Методы машинного обучения. Градиентный бустинг

Воронцов Константин Вячеславович
www.MachineLearning.ru/wiki?title=User:Vokov
вопросы к лектору: k.vorontsov@iai.msu.ru

материалы курса:
github.com/MSU-ML-COURSE/ML-COURSE-25-26
орг.вопросы по курсу: ml.cmc@mail.ru

ВМК МГУ • 18 ноября 2025

Содержание

Напоминание. Взвешенное голосование и AdaBoost

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

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

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

Эвристика: обучаем α_T, b_T при фиксированных предыдущих

Критерий минимума числа ошибок ансамбля на обучении:

$$Q_T = \sum_{i=1}^{\ell} \bigl[y_i a_T(x_i) < 0\bigr] \leqslant \sum_{i=1}^{\ell} \exp\bigl(-y_i a_T(x_i)\bigr) \to \min_{\alpha_T, b_T}$$

Ограничения (недостатки) AdaBoost:

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

Y.Freund, R.E.Schapire. A short introduction to boosting. 1999.

Градиентный бустинг для произвольной функции потерь

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

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

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

Эвристика: обучаем α_T, b_T при фиксированных предыдущих

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

$$Q(\alpha_T, b_T) = \sum_{i=1}^{\ell} \mathscr{L}\Biggl(\underbrace{\underbrace{\sum_{t=1}^{T-1} \alpha_t b_t(x_i)}_{a_{T-1}(x_i)} + \alpha_T b_T(x_i)}_{v_{T,i} = a_T(x_i)}, y_i\Biggr) \to \min_{\alpha_T, b_T},$$

$v_t = \bigl(a_t(x_i)\bigr)_{i=1}^\ell$ — 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 $Q(v) \to \min$, $v \in \mathbb{R}^\ell$:

$$v_{0,i} := \text{начальное приближение}$$
$$v_{T,i} := v_{T-1,i} - \alpha g_i, \quad i = 1, \dots, \ell$$

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

Это похоже на добавление базового алгоритма в ансамбль:

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

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

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

Алгоритм градиентного бустинга (Gradient Boosting)

Вход: обучающая выборка X^ℓ; параметр T;

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

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

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

для всех $t = 2, \dots, T$

- базовый алгоритм, приближающий антиградиент:

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

- задача одномерной минимизации:

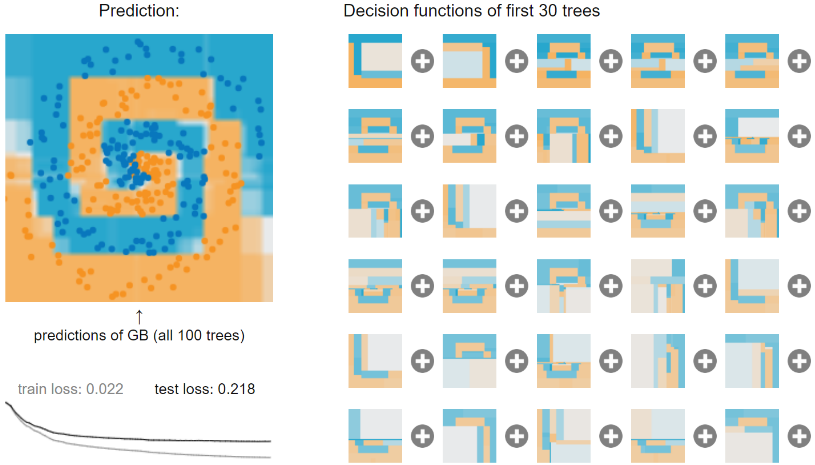
  $$\alpha_t := \arg\min_{\alpha > 0} \sum_{i=1}^{\ell} \mathscr{L}\bigl(v_i + \alpha b_t(x_i), y_i\bigr);$$

- обновление вектора значений на объектах выборки:

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

Пример. Классификация синтетической выборки

100 деревьев глубины 5

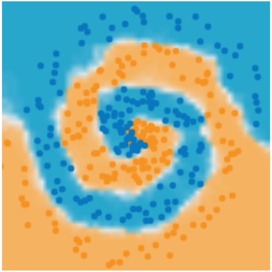


http://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html

Пример. Классификация синтетической выборки

100 деревьев глубины 5, с подбором вращения каждого дерева

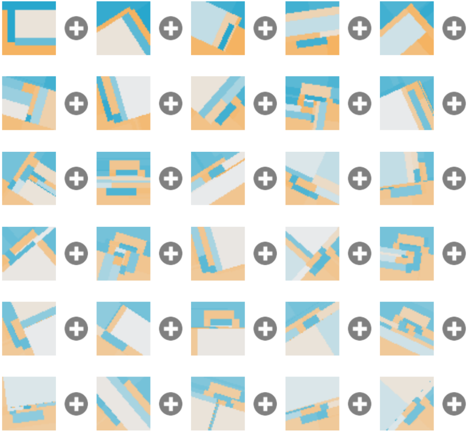
Prediction:



↑
predictions of GB (all 100 trees)

train loss: 0.013 test loss: 0.092

Decision functions of first 30 trees



http://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html

Частные случаи GB: регрессия, AdaBoost и другие

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

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

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

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

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

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

Многоклассовая задача: $a(x) = \arg\max\limits_{y\in Y} a_y(x)$, $\; b_t \in \mathbb{R}^Y$,

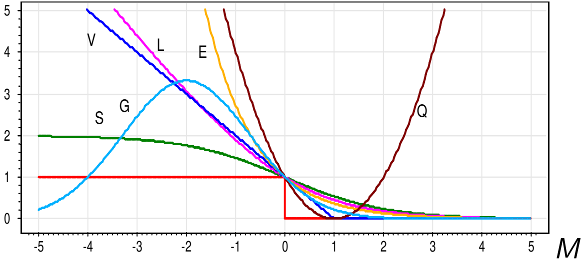
$$\mathscr{L}(a,y) = \mathcal{L}\bigl(a_y - \max_{z\neq y} a_z\bigr) \text{ или } \mathscr{L}(a,y) = -\ln\Bigl(e^{a_y} \Big/ \sum_{z\in Y} e^{a_z}\Bigr)$$

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

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);
$G(M) = \exp\bigl(-cM(M+s)\bigr)$ — гауссовская (BrownBoost);
$S(M) = 2(1 + e^M)^{-1}$ — сигмоидная;
$V(M) = (1 - M)_+$ — кусочно-линейная (из SVM);

Стохастический градиентный бустинг (SGB)

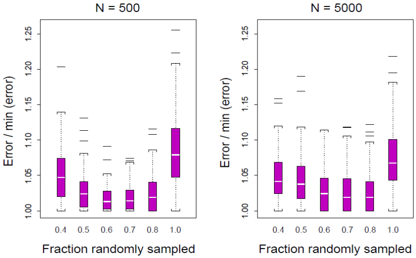
Идея: при оптимизации b_t и α_t использовать не всю выборку X^ℓ, а случайную подвыборку, по аналогии с бэггингом

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:

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

где $a(x_i) = \sum\limits_{t=1}^{T-1} \alpha_t b_t(x_i)$ — ранее построенная часть ансамбля.
В некоторых случаях задача имеет аналитическое решение.

XGBoost: приближённое аналитическое решение для w_j

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

$$Q(w) \approx \tilde{Q}(w) = \sum_{i=1}^{\ell}\Bigl(g_i b_i + \frac{1}{2}h_i b_i^2\Bigr) + \gamma|K| + \frac{\lambda}{2}\sum_{k\in K} w_k^2 \;\to\; \min_w,$$

где $b_i^p = \sum_k w_k^p B_k(x_i)$, $g_i = \mathscr{L}'\bigl(a(x_i), y_i\bigr)$, $h_i = \mathscr{L}''\bigl(a(x_i), y_i\bigr)$.

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:

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

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: $g_i = \mathscr{L}'\bigl(a_{t-1}(x_i), y_i\bigr)$ 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?

L.Prokhorenkova et al. CatBoost: unbiased boosting with categorical features. 2018.

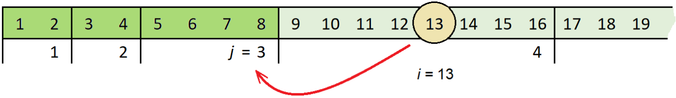
Упорядоченный бустинг (ordered boosting)

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^ℓ
X^{rj} — подвыборка первых 2^j объектов из $\sigma_r(X^\ell)$
$a_t^{rj}(x)$ — ансамбль-полуфабрикат, обученный по X^{rj}
$g_{ti}^r = -\mathscr{L}'\bigl(a_{t-1}^{rj}(x_i), y_i\bigr)$ — антиградиент в точке (x_i, y_i) для ансамбля a_{t-1}^{rj}, который по ней не обучался, $j = \lfloor \log_2(i-1) \rfloor$



L.Prokhorenkova et al. CatBoost: unbiased boosting with categorical features. 2018.

Модификация градиентного бустинга

сгенерировать случайные перестановки $\sigma_0, \sigma_1, \dots, \sigma_s$;
для всех $t = 1, \dots, T$:
- выбрать перестановку σ_r случайно из $\sigma_1, \dots, \sigma_s$;
- $g^r_{ti} := -\mathscr{L}'\bigl(a^{rj}_{t-1}(x_i), y_i\bigr)$ — несмещённый антиградиент;
- $b_t := \arg\min\limits_b \sum\limits_{i=1}^{\ell} \bigl(b(x_i) - g^r_{ti}\bigr)^2$;
- **для всех** деревьев b^{rj}_t, $r = 1, \dots, s$, $2^j \leqslant \ell$:
  - скопировать общую для них структуру дерева из b_t;
  - вычислить в листьях b^{rj}_t средние по $\bigl\{g^r_{ti} : x_i \in X^{rj}\bigr\}$;
- вычислить в листьях b_t средние по $\bigl\{g^0_{ti} : x_i \in X^{0j}\bigr\}$;
- GB: вычислить α_t и обновить $a_{t,i} := a_{t-1,i} + \alpha_t b_t(x_i)$;

L.Prokhorenkova et al. CatBoost: unbiased boosting with categorical features. 2018.

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

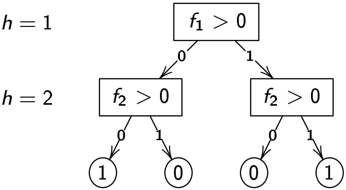
Небрежные решающие деревья (Oblivious Decision Tree, ODT)

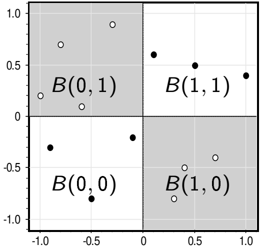
Решающая таблица: бинарное дерево глубины H;
для всех узлов уровня h условие ветвления $f_h(x)$ *одинаково*;
на уровне h число вершин 2^{h-1}; число листьев 2^H.

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

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

Пример: задача XOR, $H = 2$.





R.Kohavi, C.-H.Li. Oblivious decision trees, graphs, and top-down pruning. 1995.

Жадный алгоритм обучения ODT

Вход: выборка X^ℓ; множество признаков F; глубина дерева H;
Выход: признаки f_h, $h = 1, \dots, H$; таблица $B\colon \{0,1\}^H \to Y$;
для всех $h = 1, \dots, H$
признак с максимальным выигрышем определённости:
$$f_h := \arg\max_{f \in \mathrm{bin}\{F\}} \mathrm{Gain}\,(f_1, \dots, f_{h-1}, f);$$

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

$U_{h\beta} = \{x_i \in X^\ell\colon f_s(x_i) = \beta_s,\ s = 1..h\}$ — 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^ℓ:

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

Напоминание. Конъюнктивные логические закономерности

Семейство интерпретируемых логических правил:

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

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

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

$$\begin{cases} p_y(b) = \#\bigl\{x_i\colon b(x_i){=}1 \text{ и } y_i{=}y\bigr\} \to \max\limits_b \\ n_y(b) = \#\bigl\{x_i\colon b(x_i){=}1 \text{ и } y_i{\neq}y\bigr\} \to \min\limits_b \end{cases}$$

Критерий обучения базовых алгоритмов (из AdaBoost):

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

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

Алгоритмы Вычисления Оценок (АВО)

Бинарная функция сходства по набору признаков J:

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

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

Преимущества:

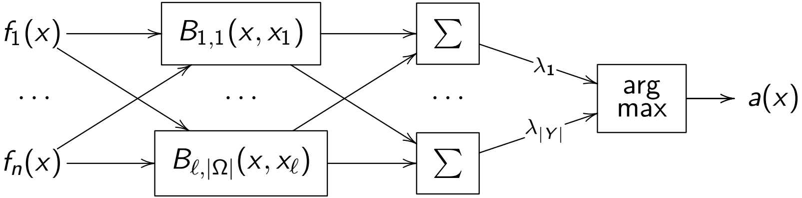
- подходит для задач с малой обучающей выборкой
- объединяет принципы голосования, сходства и поиска логических правил в информативных подпространствах

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):

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



$\approx$ метрический метод с ядрами $B_{i\omega}(x, x_i) = K\bigl(\frac{1}{h_{i\omega}} \rho_{i\omega}(x, x_i)\bigr)$

$\approx$ линейный классификатор SVM с радиальным ядром

$\approx$ байесовский классификатор с плотностями-смесями $p(x|y)$

$\approx$ отбор эталонов: $w_{i\omega} = 0$ для не-эталонов x_i

$\approx$ отбор признаков в сферических логических закономерностях

Принципы информативности, непротиворечивости, тупиковости

- *информативность* предиката $b(x)$ класса $y \in Y$:
$$\begin{cases} p_y(b) = \#\{x_i \colon b(x_i)=1 \text{ и } y_i=y\} \to \max \\ n_y(b) = \#\{x_i \colon b(x_i)=1 \text{ и } y_i\neq y\} \to \min \end{cases}$$
- *информативность* функции сходства $B(x, x')$:
$$\begin{cases} p(B) = \#\{(x_i, x_j) \colon B(x_i, x_j)=1 \text{ и } y_i=y_j\} \to \max \\ n(B) = \#\{(x_i, x_j) \colon B(x_i, x_j)=1 \text{ и } y_i\neq y_j\} \to \min \end{cases}$$
- *непротиворечивость*: $n_y(b) = 0$, $n(B) = 0$
  — *тест* J: $B_J(x_i, x_j) = 0$, $\forall i, j \colon y_i\neq y_j$
  — *представительный набор* (J, i): $B_J(x_i, x_j) = 0$, $\forall j \colon y_i\neq y_j$
- *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

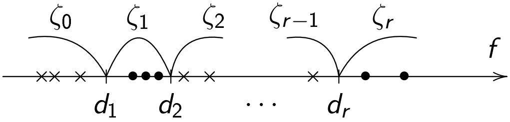
Цель — сократить перебор предикатов вида $\bigl[f(x) \leqslant \alpha\bigr]$.

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

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

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:

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

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^ℓ; 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 $y_{i-1} \neq y_i$ **то**
  добавить порог $\frac{1}{2}(f(x_{i-1}) + f(x_i))$ в конец D
повторять
 для всех $d_i \in D$, $i = 1, \dots, |D| - 1$
  $\delta I_i := I(\zeta_{i-1} \vee \zeta_i \vee \zeta_{i+1}) - \max\bigl\{I(\zeta_{i-1}), I(\zeta_i), I(\zeta_{i+1})\bigr\}$;
 $i := \arg\max\limits_s \delta I_s$;
 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

- 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