Логические алгоритмы классификации

К. В. Воронцов
vokov@forecsys.ru

Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Машинное обучение (курс лекций, К.В.Воронцов)»

октябрь 2012

Основное понятие — «закономерность»

X — пространство объектов;
Y — множество ответов;
$f_1(x), \dots, f_n(x)$ — признаки;
$X^\ell = (x_i, y_i)_{i=1}^\ell$ — обучающая выборка, $y_i = y(x_i)$.

Определение (пока неформальное)

Закономерность (правило, rule) — это предикат $\varphi\colon X \to \{0,1\}$, *удовлетворяющий двум требованиям:*

1) *интерпретируемость* (φ *зависит от 1–7 признаков);*
2) *информативность относительно класса* $c \in Y$:

$$p_c(\varphi, X^\ell) = \#\{x_i \colon \varphi(x_i) = 1 \text{ и } y_i = c\} \to \max;$$
$$n_c(\varphi, X^\ell) = \#\{x_i \colon \varphi(x_i) = 1 \text{ и } y_i \neq c\} \to \min;$$

Если $\varphi(x) = 1$, то говорят «φ выделяет x» (φ covers x).

Содержание

Требование интерпретируемости

Пример (из области медицины)

Если *возраст* > 60 ***и*** *пациент ранее перенёс инфаркт,*
то *операцию не делать, риск отрицательного исхода 60%.*

Пример (из области кредитного скоринга)

Если *в анкете указан домашний телефон*
и *зарплата* $> \$2000$ ***и*** *сумма кредита* $< \$5000$
то *кредит можно выдать, риск дефолта 5%.*

Требования интерпретируемости:

1) φ зависит от малого числа признаков;
2) формула φ выражается на естественном языке.

Виды интерпретируемых закономерностей

Параметрическое семейство *конъюнкций пороговых условий*:

$$\varphi(x) = \bigwedge_{j\in J} \bigl[\alpha_j \leqslant f_j(x) \leqslant \beta_j\bigr].$$

Параметрическое семейство *синдромных правил*:

$$\varphi(x) = \Bigl[\sum_{j\in J} \bigl[\alpha_j \leqslant f_j(x) \leqslant \beta_j\bigr] \geqslant K\Bigr].$$

Параметрическое семейство *шаров*:

$$\varphi(x) = \Bigl[\sum_{j\in J} \alpha_j |f_j(x) - f_j(x_0)|^\gamma \leqslant R^\gamma\Bigr].$$

Параметрическое семейство *полуплоскостей*:

$$\varphi(x) = \Bigl[\sum_{j\in J} \alpha_j f_j(x) \geqslant \alpha_0\Bigr].$$

Основная проблема — отбор признаков $J \subseteq \{1, \ldots, n\}$.

Логический (эвристический) критерий закономерности

$$\underbrace{\overbrace{\boxed{\begin{matrix}\varphi(x_i)=1\\ y_i=c\end{matrix}}}^{p_c(\varphi)}\boxed{\begin{matrix}\varphi(x_i)=0\\ y_i=c\end{matrix}}}_{P_c}\underbrace{\overbrace{\boxed{\begin{matrix}\varphi(x_i)=1\\ y_i\neq c\end{matrix}}}^{n_c(\varphi)}\boxed{\begin{matrix}\varphi(x_i)=0\\ y_i\neq c\end{matrix}}}_{N_c}$$

Определение

Предикат $\varphi(x)$ *— логическая* ε,δ*-закономерность класса* $c \in Y$

$$E_c(\varphi, X^\ell) = \frac{n_c(\varphi)}{p_c(\varphi) + n_c(\varphi)} \leqslant \varepsilon;$$

$$D_c(\varphi, X^\ell) = \frac{p_c(\varphi)}{\ell} \geqslant \delta.$$

Проблема: хотелось бы иметь один скалярный критерий.

Нетривиальность проблемы свёртки двух критериев

Пример.

Претенденты на звание «Критерий информативности»
при $P = 200$, $N = 100$ и различных p и n.

p	n	$p-n$	$p-5n$	$\frac{p}{P}-\frac{n}{N}$	$\frac{p}{n+1}$	I_c	IGain_c	$\sqrt{p}-\sqrt{n}$
50	0	**50**	50	0.25	50	22.65	23.70	7.07
100	50	**50**	−150	0	1.96	2.33	1.98	2.93
50	9	41	**5**	0.16	**5**	7.87	7.94	4.07
5	0	5	**5**	0.03	**5**	2.04	3.04	2.24
100	0	**100**	100	**0.5**	100	52.18	53.32	10.0
140	20	**120**	40	**0.5**	6.67	37.09	37.03	7.36

Статистический критерий информативности

Точный тест Фишера. Пусть X — в.п., выборка X^ℓ — i.i.d.
Гипотеза H_0: $y(x)$ и $\varphi(x)$ — независимые случайные величины.
Тогда вероятность реализации пары (p, n) — ГГР:

$$\mathrm{P}(p, n) = \frac{C_P^p C_N^n}{C_{P+N}^{p+n}}, \quad 0 \leqslant p \leqslant P, \quad 0 \leqslant n \leqslant N,$$

где $C_N^n = \frac{N!}{n!(N-n)!}$ — биномиальные коэффициенты.

Определение

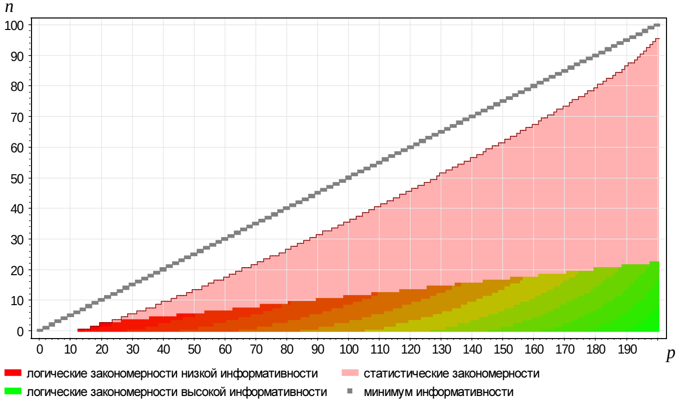
Информативность предиката $\varphi(x)$ относительно класса $c \in Y$:

$$I_c(\varphi, X^\ell) = -\ln \frac{C_{P_c}^{p_c(\varphi)} C_{N_c}^{n_c(\varphi)}}{C_{P_c+N_c}^{p_c(\varphi)+n_c(\varphi)}},$$

$I_c(\varphi, X^\ell) \geqslant I_0$ *— статистическая закономерность класса $c \in Y$.*

Соотношение логического и статистического критериев

Области логических ($\varepsilon = 0.1$) и статистических ($I_0 = 5$) закономерностей в координатах (p, n) при $P = 200$, $N = 100$.



Философский вопрос: закономерность == неслучайность?

Энтропийный критерий информативности

Пусть ω_0, ω_1 — два исхода с вероятностями q и $1-q$.
Количество информации: $I_0 = -\log_2 q$, $I_1 = -\log_2(1-q)$.
Энтропия — математическое ожидание количества информации:

$$h(q) = -q\log_2 q - (1-q)\log_2(1-q).$$

Энтропия выборки X^ℓ, если исходы — это классы $y{=}c$, $y{\neq}c$:

$$H(y) = h\left(\frac{P}{\ell}\right).$$

Энтропия выборки X^ℓ после получения информации φ:

$$H(y|\varphi) = \frac{p+n}{\ell} h\left(\frac{p}{p+n}\right) + \frac{\ell-p-n}{\ell} h\left(\frac{P-p}{\ell-p-n}\right).$$

Информационный выигрыш (Information gain, IGain):

$$\mathrm{IGain}_c(\varphi, X^\ell) = H(y) - H(y|\varphi).$$

Соотношение статистического и энтропийного критериев

Определение

Предикат φ — закономерность по энтропийному критерию, если $\mathrm{IGain}_c(\varphi, X^\ell) > G_0$ при некотором G_0.

Теорема

Энтропийный критерий IGain_c асимптотически эквивалентен статистическому I_c:

$$\mathrm{IGain}_c(\varphi, X^\ell) \to \frac{1}{\ell \ln 2}\, I_c(\varphi, X^\ell) \quad \textit{при } \ell \to \infty.$$

Доказательство:

применить формулу Стирлинга к статистическому критерию.

Задача перебора конъюнкций

Пусть $\mathscr{B}$ — конечное множество *элементарных предикатов*. Множество конъюнкций с ограниченным числом термов из $\mathscr{B}$:

$$\mathscr{K}_K[\mathscr{B}] = \bigl\{\varphi(x) = \beta_1(x) \wedge \dots \wedge \beta_k(x) \bigm| \beta_1, \dots, \beta_k \in \mathscr{B},\ k \leqslant K\bigr\}.$$

Число допустимых конъюнкций: $O(|\mathscr{B}|^K)$ — ооооочень много!

Семейство методов локального поиска

Окрестность $V(\varphi)$ — все конъюнкции, получаемые из φ добавлением, изъятием или модификацией одного из термов.

Основная идея: на t-й итерации

$$\varphi_t := \arg\max_{\varphi \in V(\varphi_{t-1})} I_c(\varphi, X^\ell).$$

Обобщённый алгоритм локального поиска

Вход: выборка X^ℓ; класс $c \in Y$;
начальное приближение φ_0; параметры $t_{\max}$, d, ε;

Выход: конъюнкция φ;

1: $I^* := I_c(\varphi_0, X^\ell)$; $\quad \varphi^* := \varphi_0$;
2: **для всех** $t = 1, \dots, t_{\max}$
3: $\quad \varphi_t := \arg\max\limits_{\varphi \in V(\varphi_{t-1})} I_c(\varphi, X^\ell)$ — перспективная конъюнкция;
4: $\quad \varphi_t^* := \arg\max\limits_{\substack{\varphi \in V(\varphi_{t-1}) \\ E_c(\varphi) < \varepsilon}} I_c(\varphi, X^\ell)$ — лучшая конъюнкция;
5: $\quad$ **если** $I_c(\varphi_t^*) > I^*$ **то** $t^* := t$; $\varphi^* := \varphi_t^*$; $I^* := I_c(\varphi^*)$;
6: $\quad$ **если** $t - t^* > d$ **то выход**;
7: **вернуть** φ^*;

Частные случаи и модификации

- **жадный алгоритм:**
  $V(\varphi)$ — только добавления термов; $\varphi_0 = \varnothing$;
- **стохастический локальный поиск (SLS):**
  $V(\varphi)$ — случайное подмножество всевозможных добавлений, удалений, модификаций термов; $\varphi_0 = \varnothing$;
- **стабилизация:**
  $V(\varphi)$ — удаления термов или изменение параметров в термах; $\varphi_0 \neq \varnothing$;
- **редукция:**
  $V(\varphi)$ — только удаления термов; $\varphi_0 \neq \varnothing$;
  $I_c(\varphi, X^k)$ оценивается по контрольной выборке X^k;
- **поиск в ширину:**
  на каждой итерации строится множество конъюнкций $\Phi_t = \{\varphi_t\}$.

Поиск закономерностей — это отбор признаков

Отличия от методов отбора признаков:

- вместо внешнего критерия $Q_{\text{ext}} \to \min$ критерий информативности $I_c \to \max$;
- есть ограничение на число признаков $|J| \leqslant K$.
- надо построить не одно, а *много различных* правил, которые должны образовать покрытие выборки.

Все механизмы отбора признаков подходят:

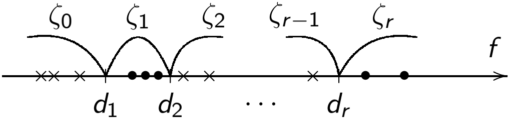
- добавления–удаления;
- поиск в глубину;
- поиск в ширину;
- генетические (эволюционные) алгоритмы;
- случайный поиск с адаптацией.

Вспомогательная задача бинаризации вещественного признака

Цель: сократить перебор предикатов вида $\bigl[\alpha \leqslant f(x) \leqslant \beta\bigr]$.

Дано: выборка значений вещественного признака $f(x_i)$, $x_i \in X^\ell$.
Найти: наиболее информативное разбиение области значений признака на относительно небольшое число зон:

$$\zeta_0(x) = [f(x) < d_1];$$
$$\zeta_s(x) = [d_s \leqslant f(x) < d_{s+1}], \qquad s = 1, \dots, r-1;$$
$$\zeta_r(x) = [d_r \leqslant f(x)].$$



Алгоритм разбиения области значений признака на зоны

Вход: выборка X^ℓ; класс $c \in Y$; параметры r и δ_0.
Выход: $D = \{d_1 < \dots < d_r\}$ — последовательность порогов;

$D := \varnothing$; упорядочить выборку X^ℓ по возрастанию $f(x_i)$;
для всех $i = 2, \dots, \ell$
  если $f(x_{i-1}) \neq f(x_i)$ и $[y_{i-1} = c] \neq [y_i = c]$ **то**
    добавить порог $\frac{1}{2}(f(x_{i-1}) + f(x_i))$ в конец D;
повторять
  для всех $d_i \in D$, $i = 1, \dots, |D| - 1$
    $\delta I_i := I_c(\zeta_{i-1} \vee \zeta_i \vee \zeta_{i+1}) - \max\{I_c(\zeta_{i-1}), I_c(\zeta_i), I_c(\zeta_{i+1})\}$;
  $i := \arg\max\limits_s \delta I_s$;
  если $\delta I_i > \delta_0$ **то**
    слить зоны ζ_{i-1}, ζ_i, ζ_{i+1}, удалив d_i и d_{i+1} из D;
пока $|D| > r + 1$.

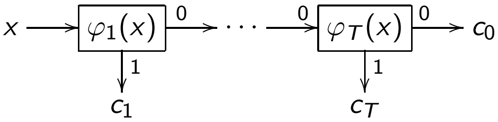
Резюме в конце лекции

- Правило — это интерпретируемый предикат $X \to \{0,1\}$.
- Закономерность — это информативное правило.
- Существует очень много критериев информативности.
- Статистический критерий — для поиска закономерностей.
- Логический ε,δ-критерий — для отбора закономерностей.
- Механизмы Rule Induction — те же, что Features Selection.
- Бинаризация — предварительный этап сокращения поиска.

Определение решающего списка

Решающий список (decision list, DL)
— алгоритм классификации $a\colon X \to Y$, который задаётся закономерностями $\varphi_1(x), \ldots, \varphi_T(x)$ классов $c_1, \ldots, c_T \in Y$:

> **для всех** $t = 1, \ldots, T$
>   **если** $\varphi_t(x) = 1$ **то**
>   **вернуть** c_t;
> **вернуть** c_0.



«Особый ответ» c_0 — отказ от классификации объекта x.

Построение решающего списка

Вход: выборка X^ℓ; семейство предикатов Φ;
параметры: $T_{\max}$, $I_{\min}$, $E_{\max}$, ℓ_0;

Выход: решающий список $\{\varphi_t, c_t\}_{t=1}^{T}$;

1: $U := X^\ell$;
2: **для всех** $t := 1, \dots, T_{\max}$
3: $c := c_t$ — выбрать класс из Y;
4: $\Phi' = \bigl\{\varphi \in \Phi \colon E_c(\varphi, U) \leqslant E_{\max}\bigr\}$;
5: $\varphi_t := \arg\max\limits_{\varphi \in \Phi'} I_c(\varphi, U)$;
6: **если** $I_c(\varphi_t, U) < I_{\min}$ **то выход**;
7: исключить из выборки объекты, выделенные правилом φ_t:
$U := \{x \in U : \varphi_t(x) = 0\}$;
8: **если** $|U| \leqslant \ell_0$ **то выход**;

Замечания к алгоритму построения решающего списка

- Параметр $E_{\max}$ позволяет управлять сложностью списка:
  $E_{\max}\downarrow \quad\Rightarrow\quad p(\varphi_t)\downarrow,\;\; T\uparrow.$
- Стратегии выбора класса c_t:
  1) все классы по очереди;
  2) на каждом шаге определяется оптимальный класс:
  $$(\varphi_t, c_t) := \mathop{\arg\max}_{\varphi\in\Phi', c\in Y} I_c(\varphi, U);$$
- Простой обход проблемы пропусков в данных.
- Другие названия:
  — комитет с логикой старшинства;
  — голосование по старшинству;
  — машина покрывающих множеств (SCM);

Решающие списки: достоинства и недостатки

Достоинства:

- Интерпретируемость и простота классификации.
- Универсальность: можно использовать любое семейство Φ.
- Допустимы разнотипные данные и данные с пропусками.
- Правила получаются различными. Можно построить несколько списков и по ним проголосовать.

Недостатки:

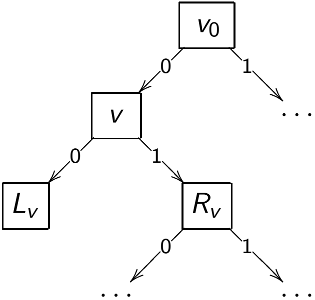
- При неудачном выборе Φ список может не построиться, будет много отказов от классификации.
- Список плохо интерпретируется, если он длинный и/или правила различных классов следуют вперемежку.
- Качество классификации обычно ниже, чем у голосования, когда правила могут компенсировать ошибки друг друга.

Определение бинарного решающего дерева

Бинарное решающее дерево — алгоритм классификации $a(x)$, задающийся бинарным деревом:

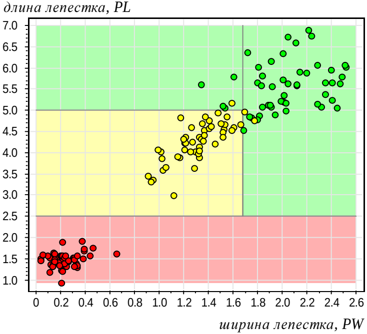
1) $\forall v \in V_{\text{внутр}} \;\to\;$ предикат $\beta_v : X \to \{0,1\}$, $\;\beta \in \mathscr{B}$
2) $\forall v \in V_{\text{лист}} \;\to\;$ имя класса $c_v \in Y$.

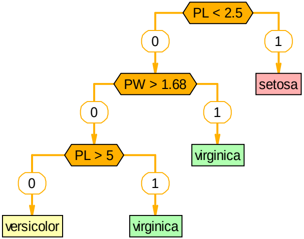
1: $v := v_0$;
2: **пока** $v \in V_{\text{внутр}}$
3:     **если** $\beta_v(x) = 1$ **то**
4:         переход вправо:
            $v := R_v$;
5:     **иначе**
6:         переход влево:
            $v := L_v$;
7: **вернуть** c_v.



Пример решающего дерева

Задача Фишера о классификации цветков ириса на 3 класса, в выборке по 50 объектов каждого класса, 4 признака.





На графике: в осях двух самых информативных признаков (из 4) два класса разделились без ошибок, на третьем 3 ошибки.

Жадный алгоритм построения дерева ID3

1: **ПРОЦЕДУРА** LearnID3 $(U \subseteq X^\ell)$;

2: **если** все объекты из U лежат в одном классе $c \in Y$ **то**

3: **вернуть** новый лист v, $c_v := c$;

4: найти предикат с максимальной информативностью:
$\beta := \arg\max\limits_{\beta \in \mathscr{B}} I(\beta, U)$;

5: разбить выборку на две части $U = U_0 \cup U_1$ по предикату β:
$U_0 := \bigl\{x \in U \colon \beta(x) = 0\bigr\}$;
$U_1 := \bigl\{x \in U \colon \beta(x) = 1\bigr\}$;

6: **если** $U_0 = \varnothing$ или $U_1 = \varnothing$ **то**

7: **вернуть** новый лист v, $c_v :=$ Мажоритарный класс(U);

8: создать новую внутреннюю вершину v: $\beta_v := \beta$;
построить левое поддерево: $L_v :=$ LearnID3 (U_0);
построить правое поддерево: $R_v :=$ LearnID3 (U_1);

9: **вернуть** v;

Разновидности критериев ветвления

1. **Отделение одного класса** (слишком сильное ограничение):

$$I(\beta, X^\ell) = \max_{c \in Y} I_c(\beta, U).$$

2. **Многоклассовый энтропийный критерий:**

$$I(\beta, X^\ell) = \sum_{c \in Y} h\left(\frac{P_c}{\ell}\right) - \frac{p}{\ell} \sum_{c \in Y} h\left(\frac{p_c}{p}\right) - \frac{\ell - p}{\ell} \sum_{c \in Y} h\left(\frac{P_c - p_c}{\ell - p}\right),$$

где $P_c = \#\{x_i\colon y_i = c\}$, $\quad p = \#\{x_i\colon \beta(x_i) = 1\}$, $\quad h(z) \equiv -z \log_2 z$.

3. **Критерий Джини:**

$$I(\beta, X^\ell) = \#\bigl\{(x_i, x_j)\colon \beta(x_i) = \beta(x_j) \text{ и } y_i = y_j\bigr\}.$$

4. *D*-**критерий В.И.Донского:**

$$I(\beta, X^\ell) = \#\bigl\{(x_i, x_j)\colon \beta(x_i) \neq \beta(x_j) \text{ и } y_i \neq y_j\bigr\}.$$

Обработка пропусков

На стадии обучения:

- Если $\beta(x_i)$ не определено, то при вычислении $I(\beta, U)$ объект x_i исключается из выборки U.
- Для $\forall v \in V_{\text{внутр}}$ оценивается:
  $\hat{p}_L = |U_0|/|U|$ — вероятность левой ветви;
  $\hat{p}_R = |U_1|/|U|$ — вероятность правой ветви.

На стадии классификации:

- $\beta_v(x)$ не определено $\Rightarrow$ *пропорциональное распределение*:
$$\hat{P}_v(y|x) = \begin{cases} [y = c_v], & v \in V_{\text{лист}}; \\ \hat{p}_L \hat{P}_{L_v}(y|x) + \hat{p}_R \hat{P}_{R_v}(y|x), & v \in V_{\text{внутр}}. \end{cases}$$
- Окончательное решение — байесовское правило:
$$y = \arg\max_{y \in Y} \hat{P}_v(y|x).$$

Решающие деревья: достоинства и недостатки

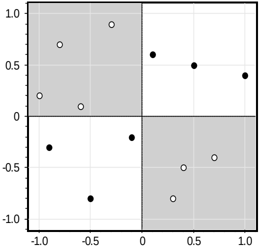
Достоинства:

- Интерпретируемость и простота классификации.
- Гибкость: можно варьировать множество $\mathscr{B}$.
- Допустимы разнотипные данные и данные с пропусками.
- Трудоёмкость линейна по длине выборки $O(|\mathscr{B}|h\ell)$.
- Не бывает отказов от классификации.

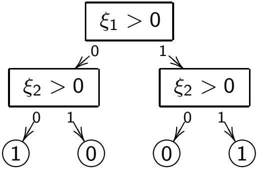
Недостатки:

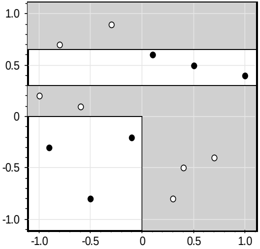
- Жадный ID3 переусложняет структуру дерева, и, как следствие, сильно переобучается.
- Фрагментация выборки: чем дальше v от корня, тем меньше статистическая надёжность выбора β_v, c_v.
- Высокая чувствительность к шуму, к составу выборки, к критерию информативности.

Жадный ID3 переусложняет структуру дерева

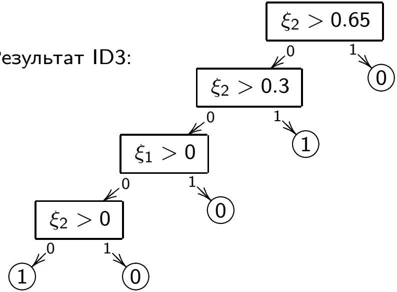


Оптимальное дерево для задачи XOR:





Результат ID3:



Стратегия пред-просмотра (look ahead)

Шаг 4:
найти предикат с максимальной информативностью:
$\beta := \arg\max\limits_{\beta \in \mathscr{B}} I(\beta, U)$;

Шаг 4 заменяется на более ресурсоёмкую процедуру:

для всех деревьев T глубины h
    $r_T(U) =$ число ошибок дерева T на выборке U;
$\beta :=$ корень лучшего поддерева $\arg\min\limits_{T} r_T(U)$;

Достоинства:

- Задача XOR решается практически идеально.

Недостатки:

- При $h > 2$ оооооооочень долго.
- На реальных данных улучшение незначительно.

Стратегия пред-редукции (pre-pruning)

Шаг 6:
если $U_0 = \varnothing$ или $U_1 = \varnothing$ **то**
  вернуть новый лист v;

Шаг 6 заменяется на более мягкое условие:

если $I(\beta, U) \leqslant I_0$ **то**
  вернуть новый лист v;

Достоинства:

- Сразу строится более простое дерево.

Недостатки:

- Качество дерева может и не улучшиться.

Стратегия пост-редукции (post-pruning: C4.5, CART)

X^k — независимая контрольная выборка, $k \approx 0.5\ell$.

1: **для всех** $v \in V_{\text{внутр}}$
2: $S_v :=$ подмножество объектов X^k, дошедших до v;
3: **если** $S_v = \varnothing$ **то**
4: **вернуть** новый лист v, $c_v :=$ Мажоритарный класс(U);
5: число ошибок при классификации S_v четырьмя способами:
$r(v)$ — поддеревом, растущим из вершины v;
$r_L(v)$ — поддеревом левой дочерней вершины L_v;
$r_R(v)$ — поддеревом правой дочерней вершины R_v;
$r_c(v)$ — к классу $c \in Y$.
6: в зависимости от того, какое из них минимально:
сохранить поддерево v;
заменить поддерево v поддеревом L_v;
заменить поддерево v поддеревом R_v;
заменить поддерево v листом, $c_v := \arg\min\limits_{c \in Y} r_c(v)$.

Преобразование решающего дерева в список (C4.5-rules)

- Для любого бинарного решающего дерева
$$a(x) = \arg\max_{y \in Y} \sum_{v \in V_{\text{лист}}} [c_v = y] K_v(x),$$
где $K_v(x)$ — конъюнкция по всем рёбрам пути $[v_0, v]$:
$$K_v(x) = \bigwedge_{(u, R_u)} \beta_u(x) \bigwedge_{(u, L_u)} \bar{\beta}_u(x).$$
- Редукция $K_v(x)$, $\forall v \in V_{\text{лист}}$ по контрольной выборке X^k.

Достоинства:

- Переобучение, как правило, уменьшается.

Недостатки:

- Преобразование в список необратимо: это уже не дерево.

Резюме в конце лекции

- Преимущества решающих деревьев:
  - интерпретируемость,
  - допускаются разнотипные данные,
  - возможность обхода пропусков;
- Недостатки решающих деревьев:
  - переобучение,
  - фрагментация,
  - неустойчивость к шуму, составу выборки, критерию;
- Способы устранения этих недостатков:
  - редукция,
  - специальные виды деревьев ADT, ODT — см. далее,
  - композиции (леса) деревьев — см. далее;

Взвешенное голосование закономерностей

$R_c = \{\varphi_c^t(x) \colon t = 1, \ldots, T_c\}$ — список закономерностей класса c.

Взвешенное голосование (weighted voting):

$$a(x) = \arg\max_{c \in Y} \Gamma_c(x), \quad \Gamma_c(x) = \sum\nolimits_{t=1}^{T_c} \alpha_c^t \varphi_c^t(x).$$

Отступ: $M(x_i) = \Gamma_{y_i}(x_i) - \max\limits_{y \in Y \setminus y_i} \Gamma_y(x_i)$.

Жадный алгоритм построения композиции:

Вход: выборка X^ℓ, семейство правил Φ;
Выход: $(\varphi_c^t, \alpha_c^t)$, $t = 1, \ldots, T_c$, $\forall c \in Y$;

1: **для** некоторых правил $\varphi \in \Phi$
2: **если** φ — закономерность для некоторого класса c
и φ существенно отличается от $\forall \psi \in R_c$
и φ существенно улучшает композицию **то**
3: добавить $\varphi_c^t = \varphi$ в R_c и оценить α_c^t;

Принцип существенной различности закономерностей (и его простейшее вероятностное обоснование)

Пусть $\mathsf{E}M > 0$ (классификация немного лучше, чем наугад).
Оценим вероятность ошибки по неравенству Чебышева:

$$\mathsf{P}\{M < 0\} \leqslant \mathsf{P}\bigl\{|\mathsf{E}M - M| > \mathsf{E}M\bigr\} \leqslant \frac{\mathsf{D}M}{(\mathsf{E}M)^2}.$$

Вывод: для уменьшения вероятности ошибки необходимо

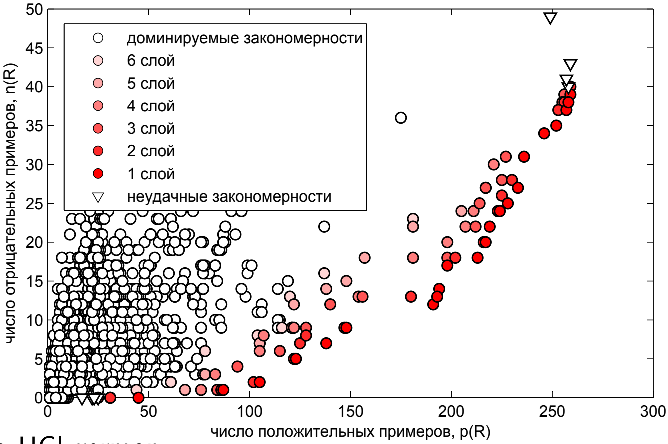
- максимизировать среднее значение отступа $\mathsf{E}M$;
- минимизировать его дисперсию $\mathsf{D}M$.

Для этого

- каждый объект должен выделяться примерно одинаковым числом закономерностей;
- закономерности должны выделять существенно различные подмножества объектов.

Что значит «правила существенно различаются»?

Парето-фронт — множество недоминируемых правил (правило R недоминируемо, если правее и ниже правил нет)

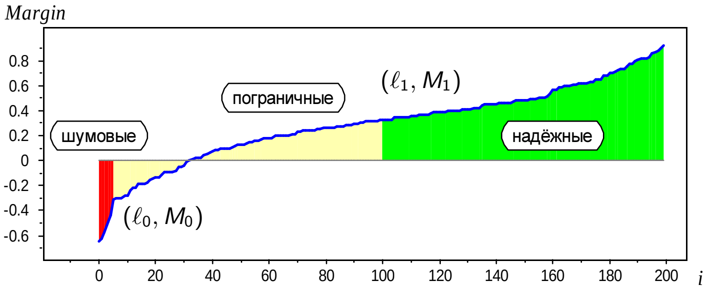


задача UCI:german

Что значит «правило существенно улучшает композицию»?

Отступ (margin) объекта x_i: $M(x_i) = \Gamma_{y_i}(x_i) - \max\limits_{y \neq y_i} \Gamma_y(x_i)$.

Распределение объектов выборки X^ℓ по значениям отступов:



Задача обучения правила φ_c^t путём выравнивания отступов:

$$\sum_{i=\ell_0}^{\ell_1} M(x_i) \to \max_{(\varphi_c^t, \alpha_c^t)}.$$

Бустинг закономерностей

Два класса, $Y = \{-1, +1\}$.
Композиция из $T = T_{-1} + T_{+1}$ закономерностей:

$$a_T(x) = \operatorname{sign}\Bigl(\underbrace{\sum_{t=1}^{T_{+1}} \alpha_{+1}^t \varphi_{+1}^t(x)}_{\Gamma_{+1}(x)} - \underbrace{\sum_{t=1}^{T_{-1}} \alpha_{-1}^t \varphi_{-1}^t(x)}_{\Gamma_{-1}(x)}\Bigr), \quad \alpha_c^t > 0.$$

Эвристика 1: добавляем $\alpha_c^{T+1}\varphi_c^{T+1}(x)$, не трогая $\alpha_c^t\varphi_c^t(x)$, $t < T$.
Эвристика 2: экспоненциальная аппроксимация:

$$Q_T = \sum_{i=1}^{\ell} \bigl[\Gamma_{y_i}(x_i) - \Gamma_{-y_i}(x_i) < 0\bigr] \leqslant$$

$$\leqslant \tilde{Q}_T = \sum_{i=1}^{\ell} \underbrace{\exp\bigl(\Gamma_{-y_i}(x_i) - \Gamma_{y_i}(x_i)\bigr)}_{w_i} = \sum_{i=1}^{\ell} w_i.$$

Основная теорема бустинга

Добавляем ещё одну закономерность $\alpha_c \varphi_c(x)$:

$$\tilde{Q}_{T+1}(\varphi_c, \alpha_c) = \sum_{y_i = c} w_i e^{-\alpha_c \varphi_c(x_i)} + \sum_{y_i \neq c} w_i e^{\alpha_c \varphi_c(x_i)}$$

Теорема

Минимум функционала $\tilde{Q}_{T+1}(\varphi_c, \alpha_c)$ достигается при

$$\varphi_c^* = \arg\max_{\varphi} \sqrt{p_c^w(\varphi)} - \sqrt{n_c^w(\varphi)};$$

$$\alpha_c^* = \frac{1}{2} \ln \frac{p_c^w(\varphi_c^*)}{n_c^w(\varphi_c^*)};$$

$$p_c^w(\varphi) = \sum_{i=1}^{\ell} w_i \, [y_i = c] \, \varphi(x_i);$$

$$n_c^w(\varphi) = \sum_{i=1}^{\ell} w_i \, [y_i \neq c] \, \varphi(x_i);$$

Алгоритм AdaBoost для закономерностей

Вход: выборка X^ℓ; семейство правил Φ; параметры T, δ;
Выход: закономерности и их веса $\varphi_c^t(x), \alpha_c^t$, $t = 1..T_c$, $c \in Y$;

1: инициализация: $w_i := 1$, $i = 1, \dots, \ell$;
2: **для всех** $t = 1, \dots, T$
3: $c := c_t$ — выбрать класс закономерности;
4: $\varphi_c^t := \arg\max\limits_{\varphi \in \Phi} \sqrt{p_c^w(\varphi)} - \sqrt{n_c^w(\varphi)}$;
5: $\alpha_c^t := \dfrac{1}{2} \ln \dfrac{p_c^w(\varphi) + \delta}{n_c^w(\varphi) + \delta}$;
6: **для всех** $i = 1, \dots, \ell$
7: **если** $\varphi_c(x_i) = 1$ **то** $w_i := \begin{cases} w_i \exp(-\alpha_c^t), & y_i = c; \\ w_i \exp(\alpha_c^t), & y_i \neq c; \end{cases}$
8: нормировка: $Z := \frac{1}{\ell} \sum\limits_{i=1}^{\ell} w_i$; $w_i := w_i / Z$, $i = 1, \dots, \ell$;

Алгоритм AdaBoost для закономерностей

Достоинства:

- Бустинг одновременно
  - максимизирует информативность правил;
  - увеличивает их различность;
  - строит из них покрытие выборки.
- Низкое переобучение благодаря выравниванию отступов.
- Универсальность: можно использовать любое семейство Φ.

Недостатки:

- Чувствительность к шуму (лучше использовать AnyBoost).
- В некоторых задачах переобучение всё же наблюдается, хотя, при очень больших $T \sim 10^2 \div 10^3$.
- Утрата интерпретируемости при больших $T \sim 10 \div 20$.

Решающие леса

Бустинг решающих деревьев:

- базовые классификаторы — деревья ограниченной глубины.

Случайные леса (Random Forests):

- T деревьев обучаются по случайным подвыборкам (bagging);
- $\mathscr{B}$ — случайное множество гиперплоскостей;
- $I(\beta, U)$ — энтропийный критерий информативности;
- Простое голосование по T деревьям.

Решающий список из решающих деревьев:

- При образовании статистически ненадёжного листа этот лист заменяется переходом к следующему дереву.
- Следующее дерево строится по всем объектам, прошедшим через ненадёжные листы предыдущего дерева.

Чередующееся решающее дерево — ADT (Alternating Decision Tree) [Freund, Mason, 1999]

Пусть $Y = \{-1, +1\}$.

Для каждой вершины $v \in V$ задаётся:

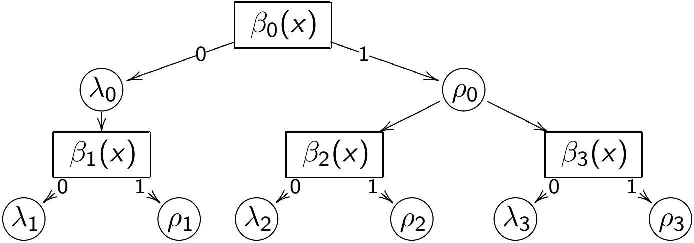
1) предикат $\beta_v : X \to \{0, 1\}$, $\beta \in \mathscr{B}$
2) L_v, R_v — множества левых и правых дочерних вершин;
3) λ_v, ρ_v — левый и правый коэффициент.

Классификация вычисляется рекурсивно:

$$a(x) = \operatorname{sign} \mathrm{ADT}(x, v_0), \quad v_0 \text{ — корень дерева;}$$

$$\mathrm{ADT}(x, v) = \begin{cases} \lambda_v + \sum\limits_{t \in L_v} \mathrm{ADT}(x, t), & \beta_v(x) = 0; \\ \rho_v + \sum\limits_{t \in R_v} \mathrm{ADT}(x, t), & \beta_v(x) = 1; \end{cases}$$

Пример



Многократное применение рекуррентной формулы приводит к взвешенному голосованию конъюнкций:

$$
\begin{aligned}
\mathrm{ADT} &= \bar\beta_0(\lambda_0 + \bar\beta_1\lambda_1 + \beta_1\rho_1) + \\
&\quad + \beta_0(\rho_0 + \bar\beta_2\lambda_2 + \beta_2\rho_2 + \bar\beta_3\lambda_3 + \beta_3\rho_3) = \\
&= (\bar\beta_0\lambda_0 + \beta_0\rho_0) + (\bar\beta_0\bar\beta_1\lambda_1 + \bar\beta_0\beta_1\rho_1) + \\
&\qquad\qquad + (\beta_0\bar\beta_2\lambda_2 + \beta_0\beta_2\rho_2) + (\beta_0\bar\beta_3\lambda_3 + \beta_0\beta_3\rho_3).
\end{aligned}
$$

Бинарное решающее дерево является частным случаем ADT

...если наложить на ADT ограничение $|L_v| = |R_v| \leqslant 1$
и положить λ_v, ρ_v равными либо 0, либо меткам классов:

$$\lambda_v = \begin{cases} 0, & |L_v| = 1; \\ c_v \in Y, & |L_v| = 0; \end{cases}$$

$$\rho_v = \begin{cases} 0, & |R_v| = 1; \\ c_v \in Y, & |R_v| = 0. \end{cases}$$

Коэффициенты $\lambda_v = c_v$ и $\rho_v = c_v$ играют роль терминальных вершин.

В ADT терминальных вершин нет, все вершины равноправны.

Обучение ADT методом бустинга

В общем случае

$$a(x) = \operatorname{sign} \sum_{v \in V} \underbrace{\lambda_v K_v(x) \bar{\beta}_v(x) + \rho_v K_v(x) \beta_v(x)}_{\Gamma(x)},$$

где $K_v(x)$ — конъюнкция всех предикатов на пути $[v_0, v)$.

Идея наращивания дерева. Создав вершину u, надо решить:
1) к какой вершине v её прицепить;
2) слева (тогда $K_u = K_v \bar{\beta}_v$) или справа (тогда $K_u = K_v \beta_v$);
3) какой взять предикат β_u из семейства $\mathscr{B}$;
4) как оптимизировать λ_u, ρ_u.

$$a(x) = \operatorname{sign}\Big(\Gamma(x) + K_u(x)\big(\lambda_u \bar{\beta}_u(x) + \rho_u \beta_u(x)\big)\Big).$$

Обучение ADT методом бустинга

Упростим обозначения:
$K_i \equiv K_u(x_i), \quad \beta_i \equiv \beta_u(x_i), \quad \bar{\beta}_i \equiv \bar{\beta}_u(x_i), \quad \lambda \equiv \lambda_u, \quad \rho \equiv \rho_u.$

Функционал числа ошибок после добавления вершины u:

$$Q = \sum_{i=1}^{\ell} \Bigl[\Gamma(x_i)y_i + \lambda K_i \bar{\beta}_i y_i + \rho K_i \beta_i y_i < 0\Bigr].$$

AbaBoost: экспоненциальная аппроксимация $[M<0] \leqslant e^{-M}$

$$Q \leqslant \tilde{Q} = \sum_{i=1}^{\ell} \underbrace{\exp\bigl(-\Gamma(x_i)y_i\bigr)}_{w_i} \exp\bigl(-K_i y_i(\lambda\bar{\beta}_i + \rho\beta_i)\bigr) \to \min_{\lambda,\rho,v,\beta}.$$

В отличие от AdaBoost, требуется найти 2 коэффициента λ, ρ.

Основная теорема бустинга

Теорема

Минимум функционала $\tilde{Q}$ достигается при

$$\lambda^* = \frac{1}{2} \ln \frac{p(K\bar{\beta}^*)}{n(K\bar{\beta}^*)}; \qquad \rho^* = \frac{1}{2} \ln \frac{p(K\beta^*)}{n(K\beta^*)};$$

$$\beta^* = \arg\min_{K,\beta} \Bigl(W(\bar{K}) + 2\sqrt{p(K\bar{\beta})n(K\bar{\beta})} + 2\sqrt{p(K\beta)n(K\beta)} \Bigr);$$

$$p(K\beta) = \textstyle\sum_{i=1}^{\ell} w_i\, K_i\beta_i[y_i = +1];$$

$$n(K\beta) = \textstyle\sum_{i=1}^{\ell} w_i\, K_i\beta_i[y_i = -1];$$

$$W(\bar{K}) = \textstyle\sum_{i=1}^{\ell} w_i\, \bar{K}_i.$$

$p(K\beta)n(K\beta)$ — «число пар объектов» разных классов справа;
$p(K\bar{\beta})n(K\bar{\beta})$ — «число пар объектов» разных классов слева;
$W(\bar{K})$ — «число объектов», не дошедших до вершины v.

Алгоритм AdaBoost для обучения ADT

Вход: выборка X^ℓ; семейство правил $\mathscr{B}$; параметры T, δ;

Выход: $\mathrm{ADT} = \langle \beta_v, L_v, R_v, \lambda_v, \rho_v \rangle_{v \in V}$;

инициализация: $V := \varnothing$; $w_i := 1$, $i = 1, \dots, \ell$;
для всех $t = 1, \dots, T$
  создать вершину u; $L_u := R_u := \varnothing$;
  минимизируя $\tilde{Q}$, найти: (1) предикат $\beta_u \in \mathscr{B}$, (2) к какой вершине $v \in V$ присоединить u, (3) слева или справа;
  если u присоединяется к v слева **то** $L_v := L_v \cup \{u\}$;
  если u присоединяется к v справа **то** $R_v := R_v \cup \{u\}$;
  $\lambda_u := \frac{1}{2} \ln \frac{p(K\bar{\beta}_u)+\delta}{n(K\bar{\beta}_u)+\delta}$; $\rho_u := \frac{1}{2} \ln \frac{p(K\beta_u)+\delta}{n(K\beta_u)+\delta}$;
  для всех $i = 1, \dots, \ell$ таких, что $K_u(x_i) = 1$
    $w_i := w_i \exp\bigl(-\lambda_u \bar{\beta}(x_i) y_i - \rho_u \beta(x_i) y_i\bigr)$;
  нормировка: $Z := \frac{1}{\ell} \sum_{i=1}^{\ell} w_i$; $w_i := w_i / Z$, $i = 1, \dots, \ell$;

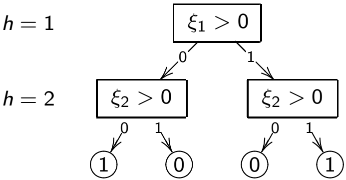
Невнимательные решающие деревья — ODT (Oblivious Decision Tree) [1993]

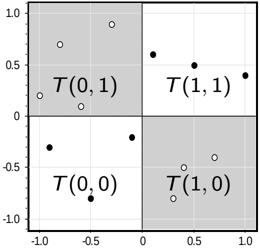
Решение проблемы фрагментации:
строится сбалансированное дерево высоты H;
для всех узлов уровня h условие ветвления $\beta_h(x)$ *одинаково*;
на уровне h ровно 2^{h-1} вершин; X делится на 2^H ячеек.

Классификатор задаётся *таблицей решений* $T\colon \{0,1\}^H \to Y$:

$$a(x) = T\bigl(\beta_1(x), \ldots, \beta_H(x)\bigr).$$

Пример: задача XOR, $H = 2$.





Алгоритм обучения ODT

Вход: выборка X^ℓ; семейство правил $\mathscr{B}$; глубина дерева H;
Выход: условия β_h, $h = 1, \dots, H$; таблица $T\colon \{0,1\}^H \to Y$;

1: **для всех** $h = 1, \dots, H$
2: найти предикат с максимальной информативностью:
$$\beta_h := \arg\max_{\beta \in \mathscr{B}} I(\beta_1, \dots, \beta_{h-1}, \beta; X^\ell);$$
3: **для всех** $b \equiv (b_1, \dots, b_H) \in \{0,1\}^H$
4: классификация по мажоритарному правилу:
$$T(b_1, \dots, b_H) := \arg\max_{c \in Y} \sum_{i=1}^{\ell} [y_i = c] \prod_{h=1}^{H} \bigl[\beta_h(x_i) = b_h\bigr];$$

$$I(\beta_1, \dots, \beta_h) = \sum_{c \in Y} h\left(\frac{P_c}{\ell}\right) - \sum_{b \in \{0,1\}^h} \frac{|X_b|}{\ell} \sum_{c \in Y} h\left(\frac{|X_b \cap X_c|}{|X_b|}\right);$$
$$X_b = \bigl\{x_i\colon \beta_s(x_i) = b_s,\ s = 1, \dots, h\bigr\}, \qquad X^\ell = \bigsqcup_{b \in \{0,1\}^h} X_b.$$

Резюме: логические алгоритмы классификации

- Основные требования к логическим закономерностям:
  - интерпретируемость,
  - информативность,
  - различность.
- Основные виды композиций закономерностей:
  - решающие списки,
  - решающие деревья,
  - решающие леса,
  - взвешенное голосование.
- Yandex MatrixNet = градиентный бустинг над ODT.