1.3. Неметрические подходы к задаче классификации

1. Поиск логических закономерностей

Идея: моделирование принятия решения человеком: на основании логики, рассуждений, обобщения фактов

Вопросы:

- Что такое закономерность?
- Как найти закономерность в данных?
- Как классифицировать новые данные?

Постановка задачи

Пусть объект o описывается переменными $X_1,...,X_j,...,X_n$; D_j область определения X_j.

Типы переменных:

1. Категориальные: $D_j = \{b_1,\ldots,b_{l_j}\}$ – некоторое множество символов, $l_j \geq 2$.
2. Вещественные: $D_j = \mathbf{R}$ это множество вещественных чисел.

Y – целевая категориальная переменная $D_Y = \{\omega_1,...,\omega_K\}$;

Набор данные $\{x_j^{(i)}, y^{(i)}\}$, $i = 1,...,N$.

Необходимо:

- найти логическую закономерность, описывающую зависимость между X and Y;
- использовать ее для предсказания Y для новых объектов.

Рассмотрим простую форму для логического утверждения об объекте o:

Конъюнкция

$$S(o) = J(o, E_{j_1}) \& \ldots \& J(o, E_{j_m}),$$

где $J(o, E_{j_i}) \in \{false, true\}$ это предикат,

$$J(o, E_j) \approx "X_j(o) \in E_j"$$

E_j это интервал $[a_j, b_j] \subset D_j$ для вещественной X_j или любое подмножество для категориальной X_j.

Рассмотрим область

$$E = E_1 \times \ldots E_j \times \ldots \times E_n,$$

где $E_j = D_j$ for $j \notin J$. Тогда утверждение

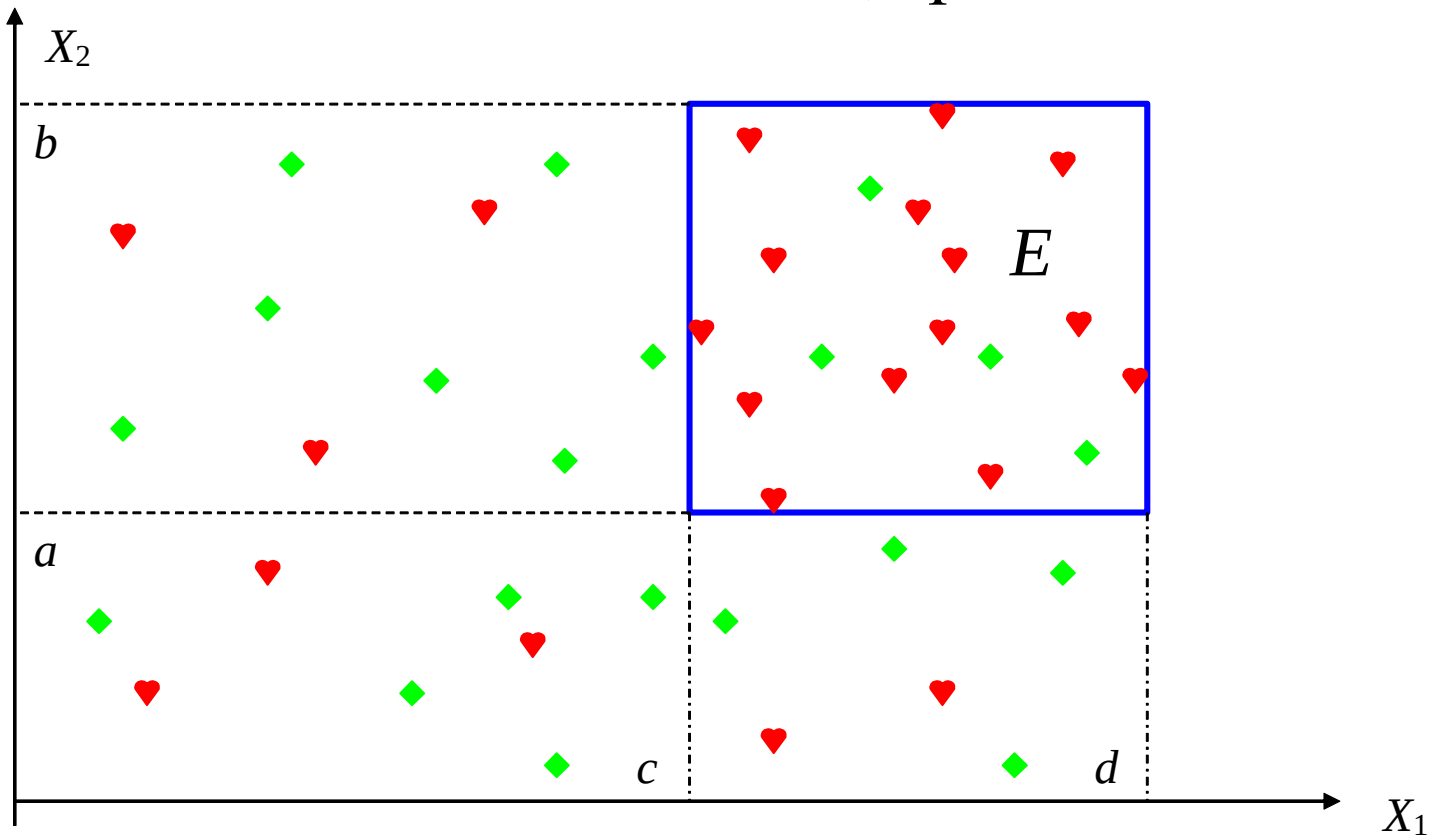
$$S(o) = S(o, E)$$

эквивалентно утверждению "$X(o) \in E$".

Логическое утверждение S о классе ω:

$$S = "Если\ x \in E\ ,\ тогда\ Y = \omega"$$

$\Leftrightarrow$ <область $E \subset D$, решение $Y = \omega$>



$$S = "ЕСЛИ\ X_1 \in [c,d]\ \ И\ \ X_2 \in [a,b],\ ТОГДА\ Y = \omega_1"$$

Эвристическое определение закономерности:

Утверждение S называется логической закономерностью, которая характеризует класс ω, если выполняются неравенства

$$\frac{N(\omega, S)}{N(\omega)} \geq \delta, \quad \frac{N(\bar{\omega}, S)}{N(\bar{\omega})} \leq \beta,$$

где $N(\omega, S)$ это число наблюдений класса ω, для которых выполняется утверждение S,
$N(\omega)$ - общее число объектов класса ω,
$N(\bar{\omega}, S)$ - число объектов других классов для которых верно утверждение S,
$N(\bar{\omega})$ - общее число наблюдений других классов,
δ и β это некоторые параметры, $0 \leq \beta < \delta \leq 1$.

Чем больше δ и меньше β, тем более “мощная” закономерность.

Алгоритм ТЕМР (Г.С. Лбов, 1976)

Пусть класс ω фиксирован. обозначим множество всех логических закономерностей как S^*.

Конъюнкция $S(o,E)$ называется *потенциальной логической закономерностью*, если выполняются следующие неравенства:

$$\frac{N(\omega,S)}{N(\omega)} \geq \delta, \quad \frac{N(\bar{\omega},S)}{N(\bar{\omega})} > \beta.$$

Обозначим через S' множество потенциальных логических закономерностей.

Если для некоторой конъюнкции $S(a,E)$ выполняется неравенство

$$\frac{N(\omega,S)}{N(\omega)} < \delta$$

то S не является логической закономерностью и не может ей стать, сколько бы предикатов мы не добавляли к ней. Обозначим их через $\bar{S}$.

Любая конъюнкция $S(a,E)$ принадлежит одному из множеств: S^*, S' and $\bar{S}$.

Шаг 1. Рассмотрим все возможные конъюнкции длины 1:

$$S(a,E) = J(a,E_j),\ j = 1,\ldots,n.$$

- Если $S(a,E) \in S^*$, тогда конъюнкция включается в список найденных логических закономерностей и область E_j исключается из дальнейшего рассмотрения;

- Если $S(a,E) \in S'$, тогда E_j остается для дальнейшего поиска;

- Если $S(a,E) \in \overline{S}$ тогда E_j исключается из дальнейшего поиска.

Пусть W_1 это множество областей E_j оставшихся после шага 1.

Шаг 2. Рассматриваются все возможные конъюнкции длины 2:

$$S(o,E) = J(o,E_i)\,\&\,J(o,E_j),\ i \neq j;\ E_i \in W_1\ ,\ E_j \in W_1\ .$$

Если $S \in S^*$, тогда она присоединяется к списку найденных закономерностей.

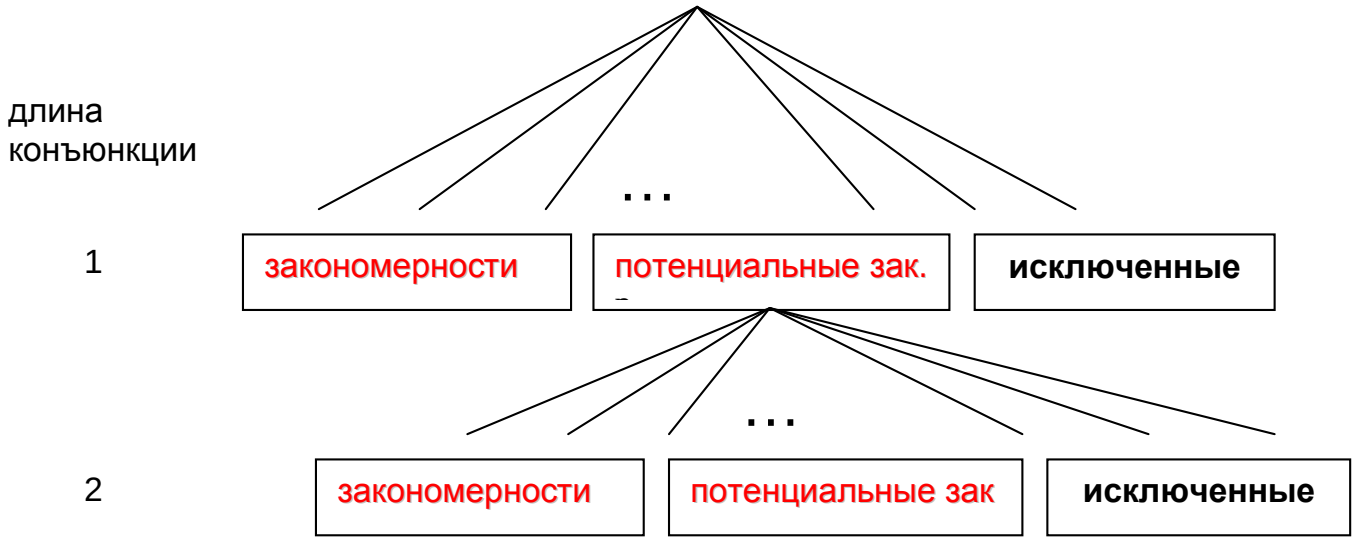
Формируется множество $W_{1,2}$ потенциальных логических закономерностей. Во время этого процесса, если $S \in S^*$ или $S \in \overline{S}$, то E_i, E_j исключаются из $W_{1,2}$.

Step 3. Рассматриваются все возможные конъюнкции длины 3:

$$S(o,E) = J(o,E_i)\,\&\,J(o,E_j)\,\&\,J(o,E_k),$$

$$i \neq j \neq k,\ (E_i,E_j) \in W_{1,2},\ E_k \in W_1\ ,$$

Далее рассматриваются конъюнкции длины 4, 5 и т.д. подобным образом.



Алгоритм работает до тех пор, пока не будут найдены все логические закономерности или будет достигнута максимальная длина конъюнкции $M_{\max}$.

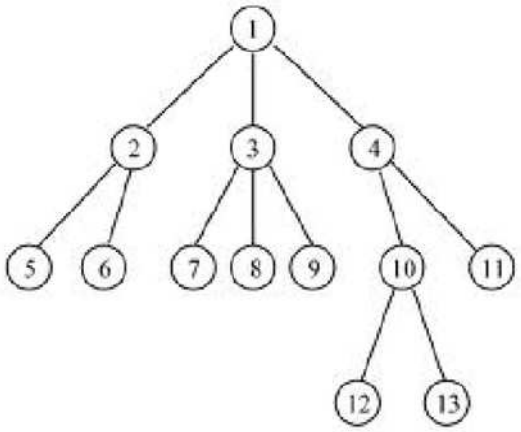
2. Деревья решений (Hoveland, Hunt ~ 1950)

Как классифицировать новые данные, используя логические закономерности?

Деревья решений - это удобная форма представления решающей функции в виде логических правил.

Дерево - это связный ненаправленный граф без циклов. Дерево может иметь корень - произвольную выбранную вершину.

Вершины: внутренние и терминальные (листья). Из каждой внутренней вершины выходят две или более ветви (ребра).



Дерево решения - это дерево, в котором

- каждый внутренний узел B соответствует признаку X_j;

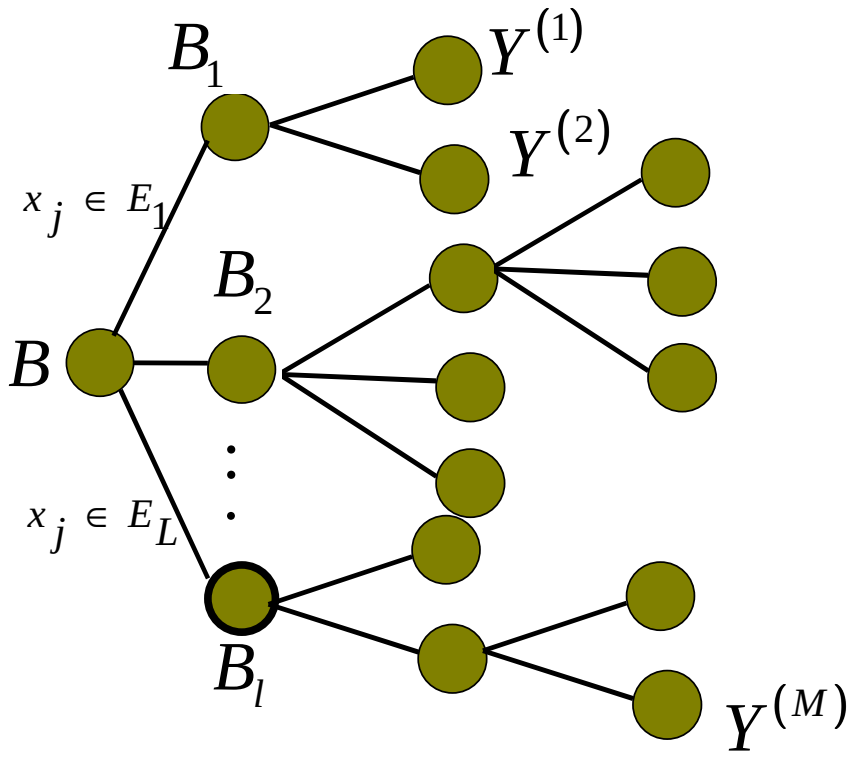
- каждая ветвь $b = (B, B_q)$, $q = 1,..., L$ представляет реализацию проверки условия "$x_j \in E_q$", где $E_q \subset D_j$, D_j это область определения X_j и $E_1,\ldots, E_L$ это разбиение D_j;

- E_q это:

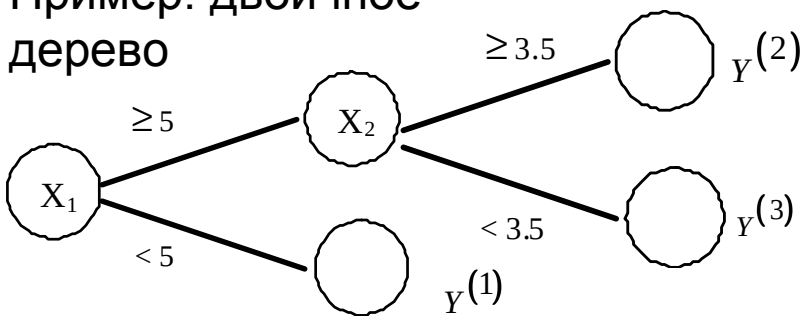
• в случае числовой переменной X_j: интервал [a,b), (-∞,a), [a,+ ∞);

• в случае категориальной переменной: множество значений;

- каждый лист представляет метку класса (решение) $Y^{(m)}$.


B_1
$Y^{(1)}$
$Y^{(2)}$
$x_j \in E_1$
B_2
B
$x_j \in E_L$
B_l
$Y^{(M)}$

Пример: двоичное дерево



Дерево решений с M листьями $\Rightarrow$ разделение пространства признаков на M непересекающихся областей $E^{(1)},\ldots,E^{(M)}$;

Каждый m-й лист соответствует области $E^{(m)}$.

Путь от корня до m-го листа $\Rightarrow$ логическое утверждение

"$IF\ x_{j_1} \in E_{j_1}\ AND \cdots AND\ x_{j_q} \in E_{j_q}$, $Then\ Y = Y^{(m)}$".

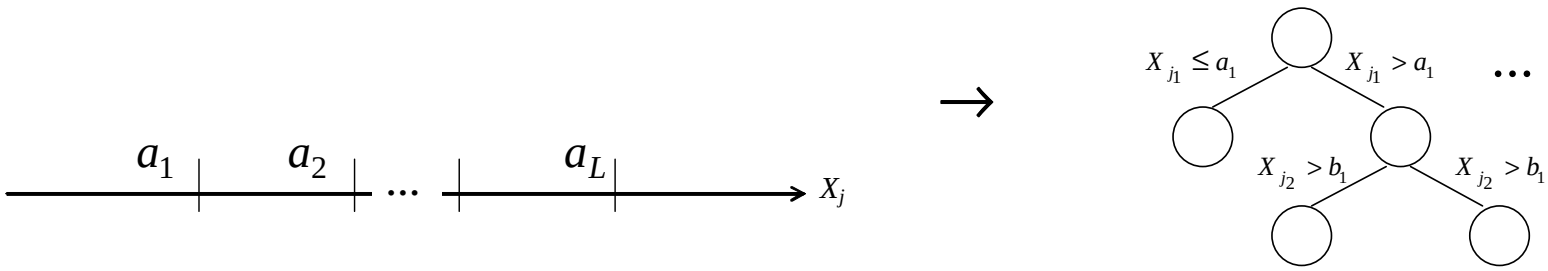
Пусть $N_m^{(k)}$ обозначает число объектов k-го класса, попавших в $E^{(m)}$.

Ошибка классификации $N_{er} = \frac{1}{N}\sum_{m=1}^{M}\ \sum_{\omega=1,\,\omega\neq Y^{(m)}}^{K} N_m^{(\omega)}$.

Если $Y^{(m)} = \arg\max_{\omega=1,\ldots,K} N_m^{(\omega)}$, тогда ошибка минимальна.

Построение оптимального дерева решений

Дискретизация количественных переменных → задача дискретной оптимизации:
найти наилучшее дерево по критерию качества среди всех



комбинаций переменных и их значений.

- Если некоторые наблюдения для X_j пропущены, то эти значения не рассматриваются при дискретизации X_j. Основные типы алгоритмов:

- Полный перебор;
- Динамическое программирование;
- Метод ветвей и границ;
- Жадный алгоритм.

Полный перебор невозможен при большой размерности, например, для бинарного дерева и n переменных число возможных вариантов:

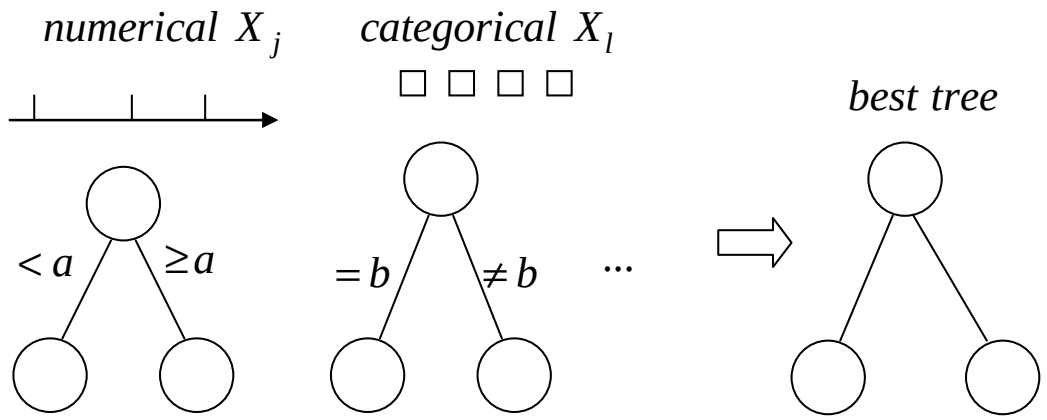
$$n\left((n-1)^2\right)\cdot\left((n-2)^4\right)\cdot\ldots\cdot 1 = n!(n-1)!\cdot\ldots\cdot 1.$$

Алгоритм поэтапного ветвления

Пусть M^* - максимально возможное число листьев,
$N_{\min}$ - минимально возможное число объектов в узле,
N_{er} допустимая величина ошибки в листе.

Шаги:

1. Делим корень на новые узлы перебирая переменные $X_1,\ldots,X_n$ и оставляя наилучший вариант по некоторому критерию;



2. Проверяем необходимость деления новых узлов:

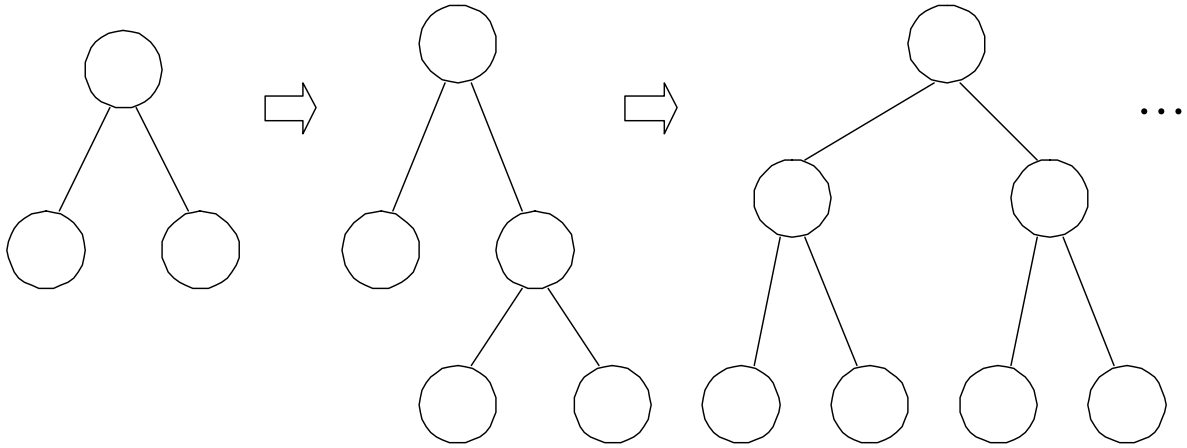
Если ветвление не требуется (число объектов в узле < $N_{\min}$),

или узел однородный (ошибка меньше, чем N_{er}),

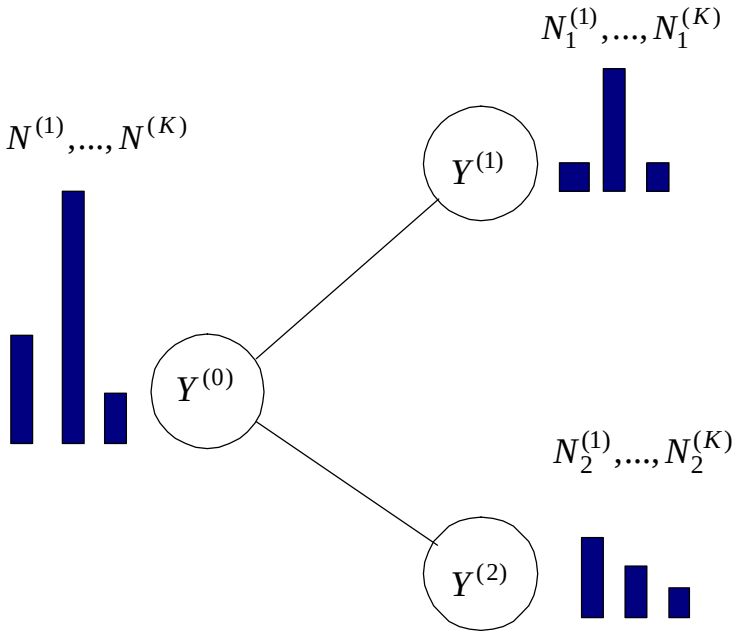
тогда ветвление не выполнятся, узел объявляется листом и наиболее частый класс присваивается этому листу.

3. Оставшиеся узлы делятся по тому же принципу, что и на шаге 1.

4. Шаги 2,3 повторяются до тех пор, пока не останется больше узлов, в которых необходимо выполнить разветвление или достигнута необходимая сложность дерева.



Критерии качества разделения

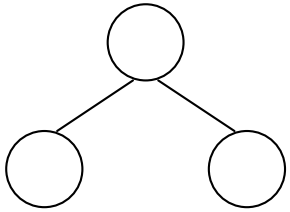


Уменьшение ошибки

$$\Delta P_{er} = \sum_{\substack{\omega=1,\\ \omega\neq Y^{(0)}}}^{K} N^{(\omega)} - \sum_{\substack{\omega=1,\\ \omega\neq Y^{(1)}}}^{K} N_1^{(\omega)} - \sum_{\substack{\omega=1,\\ \omega\neq Y^{(2)}}}^{K} N_2^{(\omega)};$$

$$\Delta P_{er} \to \max$$

Критерий Хи-квадрат (алгоритм CHAID, G.Kass, 1980)



$N_1^{(1)}=50, \quad N_2^{(1)}=20,$

$N_1^{(2)}=10, \quad N_2^{(2)}=60,$

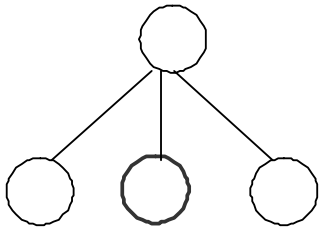
$N=120$

⇨ contingency table

	1 leaf	2 leaf
1 class	50	20
2 class	10	60

Основная гипотеза: разделение на группы (листья) не зависит от разделения на классы.

K



$1 \quad \ldots \; l \; \ldots \quad L$

В общем случае, получаем таблицу сопряженности $K \times L$:

$$S = \begin{pmatrix} N_1^{(1)} & \dots & N_L^{(1)} \\ \vdots & N_l^{(\omega)} & \vdots \\ N_1^{(K)} & \dots & N_L^{(K)} \end{pmatrix}.$$

Пуст $N^{(\omega)} = \sum_l N_l^{(\omega)}$, $N_l = \sum_\omega N_l^{(\omega)}$ - это маргинальные частоты строк и колонок.

Независимость строк и колонок означает, что

$\Rightarrow \; p_l^{(\omega)} = p^{(\omega)} p_l$

Ожидаемой число объектов в ячейке:

$$\bar{N}_l^{(\omega)} = p_l^{(\omega)} N = \frac{N^{(\omega)} N_l}{N}.$$

Обозначим $d_l^{(\omega)} = N_l^{(\omega)} - \bar{N}_l^{(\omega)}$ это отклонение от независимости.

Пусть

$$X^2 = \sum_{\omega=1}^{K}\sum_{l=1}^{L}\frac{(d_l^{(\omega)})^2}{\bar{N}_l^{(\omega)}} = N\left(\sum_{\omega,l}\frac{\left(N_l^{(\omega)}\right)^2}{N^{(\omega)}N_l} - 1\right).$$

Если H_0 верна, тогда $X^2 \approx \chi^2$ -распределение с $(K-1)(L-1)$ степенями свободы.

Variants of splitting can be compared with p-value

$$p_{value} = P(\chi^2 > x^2_{observed} \mid H_0 \ is \ \ true).$$

Наилучший вариант соответствует наибольшему значению X^2.

Информационный критерий

Пусть $H(0) = -\sum_{\omega=1}^{K} \frac{N^{(\omega)}}{N} \log \frac{N^{(\omega)}}{N}$ это энтропия в делимом узле,

$$H(L) = -\sum_{l=1}^{L} \frac{N_l}{N} \sum_{\omega=1}^{K} \frac{N_l^{(\omega)}}{N_l} \log \frac{N_l^{(\omega)}}{N_l}$$

- условная энтропия.

Чем меньше энтропия, тем больше информации.

Мера полезности деления:

$$gain = H(0) - H(L).$$

Индекс Джини

$$G(L) = \sum_{l=1}^{L} \frac{N_l}{N} \left(1 - \sum_{\omega=1}^{K} \left(\frac{N_l^{(\omega)}}{N_l} \right)^2 \right)$$

- оценивает распределение классов в дочерних вершинах.

Меньшее значение соответствует лучшему разбиению.

Algorithm ID3 (Quinlan, 1986)

Первоначально использовался для номинальных переменных.

Алгоритм:

– последовательное ветвление;

- число дочерних вершин = числу значений переменной;
- информационный критерий качества;
- остановка при достижении заданной глубины дерева.

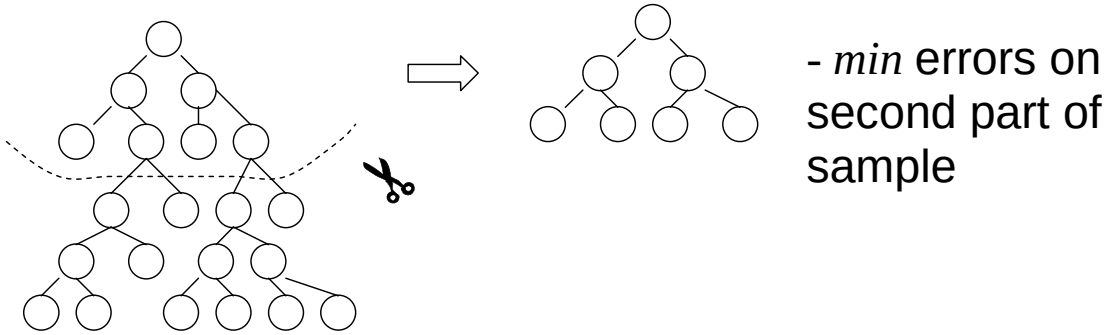
Algorithm C4.5 (Quinlan, 1993)

Усовершенствованная версия ID3:
- позволяет работать с количественными переменными;
недостаток предыдущего критерия – предпочитает варианты с большим числом дочерних вершин.
- нормированный информационный критерий:

$$gain\,ratio = \frac{H(0) - H(L)}{-\sum_{l=1}^{L} \frac{N_l}{N} log \frac{N_l}{N}}.$$

- процедура редуцирования (pruning, усечение): обучающая выборка случайным образом делится на две части (пропорция деления – параметр алгоритма).
- Первая часть - построение дерева жадным алгоритмом.
- Параметры алгоритма - такие, чтобы обеспечить максимально возможную точность решения («переобученное» дерево).

Вторая часть – для редуцирования (усечения, упрощения) полученного дерева таким образом, чтобы минимизировать частоту ошибки распознавания. Так как выборки независимы – частота ошибки для любого поддерева близка к вероятности ошибки.



Сравниваются всевозможные поддеревья исходного дерева; выбирается вариант, для которого частота ошибки по второй части выборки минимальна

Algorithm CART (Breiman, 1984)

Особенности:

- Распознавание + регрессионный анализ;
- Критерий качества ветвления:

при распознавании – индекс Gini;

при регрессионном анализе – дисперсия;

- Бинарное дерево;
- Процедура редуцирования дерева;
- Механизм обработки пропусков в данных.

Для номинальной переменной – полный перебор вариантов разбиения.

Деревья решений: достоинства

• автоматический отбор наиболее информативных переменных;
• для каждого объекта решение принимается по своему, как правило, небольшому набору переменных > повышение статистической устойчивости решений;
• линейная трудоемкость («жадный» алгоритм);
• решение - набор легко интерпретируемых логических закономерностей;
• возможность анализа разнотипных переменных, пропущенных значений;
• непараметрический подход к анализу данных;

Недостатки

• достаточно грубая аппроксимация непрерывных дискриминантных функций;
• использование простейших типов предикатов (<,>,=,?);
• трудоемкость перебора вариантов (для «нежадных» алгоритмов).