

# Методы машинного обучения. Градиентный бустинг

Воронцов Константин Вячеславович  
[www.MachineLearning.ru/wiki?title=User:Vokov](http://www.MachineLearning.ru/wiki?title=User:Vokov)  
вопросы к лектору: [k.vorontsov@iai.msu.ru](mailto:k.vorontsov@iai.msu.ru)

материалы курса:  
[github.com/MSU-ML-COURSE/ML-COURSE-25-26](https://github.com/MSU-ML-COURSE/ML-COURSE-25-26)  
орг.вопросы по курсу: [mlcmc@mail.ru](mailto:mlcmc@mail.ru)

## 1 Алгоритмы градиентного бустинга

- Градиентный бустинг Фридмана
- Стохастический градиентный бустинг
- Алгоритм XGBoost

## 2 Алгоритм CatBoost

- Упорядоченный бустинг
- Упорядоченный бустинг
- Категориальные признаки

## 3 Базовые алгоритмы

- Небрежные решающие деревья
- Логические закономерности и АВО
- Бинаризация признаков

## Напоминание. Взвешенное голосование и AdaBoost

**Дано:**  $X^\ell = (x_i, y_i)_{i=1}^\ell \subset X \times Y$ ,  $b_t: X \rightarrow \{\pm 1, 0\}$ ,  $Y = \{\pm 1\}$

**Найти** линейный ансамбль базовых алгоритмов  $b_t$  из модели  $\mathcal{B}$ :

$$a_T(x) = \sum_{t=1}^T \alpha_t b_t(x), \quad x \in X, \quad \alpha_t \geq 0$$

**Эвристика:** обучаем  $a_T, b_T$  при фиксированных предыдущих

**Критерий** минимума числа ошибок ансамбля на обучении:

$$Q_T = \sum_{i=1}^{\ell} [y_i a_T(x_i) < 0] \leq \sum_{i=1}^{\ell} \exp(-y_i a_T(x_i)) \rightarrow \min_{a_T, b_T}$$

**Ограничения** (недостатки) AdaBoost:

- $Y = \{\pm 1\}$ , хотелось бы расширить класс решаемых задач
- $b_t: X \rightarrow \{\pm 1, 0\}$ , хотелось бы  $b_t: X \rightarrow \mathbb{R}$
- $[M \leq 0] \leq e^{-M}$ , хотелось бы обобщать вид потерь  $\mathcal{L}(a, y)$

## Градиентный бустинг для произвольной функции потерь

**Дано:**  $X^\ell = (x_i, y_i)_{i=1}^\ell \subset X \times Y$ ,  $b_t: X \rightarrow \mathbb{R}$

**Найти** линейный ансамбль базовых алгоритмов  $b_t$  из модели  $\mathcal{B}$ :

$$a_T(x) = \sum_{t=1}^T \alpha_t b_t(x), \quad x \in X, \quad \alpha_t \geq 0$$

**Эвристика:** обучаем  $a_T, b_T$  при фиксированных предыдущих

**Критерий** с заданной гладкой функцией потерь  $\mathcal{L}(a, y)$ :

$$Q(\alpha_T, b_T) = \sum_{i=1}^{\ell} \mathcal{L}\left(\underbrace{\sum_{t=1}^{T-1} \alpha_t b_t(x_i)}_{a_{T-1}(x_i)} + \alpha_T b_T(x_i), y_i\right) \rightarrow \min_{\alpha_T, b_T},$$

$$v_{T,i} = a_T(x_i)$$

$v_t = (a_t(x_i))_{i=1}^\ell$  — вектор ответов ансамбля  $a_t$  на всей выборке

G.Friedman. Greedy function approximation: a gradient boosting machine. 1999.

## Параметрическая аппроксимация градиентного шага

Градиентный метод минимизации  $Q(v) \rightarrow \min, v \in \mathbb{R}^\ell$ :

$v_{0,i}$  := начальное приближение

$v_{T,i} := v_{T-1,i} - \alpha g_i, i = 1, \dots, \ell$

$g_i = \mathcal{L}'_a(v_{T-1,i}, y_i)$  — компоненты вектора градиента,  
 $\alpha$  — градиентный шаг.

Это похоже на добавление базового алгоритма в ансамбль:

$a_T(x_i) := a_{T-1}(x_i) + \alpha b(x_i), i = 1, \dots, \ell$

Идея: будем искать такой базовый алгоритм  $b_T$  из  $\mathcal{B}$ , чтобы вектор  $(b_T(x_i))_{i=1}^\ell$  приближал вектор антиградиента  $(-g_i)_{i=1}^\ell$ :

$$b_T := \arg \min_{b \in \mathcal{B}} \sum_{i=1}^{\ell} (b(x_i) + g_i)^2$$

## Алгоритм градиентного бустинга (Gradient Boosting)

**Вход:** обучающая выборка  $X^\ell$ ; **параметр  $T$** ;

**Выход:** базовые алгоритмы и их веса  $\alpha_t b_t$ ,  $t = 1, \dots, T$ ;

$$b_1 := \arg \min_{b \in \mathcal{B}} \sum_{i=1}^{\ell} \mathcal{L}(b(x_i), y_i);$$

$$\alpha_1 := 1; \quad v_i := b_1(x_i), \quad i = 1, \dots, \ell;$$

**для всех**  $t = 2, \dots, T$

базовый алгоритм, приближающий антиградиент:

$$b_t := \arg \min_{b \in \mathcal{B}} \sum_{i=1}^{\ell} (b(x_i) + \mathcal{L}'(v_i, y_i))^2;$$

задача одномерной минимизации:

$$\alpha_t := \arg \min_{\alpha > 0} \sum_{i=1}^{\ell} \mathcal{L}(v_i + \alpha b_t(x_i), y_i);$$

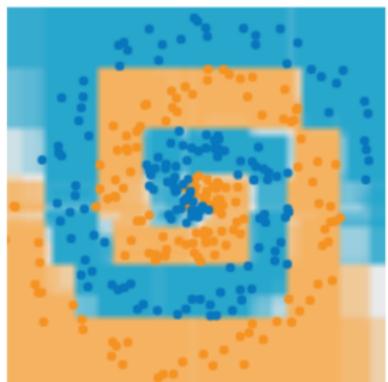
обновление вектора значений на объектах выборки:

$$v_i := v_i + \alpha_t b_t(x_i); \quad i = 1, \dots, \ell;$$

## Пример. Классификация синтетической выборки

100 деревьев глубины 5

Prediction:



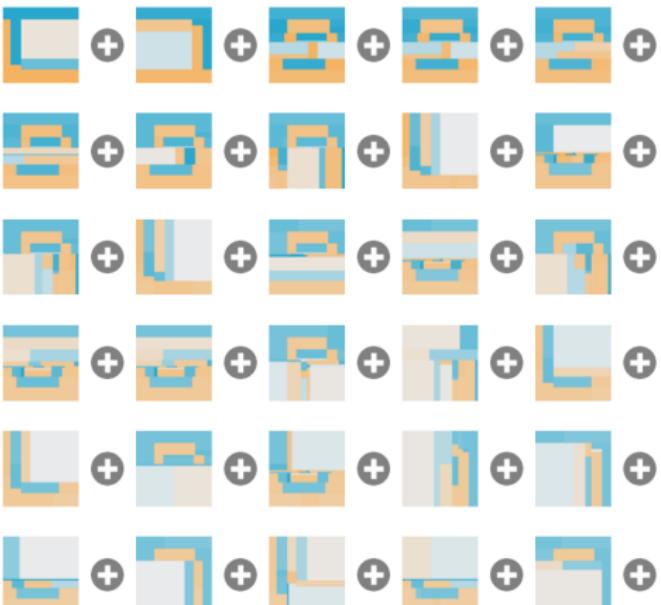
↑  
predictions of GB (all 100 trees)

train loss: 0.022

test loss: 0.218



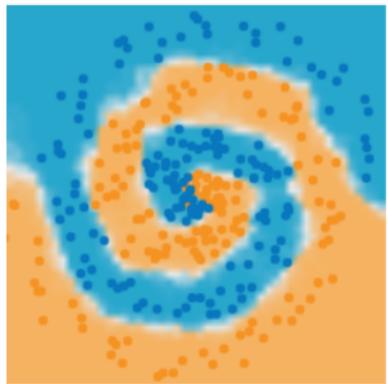
Decision functions of first 30 trees



## Пример. Классификация синтетической выборки

100 деревьев глубины 5, с подбором вращения каждого дерева

Prediction:



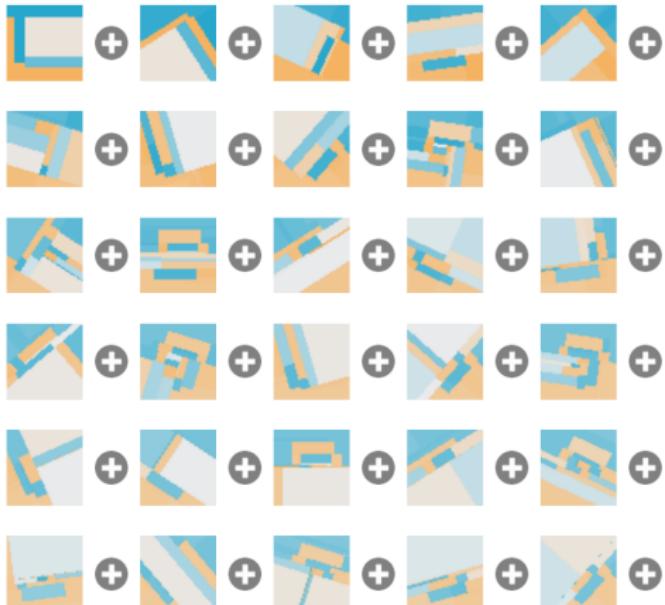
↑  
predictions of GB (all 100 trees)

train loss: 0.013

test loss: 0.092



Decision functions of first 30 trees



## Частные случаи GB: регрессия, AdaBoost и другие

**Регрессия:**  $\mathcal{L}(a, y) = (a - y)^2$

- $b_T(x)$  обучается на разностях  $y_i - \sum_{t=1}^{T-1} \alpha_t b_t(x_i)$
- если регрессии  $b_t$  линейные, то  $\alpha_t$  можно не обучать.

**Бинарная классификация:**  $\mathcal{L}(a, y) = e^{-ay}, \quad b_t \in \{\pm 1, 0\}$

- GB в точности совпадает с AdaBoost [Freund, 1995]

**Бинарная классификация:**  $\mathcal{L}(a, y) = \mathcal{L}(ay), \quad b_t \in \mathbb{R}$

- GB совпадает с AnyBoost [Mason, 2000],  $\mathcal{L}$  убывающая

**Многоклассовая задача:**  $a(x) = \arg \max_{y \in Y} a_y(x), \quad b_t \in \mathbb{R}^Y$ ,

$$\mathcal{L}(a, y) = \mathcal{L}(a_y - \max_{z \neq y} a_z) \text{ или } \mathcal{L}(a, y) = -\ln(e^{a_y} / \sum_{z \in Y} e^{a_z})$$

- GB строит  $b_{ty}$  для всех  $y \in Y, \quad b_t(x) = \arg \max_{y \in Y} b_{ty}(x)$

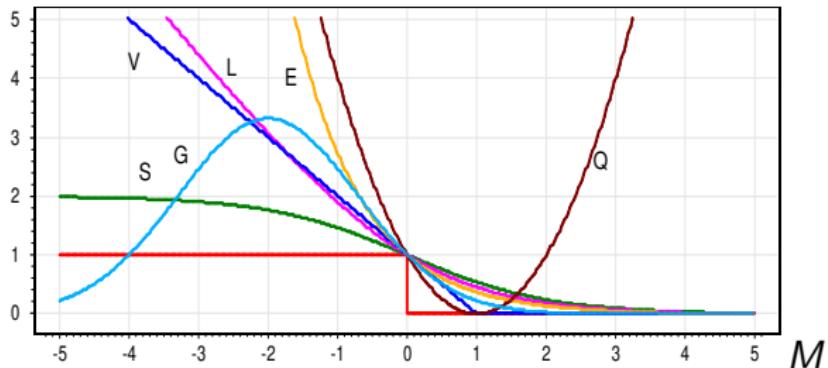
---

*Y.Freund, R.Schapire. A decision-theoretic generalization of online learning and an application to boosting. 1995.*

*L.Mason et al. Boosting algorithms as gradient descent. 2000.*

## Варианты бустинга для двухклассовой классификации

Гладкие аппроксимации пороговой функции потерь [ $M < 0$ ]:



$E(M) = e^{-M}$  — экспоненциальная (AdaBoost);

$L(M) = \log_2(1 + e^{-M})$  — логарифмическая (LogitBoost);

$Q(M) = (1 - M)^2$  — квадратичная (GentleBoost);

$G(M) = \exp(-cM(M + s))$  — гауссовская (BrownBoost);

$S(M) = 2(1 + e^M)^{-1}$  — сигмоидная;

$V(M) = (1 - M)_+$  — кусочно-линейная (из SVM);

## Стохастический градиентный бустинг (SGB)

**Идея:** при оптимизации  $b_t$  и  $\alpha_t$  использовать не всю выборку  $X^\ell$ , а случайную подвыборку, по аналогии с бэггингом

### Преимущества:

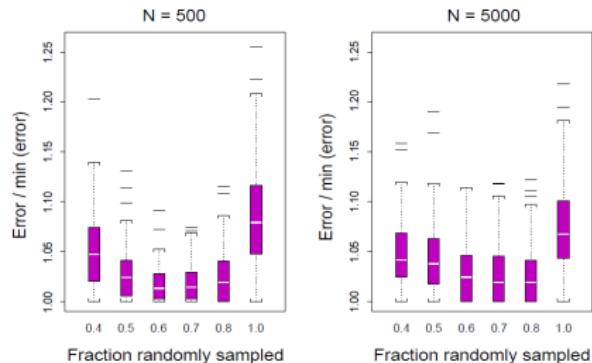
- улучшается сходимость, уменьшается время обучения
- улучшается обобщающая способность ансамбля
- можно использовать несмещённые оценки out-of-bag

### Эксперименты:

относительная ошибка при различном объёме выборки  $N$

### Вывод:

оптимально сэмплировать  
около 60–80% выборки



Friedman G. Stochastic Gradient Boosting. 1999.

## XGBoost: популярная и быстрая реализация GB над деревьями

Деревья регрессии и классификации (CART):

$$b(x, w) = \sum_{k \in K} w_k B_k(x)$$

где  $B_k(x)$  — бинарный индикатор [ $x$  попадает в лист  $k$ ],  
 $w_k$  — значение в листе  $k$ ,  $K$  — множество листьев дерева.  
Для любого  $x$  одно и только одно слагаемое не равно нулю.

Критерий качества с суммой  $L_0$  и  $L_2$  регуляризаторов:

$$Q(w) = \sum_{i=1}^{\ell} \mathcal{L}(a(x_i) + b(x_i, w), y_i) + \gamma|K| + \frac{\lambda}{2} \sum_{k \in K} w_k^2 \rightarrow \min_w,$$

где  $a(x_i) = \sum_{t=1}^{T-1} \alpha_t b_t(x_i)$  — ранее построенная часть ансамбля.

В некоторых случаях задача имеет аналитическое решение.

## XGBoost: приближённое аналитическое решение для $w_j$

Приблизим  $\mathcal{L}(a + b, y) \approx \mathcal{L}(a, y) + b\mathcal{L}'(a, y) + \frac{b^2}{2}\mathcal{L}''(a, y)$ :

$$Q(w) \approx \tilde{Q}(w) = \sum_{i=1}^{\ell} \left( g_i b_i + \frac{1}{2} h_i b_i^2 \right) + \gamma |K| + \frac{\lambda}{2} \sum_{k \in K} w_k^2 \rightarrow \min_w$$

где  $b_i^P = \sum_k w_k^P B_k(x_i)$ ,  $g_i = \mathcal{L}'(a(x_i), y_i)$ ,  $h_i = \mathcal{L}''(a(x_i), y_i)$ .

Из условий  $\frac{\partial \tilde{Q}(w)}{\partial w_k} = 0$  находим оптимальное значение листа  $k$ :

$$w_k = -\frac{\sum_i g_i B_k(x_i)}{\lambda + \sum_i h_i B_k(x_i)}$$

Подставляя  $w_k$  обратно в  $\tilde{Q}(w)$ , выводим критерий ветвления:

$$\tilde{Q}(B_1, \dots, B_k) = -\frac{1}{2} \sum_{k \in K} \frac{(\sum_i g_i B_k(x_i))^2}{\lambda + \sum_i h_i B_k(x_i)} + \gamma |K| \rightarrow \min$$

## XGBoost и другие варианты GB

Преимущества XGBoost (eXtreme Gradient Boosting):

- $L_2$  регуляризация сокращает переобучение
- $L_0$  регуляризация упрощает деревья (pruning)
- как и общий GB, допускает произвольные функции потерь
- очень быстрая реализация за счёт аналитических формул
- имеет механизм обработки пропущенных значений

Что ещё бывает:

- Light GBM — для обучения на сверхбольших данных
- Яндекс.MatrixNet — GB над Oblivious Decision Tree
- Яндекс.CatBoost — для категориальных признаков

---

Tianqi Chen, Carlos Guestrin. XGBoost: A scalable tree boosting system. 2016.  
G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu.  
LightGBM: A highly efficient gradient boosting decision tree. 2017

## Основные мотивации CatBoost

### Две проблемы:

- Надо обрабатывать категориальные признаки с большим числом редких значений (пользователь, регион, город, реклама, рекламодатель, товар, документ, автор, и т.д.)
- Переобучение (смещённость, target leakage) в градиентах:  $g_i = \mathcal{L}'(a_{t-1}(x_i), y_i)$  вычисляются в тех же точках  $x_i$ , по которым ансамбль  $a_{t-1}(x)$  обучался аппроксимировать  $y_i$

Приём, похожий на Out-Of-Bag и на онлайновые методы:

- для получения несмещённых оценок на объекте  $x_i$  хранить и дообучать ансамбль на выборках без этого объекта
- как сделать, чтобы этих выборок было  $O(\log \ell)$ , а не  $O(\ell)$ ?
- как сделать, чтобы они не сильно перекрывались?

## Упорядоченный бустинг (ordered boosting)

Идеи:

- вычислять  $g_i$  по модели  $a_{t-1}$ , которая не обучалась на  $x_i$
- строить обучающие подвыборки удваивающейся длины
- построить много таких случайно перемешанных выборок

Обозначения:

$\sigma_1, \dots, \sigma_s$  — случайные перестановки выборки  $X^\ell$

$X^{rj}$  — подвыборка первых  $2^j$  объектов из  $\sigma_r(X^\ell)$

$a_t^{rj}(x)$  — ансамбль-полуфабрикат, обученный по  $X^{rj}$

$g_{ti}^r = -\mathcal{L}'(a_{t-1}^{rj}(x_i), y_i)$  — антиградиент в точке  $(x_i, y_i)$  для ансамбля  $a_{t-1}^{rj}$ , который по ней не обучался,  $j = \lfloor \log_2(i-1) \rfloor$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1		2														4		

$j = 3$

$i = 13$

## Модификация градиентного бустинга

сгенерировать случайные перестановки  $\sigma_0, \sigma_1, \dots, \sigma_s$ ;

**для всех**  $t = 1, \dots, T$ :

выбрать перестановку  $\sigma_r$  случайно из  $\sigma_1, \dots, \sigma_s$ ;

$g_{ti}^r := -\mathcal{L}'(a_{t-1}^{rj}(x_i), y_i)$  — несмещённый антиградиент;

$b_t := \arg \min_b \sum_{i=1}^{\ell} (b(x_i) - g_{ti}^r)^2$ ;

**для всех** деревьев  $b_t^{rj}$ ,  $r = 1, \dots, s$ ,  $2^j \leq \ell$ :

скопировать общую для них структуру дерева из  $b_t$ ;

вычислить в листьях  $b_t^{rj}$  средние по  $\{g_{ti}^r : x_i \in X^{rj}\}$ ;

вычислить в листьях  $b_t$  средние по  $\{g_{ti}^0 : x_i \in X^{0j}\}$ ;

GB: вычислить  $\alpha_t$  и обновить  $a_{t,i} := a_{t-1,i} + \alpha_t b_t(x_i)$ ;

## Способы обработки категориальных признаков

Пусть  $V$  — множество (словарь) значений признака  $f(x)$

Стандартные методы либо громоздкие, либо переобучаются:

- бинаризация (one-hot encoding):  $b_v(x) = [f(x) = v]$
- группирование (кластеризация) значений (LightGBM)
- статистика по целевому признаку (target statistics, TS):

$$\tilde{f}(x_i) = \frac{\sum_{k=1}^{\ell} [f(x_k) = f(x_i)] y_k + \gamma p}{\sum_{k=1}^{\ell} [f(x_k) = f(x_i)] + \gamma}$$

CatBoost:

- статистика TS вычисляется по перестановкам  $X^{rj}$ :  
$$\tilde{f}(x_i) = \frac{\sum_{x_k \in X^{rj}} [f(x_k) = f(x_i)] y_k + \gamma p}{\sum_{x_k \in X^{rj}} [f(x_k) = f(x_i)] + \gamma}, \quad j = \lfloor \log_2(i - 1) \rfloor$$
- конъюнкции категориальных признаков создаются «налету» в процессе построения деревьев

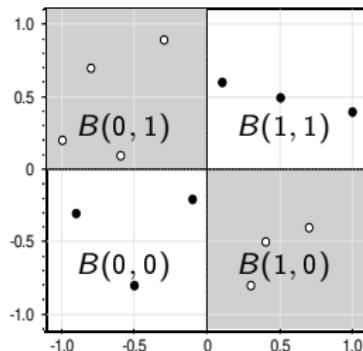
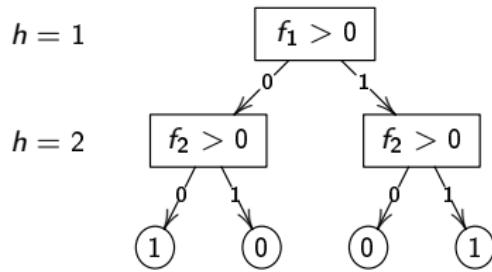
## Небрежные решающие деревья (Oblivious Decision Tree, ODT)

**Решающая таблица:** бинарное дерево глубины  $H$ ;  
для всех узлов уровня  $h$  условие ветвления  $f_h(x)$  одинаково;  
на уровне  $h$  число вершин  $2^{h-1}$ ; число листьев  $2^H$ .

Классификатор задаётся таблицей решений  $B: \{0, 1\}^H \rightarrow Y$ :

$$b(x) = B(f_1(x), \dots, f_H(x)).$$

**Пример:** задача XOR,  $H = 2$ .



## Жадный алгоритм обучения ODT

**Вход:** выборка  $X^\ell$ ; множество признаков  $F$ ; глубина дерева  $H$ ;

**Выход:** признаки  $f_h$ ,  $h = 1, \dots, H$ ; таблица  $B: \{0, 1\}^H \rightarrow Y$ ;  
**для всех**  $h = 1, \dots, H$

признак с максимальным выигрышем определённости:

$$f_h := \arg \max_{f \in \text{bin}\{F\}} \text{Gain}(f_1, \dots, f_{h-1}, f);$$

$$B(\beta) := \begin{cases} \text{Major}(U_{H\beta}), \text{ мажоритарное правило в алгоритме ID3} \\ \text{avg}\{y_i : x_i \in U_{H\beta}\}, \text{ в алгоритме CART} \\ \text{avg}\{g_{ti}^r : x_i \in U_{H\beta}\}, \text{ в алгоритме CatBoost} \end{cases}$$

$U_{h\beta} = \{x_i \in X^\ell : f_s(x_i) = \beta_s, s = 1..h\}$  — выборка объектов  $x_i$ , дошедших до вершины  $\beta = (\beta_1, \dots, \beta_h) \in \{0, 1\}^h$  уровня  $h$

Выигрыш от ветвления на уровне  $h$  по всей выборке  $X^\ell$ :

$$\text{Gain}(f_1, \dots, f_h) = Q(X^\ell) - \sum_{\beta \in \{0, 1\}^h} \frac{|U_{h\beta}|}{\ell} Q(U_{h\beta})$$

## Напоминание. Конъюнктивные логические закономерности

Семейство интерпретируемых логических правил:

$$b(x) = \bigwedge_{j \in J} [\alpha_j \leq f_j(x) \leq \beta_j]$$

$J \subset \{1, \dots, n\}$  — подмножество небольшого числа признаков  
 $[\alpha_j, \beta_j]$  — отрезок значений признака  $f_j$

Информативность предиката  $b(x)$  относительно класса  $y \in Y$ :

$$\begin{cases} p_y(b) = \#\{x_i : b(x_i) = 1 \text{ и } y_i = y\} \rightarrow \max_b \\ n_y(b) = \#\{x_i : b(x_i) = 1 \text{ и } y_i \neq y\} \rightarrow \min_b \end{cases}$$

Критерий обучения базовых алгоритмов (из AdaBoost):

$$\sqrt{p_y(b)} - \sqrt{n_y(b)} \rightarrow \max_b$$

---

R.E.Schapire, Y.Singer. Improved boosting using confidence-rated predictions. 1999

## Алгоритмы Вычисления Оценок (АВО)

Бинарная функция сходства по набору признаков  $J$ :

$$b(x) = B_J(x, x_i) = \bigwedge_{j \in J} [ |f_j(x) - f_j(x_i)| \leq \varepsilon_j ]$$

$J \subset \{1, \dots, n\}$  — подмножество небольшого числа признаков  
 $x_i \in X^\ell$  — эталонный объект из обучающей выборки  
 $\varepsilon_j$  — порог сходства объектов по признаку  $f_j$

Преимущества:

- подходит для задач с малой обучающей выборкой
- объединяет принципы голосования, сходства и поиска логических правил в информативных подпространствах

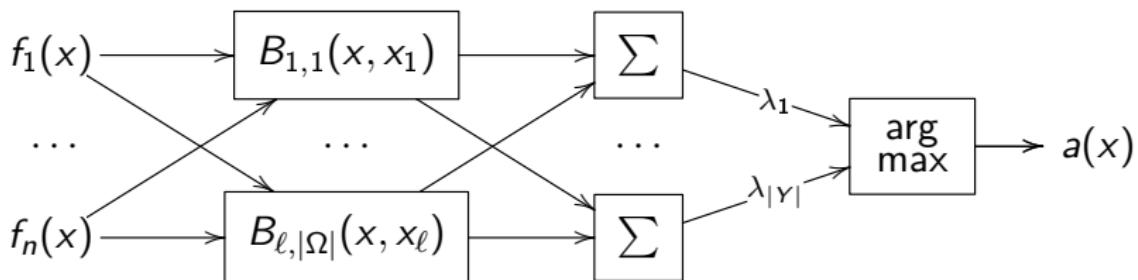
---

Дмитриев А. Н., Журавлев Ю. И., Кренделев Ф. П. Об одном принципе классификации и прогноза геологических объектов и явлений. 1968.  
Журавлёв Ю. И., Никифоров В. В. Алгоритмы распознавания, основанные на вычислении оценок. 1971.

## АВО объединяет многие эвристические принципы

Трёхслойная нейросеть RBF (Radial Basis Function):

$$a(x) = \arg \max_{y \in Y} \lambda_y \sum_{i,\omega} [y_i = y] w_{i\omega} B_{i\omega}(x, x_i)$$



- ≈ метрический метод с ядрами  $B_{i\omega}(x, x_i) = K(\frac{1}{h_{i\omega}} \rho_{i\omega}(x, x_i))$
- ≈ линейный классификатор SVM с радиальным ядром
- ≈ байесовский классификатор с плотностями-смесями  $p(x|y)$
- ≈ отбор эталонов:  $w_{i\omega} = 0$  для не-эталонов  $x_i$
- ≈ отбор признаков в сферических логических закономерностях

## Принципы информативности, непротиворечивости, тупиковости

- **информативность** предиката  $b(x)$  класса  $y \in Y$ :

$$\begin{cases} p_y(b) = \#\{x_i : b(x_i)=1 \text{ и } y_i=y\} \rightarrow \max \\ n_y(b) = \#\{x_i : b(x_i)=1 \text{ и } y_i \neq y\} \rightarrow \min \end{cases}$$

- **информативность** функции сходства  $B(x, x')$ :

$$\begin{cases} p(B) = \#\{(x_i, x_j) : B(x_i, x_j)=1 \text{ и } y_i=y_j\} \rightarrow \max \\ n(B) = \#\{(x_i, x_j) : B(x_i, x_j)=1 \text{ и } y_i \neq y_j\} \rightarrow \min \end{cases}$$

- **непротиворечивость**:  $n_y(b) = 0$ ,  $n(B) = 0$

— тест  $J$ :  $B_J(x_i, x_j) = 0$ ,  $\forall i, j$ :  $y_i \neq y_j$

— представительный набор  $(J, i)$ :  $B_J(x_i, x_j) = 0$ ,  $\forall j$ :  $y_i \neq y_j$

- **тупиковость**: никакое подмножество признаков  $J' \subset J$  не является тестом (или представительным набором), т. е. станет  $n(B) > 0$ , если выкинуть хотя бы один признак

---

Журавлёв Ю. И., Никифоров В. В. Алгоритмы распознавания, основанные на вычислении оценок, 1971.

## Вспомогательная задача бинаризации вещественного признака

Цель — сократить перебор предикатов вида  $[f(x) \leq \alpha]$ .

Дано: выборка значений признака  $f(x_i) \in \mathbb{R}$ ,  $x_i \in X^\ell$ .

Найти: разбиение области значений признака на зоны:

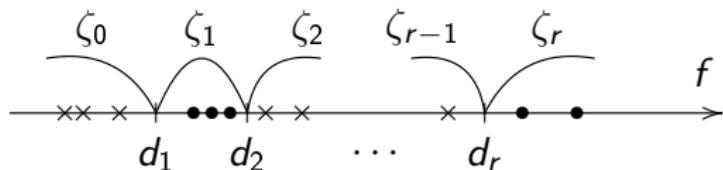
$$\zeta_0(x) = [f(x) < d_1];$$

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

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

Критерий: максимум информативности при минимуме  $r$ .

Пороги  $d_i$  нет смысла ставить между точками одного класса:



## Способы разбиения области значений признака на зоны

- ① Разбиение на квантили (равномощные подвыборки)
- ② Разбиение по равномерной сетке «удобных» значений
- ③ Жадная максимизация информативности путём слияний
- ④ Объединение нескольких разбиений

### Выбор «удобных» пороговых значений

**Задача:** на отрезке  $[a, b]$  найти значение  $x^*$  с минимальным числом значащих цифр.

Если таких  $x^*$  несколько, выбрать из них наиболее близкий к середине отрезка:

$$x^* = \arg \min_x \left| \frac{1}{2}(a + b) - x \right|.$$

$a = 2,16667$
2,19
$x^* = 2,2$
2,21
$(a+b)/2 = 2,23889$
2,29
2,3
2,31
$b = 2,31111$

## Жадный алгоритм слияния зон по критерию информативности

**Вход:** выборка  $X^\ell$ ; параметры  $r$  и  $\delta_0$ ;

**Выход:**  $D = \{d_1 < \dots < d_r\}$  — последовательность порогов;

$D := \emptyset$ ; упорядочить выборку  $X^\ell$  по возрастанию  $f(x_i)$ ;

**для всех**  $i = 2, \dots, \ell$

**если**  $f(x_{i-1}) \neq f(x_i)$  и  $y_{i-1} \neq y_i$  **то**

**добавить** порог  $\frac{1}{2}(f(x_{i-1}) + f(x_i))$  в конец  $D$

**повторять**

**для всех**  $d_i \in D$ ,  $i = 1, \dots, |D| - 1$

$\delta I_i := I(\zeta_{i-1} \vee \zeta_i \vee \zeta_{i+1}) - \max\{I(\zeta_{i-1}), I(\zeta_i), I(\zeta_{i+1})\}$ ;

$i := \arg \max_s \delta I_s$ ;

**если**  $\delta I_i > \delta_0$  **то**

**слить** зоны  $\zeta_{i-1}$ ,  $\zeta_i$ ,  $\zeta_{i+1}$ , удалив  $d_i$  и  $d_{i+1}$  из  $D$ ;

**пока**  $|D| > r + 1$ ;

- Ансамбли позволяют решать сложные задачи, которые плохо решаются отдельными базовыми алгоритмами
- Важное открытие середины 90-х: обобщающая способность бустинга не ухудшается с ростом сложности  $T$
- Градиентный бустинг — наиболее общий из всех бустингов:
  - произвольная функция потерь
  - произвольное пространство оценок  $R$
  - подходит для регрессии, классификации, ранжирования
- Чаще всего GB применяется к решающим деревьям
- RF и SGB — универсальные модели машинного обучения
- CatBoost — общедоступная реализация от Яндекса:
  - для категориальных признаков
  - для уменьшения переобучения
- АВО — старинный рецепт для малых данных:
  - голосование низкоразмерных непротиворечивых правил,
  - определяемых не на объектах, а на парах объектов