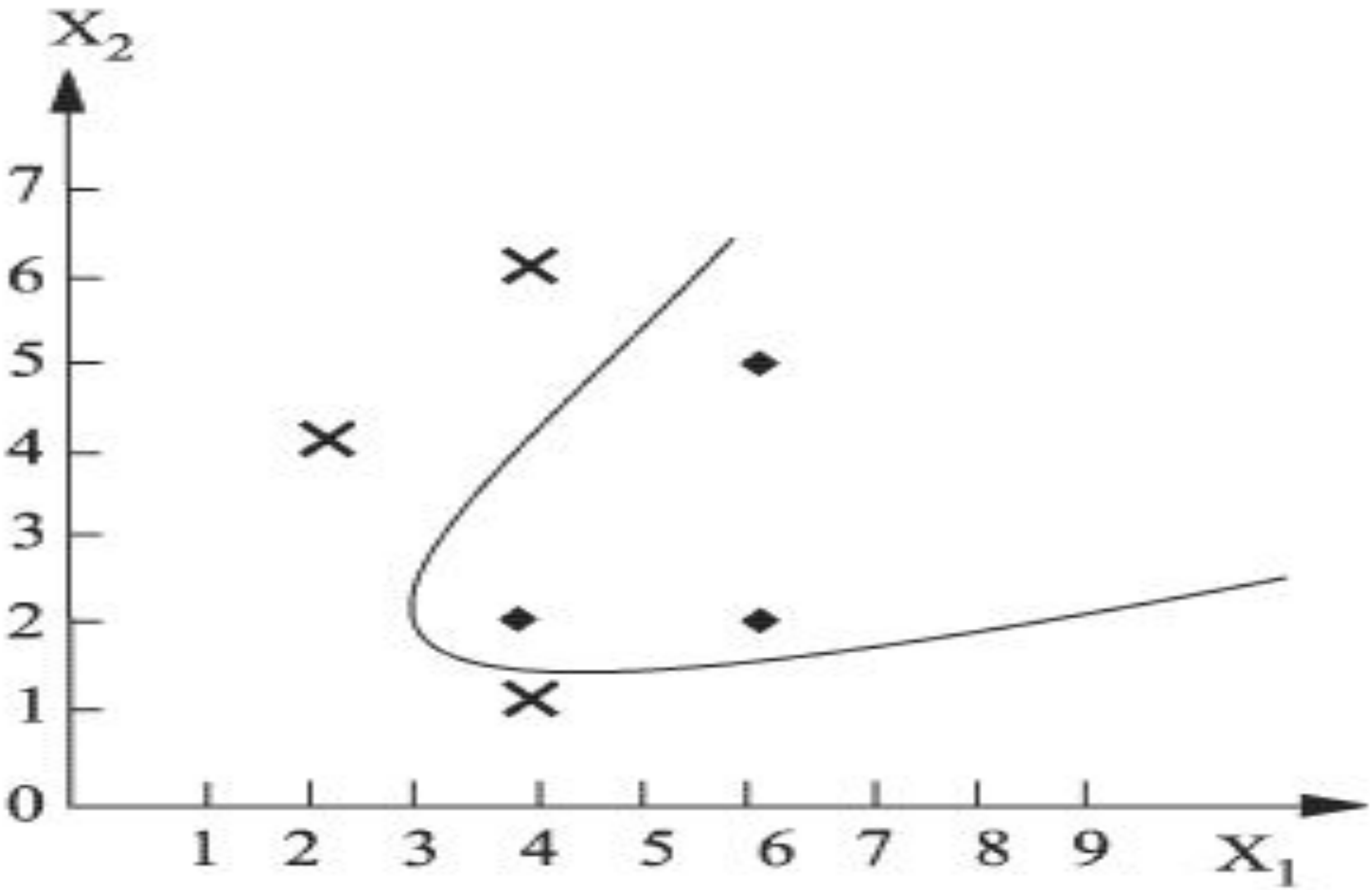

Рис. 12.10.

12.5. Распознавание на основе приближенных признаков

Значения, которые принимают признаки в обучающих выборках, часто являются результатами измерений различных параметров. Любые измерения связаны с возможностью возникновения ошибок, таким образом числовые значения признаков мы получаем с некоторой достоверностью, или уверенностью в их истинности. В ситуациях, когда вероятность получения неверных значений достаточно велика, введем в рассмотрение риск, связанный с решением об отнесении объекта к некоторому классу. Под риском будем понимать «цену», связанную с правильным или неправильным распознаванием объекта. Например, риском может служить величина потерь, связанная с тем, что робот неправильно обработает деталь, ошибочно отнеся ее к другому классу.

Введем матрицу рисков, связанных с правильными и ошибочными решениями:

$$R = \begin{vmatrix} R_{11} & R_{12} & \cdots & R_{1M} \\ R_{21} & R_{22} & \cdots & R_{2M} \\ \dots & \dots & \dots & \dots \\ R_{M1} & R_{M2} & \cdots & R_{MM} \end{vmatrix},$$

где R_{ij} – риск, связанный с отнесением к классу j объекта, в действительности принадлежащего классу i, M – общее число классов.

В матрице имеются диагональные элементы – R_{ii}. Они отображают риск, связанный с правильным решением (обычно $R_{ii} \leqslant 0$, и эту величину можно рассматривать как выигрыш, достигнутый за счет правильного распознавания).

Пусть для распознавания предъявлен объект, у которого известны значения признаков $X = \langle x_1, x_2, \ldots, x_n \rangle$. Назовем это событием A. Значение риска, связанное с решением о принадлежности объекта X классу C_j при условии, что имеет место событие A, выражается формулой

$$R(X \in C_j|A) = R(C_j|A) = \sum_{i=1}^{M} R_{ij} \cdot P(C_i|A).$$

где $P(C_i)$ — априорная вероятность появления объекта, принадлежащего классу i, она может быть известна, например, из статистических данных; $P(A|C_i)$ — вероятность появления признаков $X = \langle x_1, x_2, \ldots, x_n \rangle$ у объекта, принадлежащего классу i.

Решение о том, что объект X принадлежит классу C_z, принимается в случае, если риск, связанный с данным решением, минимален: $R(C_z|A) = \min\limits_i R(C_i|A)$.

Введенное правило распознавания называется *критерием Байеса*. Применение этого правила обеспечивает минимальный средний риск, то есть риск, усредненный по всем возможным решениям задачи распознавания.

ОБУЧЕНИЕ С УЧИТЕЛЕМ

Алгоритмы, исследованные в предыдущей главе, решали задачи классификации и распознавания для числовых признаков. Качественные, а также шкалированные признаки не использовались, потому что для подобных признаков сложно ввести меру, задающую близость объектов. При исследовании объектов, содержащих качественные признаки, необходимо определять близость или сходство объектов прежде всего на основании совпадения качественных значений. Важным моментом является определение правильных сочетаний значений некоторых качественных признаков. Поиск наиболее существенных сочетаний признаков удобно проводить с помощью аппарата логических функций. Алгоритмы, использующие такой подход, относятся к алгоритмам обучения с учителем и будут рассмотрены в настоящей главе.

13.1. Постановка задачи

Поставим задачу обобщения понятий по признакам. Пусть имеется множество объектов, состоящее из положительных и отрицательных примеров формируемых понятий. Назовем такое множество обучающей выборкой. На основании обучающей выборки необходимо построить понятие, разделяющее положительные и отрицательные объекты.

Под обобщением, как правило, понимается переход от рассмотрения единичного объекта o или некоторой совокупности объектов T к рассмотрению множества объектов V такого, что $o \in V$ или $T \subset V$. Здесь в качестве объекта может рассматриваться как реальный объект, так и модель некоторого процесса или явления.

Пусть $O = \{o_1, o_2, \ldots, o_n\}$ — множество объектов, которые могут быть представлены в интеллектуальной системе S. Каждый объект характеризуется r признаками. Обозначим через $Z_1, Z_2, \ldots, Z_r$ множество допустимых признаков, где $Z_k = \{z_{k1}, z_{k2}, \ldots, z_{km}\}$ $(1 \leqslant k \leqslant r)$ и z_{ki} являются значениями признаков. Каждый объект $o_i \in O$, $1 \leqslant i \leqslant n$, представляется как множество значений признаков, т.е. $o_i = \{z_{kj}\}$, где $z_{kj} \in Z_k$, $1 \leqslant k \leqslant r$, $1 \leqslant j \leqslant m$. Такое описание объекта называется *признаковым описанием*. В качестве признаков объектов могут использоваться количественные, качественные, либо шкалированные признаки.

В основе процесса обобщения лежит сравнение описаний исходных объектов, заданных совокупностью значений признаков, и выделение наиболее характерных фрагментов этих описаний. В зависимости от того, входит или не входит объект в объем некоторого понятия, назовем его *положительным* или *отрицательным* объектом для этого понятия.

Пусть O – множество всех объектов, представленных в некоторой системе знаний, V – множество положительных объектов и W – множество отрицательных объектов. Будем рассматривать случай, когда $O = V \cup W$, $V \cap W = \emptyset$, $W = \bigcup_i W_i$ и $W_i \cap W_j = \emptyset$ $(i \neq j)$.

Пусть K – непустое множество объектов такое, что $K = K^+ \cup K^-$, где $K^+ \subset V$ и $K^- \subset W$. Будем называть K обучающей выборкой. На основании обучающей выборки надо построить правило, разделяющее положительные и отрицательные объекты обучающей выборки.

Понятие, таким образом, сформировано, если удалось построить решающее правило, которое для любого примера из обучающей выборки указывает, принадлежит ли этот пример понятию или нет. Алгоритмы, которые мы исследуем, формируют решение в виде правил типа «ЕСЛИ условие, ТО искомое понятие». Условие представляется в виде логической функции, в которой булевы переменные, отражающие значения признаков, соединены логическими операциями конъюнкции, дизъюнкции и отрицания. Решающее правило является корректным, если оно в дальнейшем успешно распознает объекты, не вошедшие первоначально в обучающую выборку.

Результатом обучения будет некоторое решающее правило. Оно должно позволять принять решение об отнесении конкретных примеров к понятию. Правило считается корректным, если оно успешно разделяет все примеры и контрпримеры обучающей выборки.

После того, как распознающее правило на обучающей выборке построено, проводится экзамен – с помощью распознающего правила надо разделить объекты новой, экзаменационной выборки на примеры и контрпримеры. Если решающее правило правильно проводит такое разделение, обучение заканчивается. Если результат экзамена неудовлетворителен, то можно проводить дополнительное обучение на новой обучающей выборке (например, к исходной обучающей выборке можно добавить примеры, на которых при распознавании возникали ошибки).

В отличие от алгоритмов обучения без учителя основное внимание будем уделять качественным признакам. Главная задача алгоритмов обобщения заключается в построении решающего правила, которое в дальнейшем используется для распознавания новых объектов. Критерии качества построенных правил — это компактность и высокое качество распознавания вновь предъявленных объектов.

Прежде чем перейти к рассмотрению реальных алгоритмов обучения с учителем, исследуем способы задания решающего правила на примере.

Пример 13.1.

В табл. 13.1 приведен пример обучающего множества. Здесь каждый объект имеет четыре атрибута: *класс*, *рост*, *цвет волос* и *цвет глаз*.

Таблица 13.1

Класс	Рост	Волосы	Глаза
−	Низкий	Светлые	Карие
−	Высокий	Темные	Карие
+	Высокий	Светлые	Голубые
−	Высокий	Темные	Голубые
−	Низкий	Темные	Голубые
−	Высокий	Светлые	Карие
+	Низкий	Светлые	Голубые

Наблюдая примеры, приведенные в таблице, сформулируем закономерность их разделения на классы: все голубоглазые объекты со светлыми волосами относятся к классу $+$; все темноволосые объекты, либо светловолосые, но с карими глазами, относятся к классу $-$. Заметим также, что признак *рост* не влияет на выбор класса. Рассмотрим способы описания полученных решений в автоматизированной системе.

Первый способ задания решающего правила — использование логических функций. Две функции, задающие классы $+$ и $-$, могут иметь вид:

$$P^{+}(X) = (\textit{глаза} = \textit{голубые})\ \&\ (\textit{волосы} = \textit{светлые})$$

$$P^{-}(X) = (\textit{волосы} = \textit{темные}) \vee$$

$$\vee (\textit{волосы} = \textit{светлые})\ \&\ (\textit{глаза} = \textit{карие}).$$

Дерево решений – это дерево, внутренние узлы которого представляют собой проверки для входных примеров из обучающего множества, а вершины-листы являются категориями, классами (примеров). Пример дерева решений приведен на рис. 13.1. Дерево решений ка-

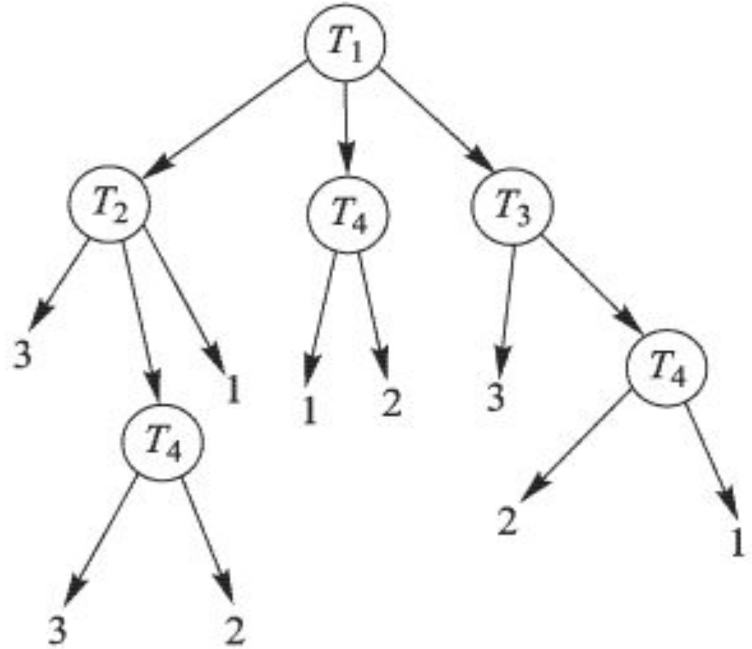

Рис. 13.1.

Пример дерева решений приведен на рис. 13.1. Дерево решений каждому входному примеру ставит в соответствие номер класса (или выходное значение) путем фильтрации этого примера через узлы проверки дерева сверху вниз. Результаты каждой проверки являются взаимоисключающими и исчерпывающими. Например, проверка T_2 в дереве, изображенном на рис. 13.1, имеет три возможных исхода: самый левый относит входной пример к классу 3, средний перенаправляет входной пример вниз к проверке T_4, а самый правый относит пример к классу 1. Мы будем следовать обычной договоренности об изображении узлов-листьев номерами классов.

Существует несколько характеристик, по которым деревья решений могли бы различаться:

1. Проверки могут быть *многопризнаковыми* (выполняется проверка нескольких признаков входного примера за один раз) или *однопризнаковыми* (выполняется проверка только одного признака).
2. Проверки могут приводить к двум результатам или более чем к двум. (Если все проверки приводят к двум результатам, то мы получаем двоичное дерево решений).

3. Признаки (или атрибуты), которые используются в узлах дерева, могут быть качественными или количественными. Бинарные признаки могут рассматриваться как любые из них.

4. Классов может быть два или более. Если мы имеем два класса, и объекты классификации представляют собой двоичные входные векторы, то дерево реализует булеву функцию и называется булевым деревом решений.

Удобно использовать дерево решений с пометками на ребрах. В нем вершины, не являющиеся концевыми, помечены именами атрибутов, ребра — допустимыми значениями соответствующего атрибута, и концевые вершины — именами классов. Процесс классификации заключается в прохождении пути из корня к листьям, следуя ребрам, соответствующим значениям атрибутов объекта.

На рис. 13.2 приведен пример дерева решений для обучающей выборки из табл. 13.1.

Рис. 13.2

Еще один способ представления решающего правила – продукции. Продукционные правила представляются в виде

если <посылка>, **то** <заключение>

В системах извлечения знаний в качестве посылки выступает описание объекта через его свойства, а заключением будет вывод о принадлежности объекта к определенному классу. Примером такого продукционного правила является

если pH < 6, **то** жидкость – кислота.

В экспертных системах часто используются правила, в которых посылкой является описание ситуации, а заключением — действия, которые необходимо предпринять в данной ситуации. Основные достоинства, благодаря которым продукционные правила получили широкое распространение, заключаются в следующем:

1) продукционные правила легки для восприятия человеком;
2) отдельные продукционные правила могут быть независимо добавлены в базу знаний, исключены или изменены, при этом не

требуется перепрограммирование всей системы. Как следствие этого, представление больших объемов знаний не вызывает затруднений;

3) с помощью продукционных правил выражаются как декларативные, так и процедурные знания.

Решающие деревья более сложны для понимания, чем продукционные правила, что является их недостатком. С другой стороны, любое решающее дерево может быть преобразовано в набор продукционных правил: каждому пути от корня дерева до концевой вершины соответствует одно продукционное правило. Его посылкой является конъюнкция условий «атрибут — значение», соответствующих пройденным вершинам и ребрам дерева, а заключением – имя класса, соответствующего концевой вершине. Так, приведенное выше решающее дерево может быть записано в виде следующего набора продукционных правил:

если *волосы* = *светлые* & *глаза* = *голубые* **то** *класс* = «+»

если *волосы* = *светлые* & *глаза* = *карие* **то** *класс* = «−»

если *волосы* = *темные* **то** *класс* = «−».

Эти наиболее популярные модели описания класса объектов равносильны. Перейдем к рассмотрению конкретных алгоритмов, использующих такие модели.

13.3. Построение решающего дерева с использованием метрики Хемминга

Дерево решений – это по сути алгоритм, представленный в специальной форме. Рассмотрим алгоритм, который строит бинарное дерево на основании поиска существенного (разделяющего) значения некоторого признака. Каждой вершине дерева приписывается вопрос «Обладает ли пример данным значением признака?», а дуги взвешиваются ответами (Да, Нет). Конечные (висячие) вершины дерева будут помечены утверждениями, соответствующими решениям.

Распознавание начинается с корневой вершины, откуда строится путь до конечной (висячей) вершины. В каждом внутреннем узле необходимо отвечать на вопрос и выбирать соответствующую дугу, ведущую к следующему вопросу. Заметим, что вершины с одинаковыми вопросами могут встречаться несколько раз в разных местах дерева, но в каждом конкретном пути от «корня» к «листу» каждый вопрос встретится только один раз.

Рассмотрим рекурсивный алгоритм построения бинарного дерева на основе обучающей выборки S (далее будем называть этот алгоритм AMX – «Алгоритм, основанный на метрике Хемминга»).

Для предъявленной обучающей выборки S ищется наиболее важный вопрос, который помещается в корневую вершину дерева. Например, при решении задачи выбора меню для обеда, в корневую вершину разумно поместить вопрос, является ли клиент вегетарианцем. Обучающее множество S в соответствии с ответами ДА, НЕТ разделяется на две подвыборки. Далее для каждой подвыборки снова ищется наиболее важный вопрос и порождаются две новые ветви. Процесс завершается, когда использованы все вопросы.

Для нахождения наиболее важного вопроса используется следующий подход. Предположим, обучающее множество S содержит K примеров. Построим метрику на множестве вопросов и упорядочим S по следующему правилу. Для каждого вопроса Q строится K-арный кортеж, где j-й элемент в кортеже имеет значение

$$j = \begin{cases} 1, \text{ если } j\text{-й пример из } S \text{ дает ответ ДА на вопрос } Q, \\ 0 \text{ в противном случае.} \end{cases}$$

Расстояние между двумя вопросами определяется как расстояние между двумя двоичными векторами по Хеммингу. Обозначим такое расстояние между векторами X и Y как $Hd(X,Y)$. При этом из рассмотрения исключаются вопросы, на которые все примеры из S дают одинаковый ответ (все 0 или все 1).

Для искомого свойства P и множества объектов S строится кортеж длины K по правилу: j-е значение в кортеже P равно 1, если j-й пример из S обладает искомым свойством P; j-е значение в кортеже P равно 0, если j-й пример из S этим свойством не обладает. Чтобы выбрать наиболее важный вопрос, ищем среди всех подходящих вопросов Q такой, чья связь с P наиболее тесная. Для этого воспользуемся правилом: ищем $pm = \min\{Hd(P,Q), K - Hd(P,Q)\}$, то есть берем наименьшее расстояние по Хеммингу от Q до P и от Q до $\neg P$ (pm далее называем псевдометрикой).

Когда наиболее важный вопрос выбран, выполняется расщепление обучающего множества на $S_{\text{да}}$ и $S_{\text{нет}}$, затем для каждого подмножества выполняются аналогичные вычисления.

Пример 13.3.

Рассмотрим пример построения дерева решений для анализа поведения Лондонского фондового рынка. Пусть главные факторы, влияющие на фондовый рынок, это

- состояние на вчерашний день;
- состояние фондового рынка в Нью-Йорке;
- банковские тарифы (ставки);
- процент безработицы;
- перспективы для Англии.

Таблица 13.2

	1	2	3	4	5	6	Hd	pm
Наблюдается ли сегодня рост на фондовом рынке?	да	да	да	нет	нет	нет		
Наблюдался ли рост вчера?	да	да	нет	да	нет	нет	2	2
Наблюдается ли сегодня рост в Нью-Йорке?	да	нет	нет	нет	нет	нет	2	2
Растут ли банковские ставки?	нет	да	нет	да	нет	да	4	2
Безработица растет?	нет	да	да	нет	нет	нет	**1**	**1**
Несет ли Англия убытки?	да	да	да	да	да	да	3	3

В табл. 13.2 приведено обучающее множество из шести примеров.

Пусть наша цель — построить дерево, определяющее, наблюдается ли рост на Лондонском фондовом рынке. Искомое свойство «Сегодня наблюдается рост». Ищем расстояние по Хеммингу (Hd) между этим свойством и каждым атрибутом. Так, для атрибута «Банковские ставки растут» расстояние по Хеммингу $Hd(1\ 1\ 1\ 0\ 0\ 0,\ 0\ 1\ 0\ 1\ 0\ 1) = 4$ (значения различны в 4-х позициях). Тогда $Hd = 4$, $K - Hd = 6 - 4 =$ $= 2$, $\min(4,2) = 2$. Следовательно, pm («Сегодня наблюдается рост», «Банковские ставки растут») $= 2$.

Поскольку критерием выбора является минимальное значение pm, выбираем наиболее важный вопрос «Безработица растет?». Проведем расщепление S: $S_{да}$ включает примеры 2 и 3. В дереве эта вершина будет конечной (для всех примеров множества наблюдается рост на фондовом рынке). $S_{нет}$ содержит примеры 1, 4, 5, 6. Результаты повторных вычислений приведены ниже в табл. 13.3.

Таблица 13.3

	1	4	5	6	Hd	pm
Наблюдается ли сегодня рост на фондовом рынке?	да	нет	нет	нет		
Наблюдался ли рост вчера?	да	да	нет	нет	1	1
Наблюдается ли сегодня рост в Нью-Йорке?	да	нет	нет	нет	**0**	**0**
Растут ли банковские ставки?	нет	да	нет	да	3	1
Безработица растет?	нет	нет	нет	нет	1	1
Несет ли Англия убытки?	да	да	да	да	3	1

Из результата видно, что когда безработица не растет, Лондонский фондовый рынок в точности следует за Нью-Йоркским. Приведем ниже окончательное дерево решений (рис. 13.4).

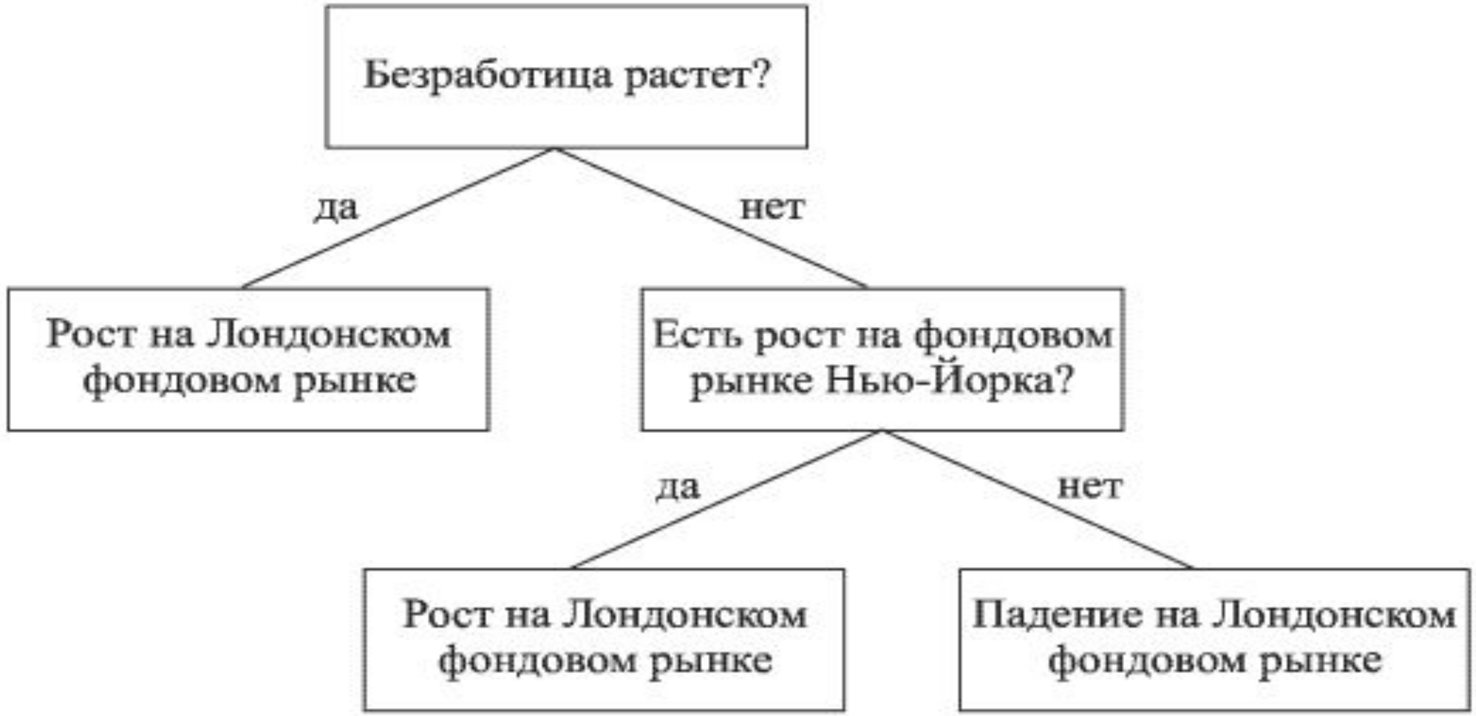

Рис. 13.4.