

Логические алгоритмы классификации

К. В. Воронцов
vokov@forecsys.ru
<http://www.ccas.ru/voron>

23–30 апреля 2009

Определения и обозначения

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 $\varphi: X \rightarrow \{0, 1\}$, удовлетворяющий двум требованиям:

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

$$p_c(\varphi) = \#\{x_i: \varphi(x_i) = 1 \text{ и } y_i = c\} \rightarrow \max;$$

$$n_c(\varphi) = \#\{x_i: \varphi(x_i) = 1 \text{ и } y_i \neq c\} \rightarrow \min;$$

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

Содержание

- 1 Понятия закономерности и информативности**
 - Определения и обозначения
 - Интерпретируемость
 - Информативность
- 2 Методы поиска информативных закономерностей**
 - Жадный алгоритм
 - Алгоритмы на основе отбора признаков
 - Бинаризация данных
- 3 Композиции закономерностей**
 - Решающий список
 - Решающие деревья и леса
 - Голосование закономерностей

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 закономерностей

Параметрическое семейство *конъюнкций пороговых термов*:

$$\varphi(x) = \bigwedge_{j \in \omega} [\alpha_j \leq f_j(x) \leq \beta_j].$$

Параметрическое семейство *шаров*:

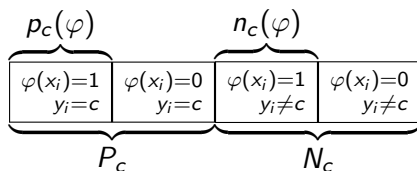
$$\varphi(x) = \left[\sum_{j \in \omega} \alpha_j |f_j(x) - f_j(x_0)|^\gamma \leq R_0 \right].$$

Параметрическое семейство *полуплоскостей*:

$$\varphi(x) = \left[\sum_{j \in \omega} \alpha_j f_j(x) \geq \alpha_0 \right].$$

Основная проблема — отбор признаков $\omega \subseteq \{1, \dots, n\}$.

Логический (эвристический) критерий закономерности



Определение

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

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

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

Проблема: хотелось бы иметь один скалярный критерий.

Нетривиальность проблемы свёртки двух критериев

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

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:

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

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

Определение

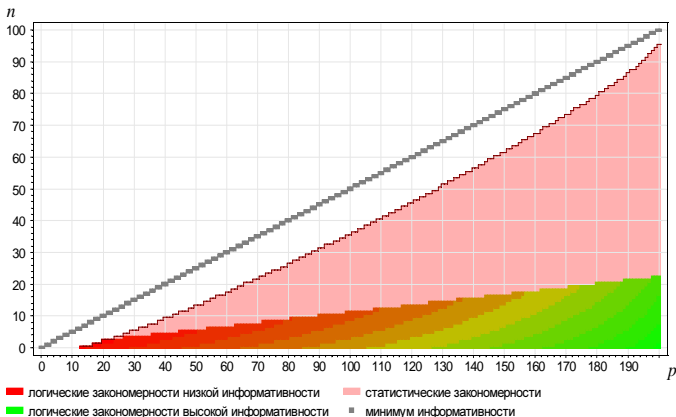
Информативность предиката $\varphi(x)$ относительно класса $c \in Y$:

$$I_c(\varphi, X^\ell) = -\ln h_{P_c, N_c}(p_c(\varphi), n_c(\varphi)).$$

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

Соотношение логического и статистического критериев

Области логических ε, δ -закономерностей (при $\varepsilon = 0.1$)
и статистических закономерностей (при $l_0 = 5$)
в координатах (p, n) при $P = 200, N = 100$.



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 — матожидание 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):

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

Соотношение статистического и энтропийного критериев

Определение

Предикат φ — закономерность по энтропийному критерию, если $IGain_c(\varphi, X^\ell) > G_0$ при некотором G_0 .

Теорема

Энтропийный критерий $IGain_c$ асимптотически эквивалентен статистическому I_c :

$$IGain_c(\varphi, X^\ell) \rightarrow \frac{1}{\ell \ln 2} I_c(\varphi, X^\ell) \quad \text{при } \ell \rightarrow \infty.$$

Доказательство:

применить формулу Стирлинга к статистическому критерию.

Задача перебора конъюнкций

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

$$\mathcal{K}_K[\mathcal{B}] = \{ \varphi(x) = \beta_1(x) \wedge \dots \wedge \beta_k(x) \mid \beta_1, \dots, \beta_k \in \mathcal{B}, k \leq K \}.$$

Число допустимых конъюнкций: $O(|\mathcal{B}|^K)$.

Семейство методов локального поиска

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

Частные случаи

- **жадный алгоритм:**
 $V(\varphi)$ — только добавления термов; $\varphi_0 = \emptyset$;
- **стохастический локальный поиск (SLS):**
 $V(\varphi)$ — случайное подмножество всевозможных добавлений, удалений, модификаций термов; $\varphi_0 = \emptyset$;
- **стабилизация:**
 $V(\varphi)$ — удаления термов или изменение параметров в термах; $\varphi_0 \neq \emptyset$;
- **редукция:**
 $V(\varphi)$ — только удаления термов; $\varphi_0 \neq \emptyset$;
 $I_c(\varphi, X^k)$ оценивается **по контрольной выборке** X^k .

Поиск закономерностей — это отбор признаков

Отличия от методов отбора признаков:

- вместо внешнего критерия $Q_{\text{ext}} \rightarrow \min$ критерий информативности $I_c \rightarrow \max$;
- вместо одного набора признаков строится множество закономерностей.

Все методы отбора признаков подходят:

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

Задача бинаризации вещественного признака

Дано:

вещественный признак $f(x)$;

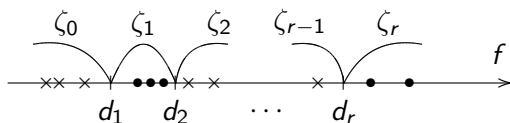
Построить:

наиболее информативное разбиение области значений признака на относительно небольшое число зон:

$$\zeta_0(x) = [f(x) < d_1];$$

$$\zeta_s(x) = [d_s \leq f(x) < d_{s+1}], \quad s = 1, \dots, r-1;$$

$$\zeta_r(x) = [d_r \leq f(x)].$$



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;

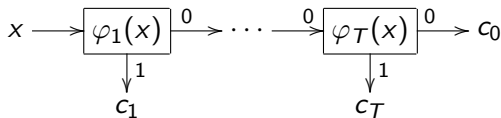
-
- 1: $D := \emptyset$; упорядочить выборку X^ℓ по возрастанию $f(x_i)$;
 - 2: **для всех** $i = 2, \dots, \ell$
 - 3: **если** $f(x_{i-1}) \neq f(x_i)$ и $[y_{i-1} = c] \neq [y_i = c]$ **то**
 - 4: добавить порог $\frac{1}{2}(f(x_{i-1}) + f(x_i))$ в конец D ;
 - 5: **повторять**
 - 6: **для всех** $d_j \in D, j = 1, \dots, |D| - 1$
 - 7: $\delta l_j := I_c(\zeta_{i-1} \vee \zeta_i \vee \zeta_{i+1}) - \max\{I_c(\zeta_{i-1}), I_c(\zeta_i), I_c(\zeta_{i+1})\}$;
 - 8: $i := \arg \max_s \delta l_s$;
 - 9: **если** $\delta l_j > \delta_0$ **то**
 - 10: слить зоны $\zeta_{i-1}, \zeta_i, \zeta_{i+1}$, удалив d_j и d_{j+1} из D ;
 - 11: **пока** $|D| > r + 1$.

Определение решающего списка

Решающий список (decision list, DL)

— алгоритм классификации $a: X \rightarrow Y$, который задаётся закономерностями $\varphi_1(x), \dots, \varphi_T(x)$ классов $c_1, \dots, c_T \in Y$:

- 1: **для всех** $t = 1, \dots, T$
- 2: **если** $\varphi_t(x) = 1$ **то**
- 3: **вернуть** c_t ;
- 4: **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: $\Phi' = \{\varphi \in \Phi : E_c(\varphi, U) \leq E_{\max}\}$;
 - 5: $\varphi_t := \arg \max_{\varphi \in \Phi'} I_c(\varphi, U)$;
 - 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** $|U| \leq \ell_0$ **то выход**;

Замечания к алгоритму построения решающего списка

- Параметр E_{\max} позволяет управлять сложностью списка:
 $E_{\max} \downarrow \Rightarrow p(\varphi_t) \downarrow, T \uparrow.$
- Стратегии выбора класса c_t :
 - 1) все классы по очереди;
 - 2) на каждом шаге определяется оптимальный класс:
$$(\varphi_t, c_t) := \arg \max_{\varphi \in \Phi', c \in Y} I_c(\varphi, U);$$
- 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.
- Гибкость: можно варьировать множество Φ .
- Допустимы разнотипные данные и данные с пропусками.

Недостатки:

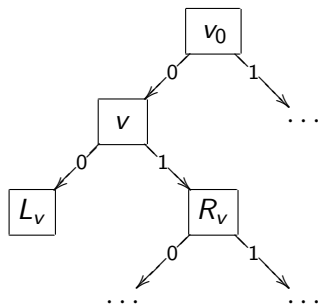
- При неудачном выборе Φ список может не построиться, будет много отказов от классификации.
- Список плохо интерпретируется, если он длинный и/или правила различных классов следуют попеременно.
- Каждый объект классифицируется только одним правилом, правила не могут компенсировать ошибки друг друга.

Определение бинарного решающего дерева

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

- 1) $\forall v \in V_{\text{внутр}} \rightarrow$ предикат $\beta_v : X \rightarrow \{0, 1\}$, $\beta \in \mathcal{B}$
- 2) $\forall v \in V_{\text{лист}} \rightarrow$ 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 .



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:
$$\beta := \arg \max_{\beta \in \mathcal{B}} I(\beta, U);$$
- 5: разбить выборку на две части $U = U_0 \cup U_1$ по предикату β :
$$U_0 := \{x \in U : \beta(x) = 0\};$$
$$U_1 := \{x \in U : \beta(x) = 1\};$$
- 6: **если** $U_0 = \emptyset$ или $U_1 = \emptyset$ **то**
- 7: **вернуть** новый лист v , $c_v := \text{Мажоритарный класс}(U)$;
- 8: создать новую внутреннюю вершину v : $\beta_v := \beta$;
построить левое поддерево: $L_v := \text{LearnID3}(U_0)$;
построить правое поддерево: $R_v := \text{LearnID3}(U_1)$;
- 9: **вернуть** v ;

Разновидности критериев ветвления

1. Отделение одного класса:

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

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 $P_c = \#\{x_i : y_i = c\}$, $h(z) \equiv -z \log_2 z$.

3. D-критерий:

$$I(\beta, X^\ell) = \#\{(x_i, x_j) : \beta(x_i) \neq \beta(x_j), y_i \neq y_j\}.$$

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

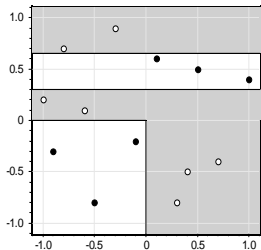
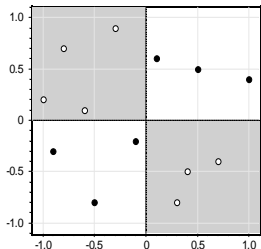
Достоинства:

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

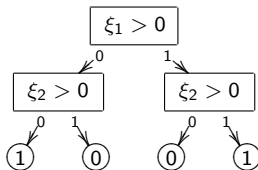
Недостатки:

- Жадный ID3 переусложняет структуру дерева.
- Чем дальше v от корня, тем меньше выборка $|U|$, тем меньше статистическая надёжность выбора β_v, c_v .
- Высокая чувствительность к составу выборки.
- Как правило, ID3 сильно переобучается.

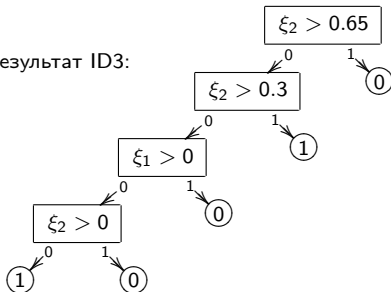
Жадный ID3 переусложняет структуру дерева



Оптимальное дерево для задачи XOR:



Результат ID3:



Стратегия пред-просмотра (look ahead)

Шаг 6:

если $U_0 = \emptyset$ или $U_1 = \emptyset$ **то**

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

Шаг 6 заменяется на более ресурсоёмкую процедуру:

для всех деревьев T глубины h

$r(U)$ = число ошибок дерева T на выборке U ;

вернуть новый лист v , $\beta_v :=$ корень лучшего поддеревя;

Достоинства:

- Задача XOR уже решается почти идеально.

Недостатки:

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

Стратегия пред-редукции (pre-pruning)

Шаг 6:

если $U_0 = \emptyset$ или $U_1 = \emptyset$ **то**
вернуть новый лист v ;

Шаг 6 заменяется на более мягкое условие:

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

Достоинства:

- Сразу строится более простое дерево.

Недостатки:

- Качество дерева может и не улучшиться.

Стратегия пост-редукции (post-pruning)

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

- 1: **для всех** $v \in V_{\text{внутр}}$
- 2: $S_v :=$ подмножество объектов X^k , дошедших до v ;
- 3: **если** $S_v = \emptyset$ **то**
- 4: **вернуть** новый лист v , $c_v := \text{Мажоритарный класс}(U)$;
- 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, $c_v := \arg \min_{c \in Y} r_c(v)$.

Преобразование решающего дерева в решающий список

- 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 из v_0 в v :

$$K_v(x) = \bigwedge_{(u, R_u)} \beta_u(x) \bigwedge_{(u, L_u)} \neg \beta_u(x).$$

- Редукция конъюнкций $K_v(x)$, $\forall v \in V_{\text{лист}}$
по контрольной выборке X^k .

Достоинства:

- Переобучение, как правило, уменьшается.

Недостатки:

- Преобразование в список необратимо: это уже не дерево.

Решающие леса

Случайные леса:

- T деревьев обучаются по случайным подвыборкам (bagging).
- Построение внутренних вершин, $\beta := \arg \max_{\beta \in \mathcal{B}} I(\beta, U)$:
 - \mathcal{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 объединению подвыборок, прошедших через ненадёжные листы предыдущего дерева.

Простое и взвешенное голосование

$\varphi_c^t(x)$, $t = 1, \dots, T_c$ — закономерности класса c .

Простое голосование (simple voting):

$$a(x) = \arg \max_{c \in Y} \underbrace{\frac{1}{T_c} \sum_{t=1}^{T_c} \varphi_c^t(x)}_{\Gamma_c(x)}.$$

Взвешенное голосование (weighted voting):

$$a(x) = \arg \max_{c \in Y} \underbrace{\sum_{t=1}^{T_c} \alpha_c^t \varphi_c^t(x)}_{\Gamma_c(x)}.$$

Отступ объекта x_i :

$$M(x_i) = \Gamma_{y_i}(x_i) - \max_{y \in Y \setminus y_i} \Gamma_y(x_i).$$

Бустинг закономерностей

Два класса, $Y = \{-1, +1\}$.

Композиция из $T = T_{-1} + T_{+1}$ закономерностей:

$$a_T(x) = \text{sign} \left(\underbrace{\sum_{t=1}^{T+1} \alpha_{+1}^t \varphi_{+1}^t(x)}_{\Gamma_{+1}(x)} - \underbrace{\sum_{t=1}^{T-1} \alpha_{-1}^t \varphi_{-1}^t(x)}_{\Gamma_{-1}(x)} \right), \quad \alpha_c^t > 0.$$

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:

$$\begin{aligned} Q_T &= \sum_{i=1}^{\ell} [\Gamma_{y_i}(x_i) - \Gamma_{-y_i}(x_i) < 0] \leq \\ &\leq \tilde{Q}_T = \sum_{i=1}^{\ell} \underbrace{\exp(\Gamma_{-y_i}(x_i) - \Gamma_{y_i}(x_i))}_{w_i} = \sum_{i=1}^{\ell} w_i. \end{aligned}$$

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

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

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

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

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

Алгоритм AdaBoost для закономерностей

Вход: выборка X^ℓ ; семейство предикатов Φ ; параметры T, λ ;
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: $\varphi_c^t := \arg \max_{\varphi \in \Phi} \sqrt{p_c^w(\varphi)} - \sqrt{n_c^w(\varphi)}$;
- 5: $\alpha_c^t := \frac{1}{2} \ln \frac{p_c^w(\varphi)}{n_c^w(\varphi) + \lambda}$;
- 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: $Z := \frac{1}{\ell} \sum_{i=1}^{\ell} w_i$; $w_i := w_i / Z, i = 1, \dots, \ell$;

Алгоритм AdaBoost для закономерностей

Достоинства:

- Переобучение, как правило, не увеличивается с ростом T за счёт максимизации и выравнивания отступов.
- Универсальность: можно использовать любое семейство Φ .
- Возможность выделения и фильтрации шумовых объектов.
- Низкая чувствительность к пропускам в данных.

Недостатки:

- В некоторых задачах переобучение всё же наблюдается, хотя, при очень больших $T \sim 10^2 \div 10^3$.
- Утрата интерпретируемости при больших T .