Курс «Введение в машинное обучение»

Композиционные методы машинного обучения

Воронцов Константин Вячеславович
k.v.vorontsov@phystech.edu
http://www.MachineLearning.ru/wiki?title=User:Vokov

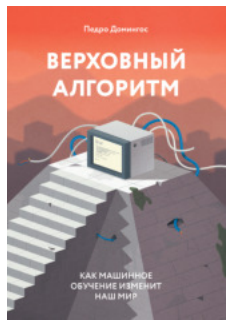
Этот курс доступен на странице вики-ресурса
http://www.MachineLearning.ru/wiki
«Введение в машинное обучение (курс лекций, К.В.Воронцов)»

МФТИ.ФПМИ.ИС.ИАД • 24 апреля 2025

Основные научные школы машинного обучения

1. *символизм* – поиск логических закономерностей
   - Decision Tree, Rule Induction
2. *коннекционизм* – обучаемые нейронные сети
   - BackPropagation, Deep Belief Nets, Deep Learning
     CNN, ResNet, LSTM, GRU, Attention, Transformer
3. *эволюционизм* – саморазвитие сложных моделей
   - Genetic Algorithms, Genetic Programming, Symbolic Regression
4. *байесионизм* и вероятностно-статистические методы
   - MLE, EM, GLM, LR, OBC, Naive Bayes, QD, LDF
     Bayesian Networks, Bayesian Learning, Graphical Models
5. *аналогизм* – «близким объектам близкие ответы»
   - kNN, RBF, SVM, KDE, Kernel Smoothing

⊕ *композиционизм* – кооперация моделей
   - Weighted Voting, Boosting, Bagging, Stacking,
     Random Forest, Яндекс.CatBoost



Педро Домингос. Верховный алгоритм. 2016. 336 с.

Содержание

Научная школа Ю. И. Журавлёва

Объединение принципов отбора признаков, информативности, голосования и сходства

- *алгоритмы вычисления оценок* (ABO, 1971)

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

$B_{i\omega}$ — бинарные функции сходства по информативным наборам признаков ω:

$$B_{i\omega}(x, x_i) = \prod_{j \in \omega} \big[\, |f_j(x) - f_j(x_i)| < \varepsilon_j \big]$$

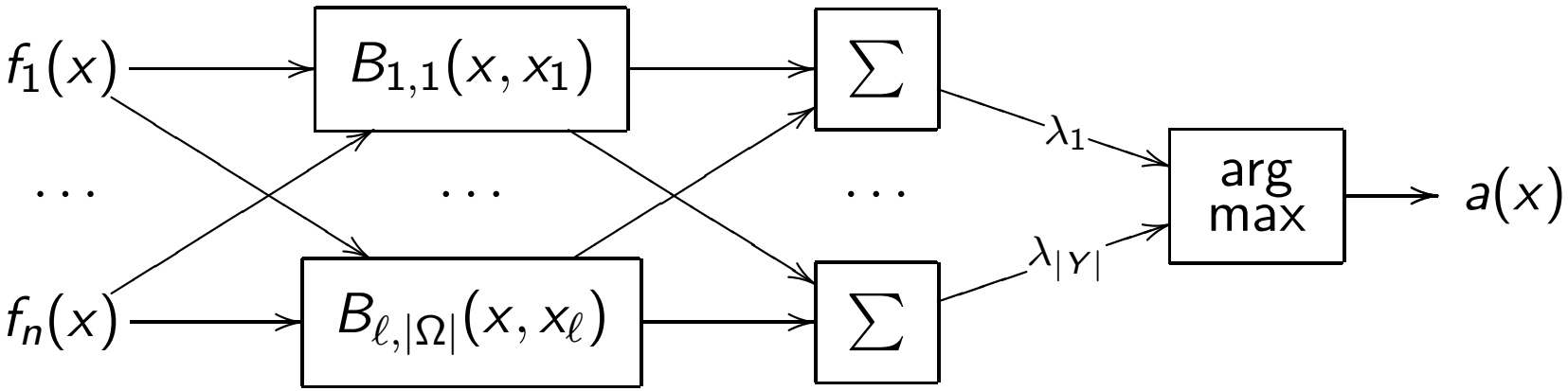
Юрий Иванович Журавлёв (1935–2022)

- принципы информативности, непротиворечивости, простоты
  $\rightarrow$ *тупиковые тесты* / *тупиковые представительные наборы*
- принцип голосования $\rightarrow$ *алгебраический подход* к построению корректных композиций алгоритмов (1977)

АВО объединяет эвристические принципы всех основных школ

Трёхслойная нейросеть RBF (Radial Basis Functions):

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



Метрический метод с ядрами $B_{i\omega}(x, x_i) = K\left(\frac{1}{h_{i\omega}} \rho_{i\omega}(x, x_i)\right)$

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 $R(x)$ класса $y \in Y$:
$$\begin{cases} p_y(R) = \#\{x_i\colon R(x_i)=1 \text{ и } y_i=y\} \to \max \\ n_y(R) = \#\{x_i\colon R(x_i)=1 \text{ и } y_i\neq y\} \to \min \end{cases}$$
- *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*: $n_y(B) = 0$
— *тест* ω: $B_\omega(x_i, x_j) = 0$, $\forall i, j$: $y_i \neq y_j$
— *представительный набор* (ω, i): $B_\omega(x_i, x_j) = 0$, $\forall j$: $y_i \neq y_j$
- *тупиковость*: никакое подмножество признаков $\omega' \subset \omega$ не является тестом (или представительным набором)

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.

Ансамблирование предсказательных моделей

$X^\ell = (x_i, y_i)_{i=1}^\ell \subset X \times Y$ — обучающая выборка, $y_i = y(x_i)$

$a_t \colon X \to Y, \quad t = 1, \ldots, T$ — обучаемые *базовые алгоритмы*

Идея ансамблирования (Ю.И.Журавлёв): как из множества по отдельности плохих алгоритмов a_t построить один хороший?

Декомпозиция базовых алгоритмов $a_t(x) = C(b_t(x))$

$a_t \colon X \xrightarrow{b_t} R \xrightarrow{C} Y$, где R — удобное *пространство оценок*,

b_t — базовые *алгоритмические операторы*,

C — *решающее правило* простого вида.

Ансамбль (композиция) базовых алгоритмов $a_1, \ldots, a_T$,

$F \colon R^T \to R$ — *корректирующая* (агрегирующая) *операция*

$$a(x) = C\big(F(b_1(x), \ldots, b_T(x))\big)$$

Ю.И.Журавлёв. Об алгебраическом подходе к решению задач распознавания или классификации. Проблемы кибернетики, 1978.

Агрегирующие (корректирующие) функции

Общие требования к агрегирующей функции:

- $F(b_1, \ldots, b_T, x) \in \left[\min_t b_t, \max_t b_t\right]$ — среднее по Коши $\forall x$
- $F(b_1, \ldots, b_T, x)$ монотонно не убывает по всем b_t

Примеры агрегирующих функций:

- простое голосование (simple voting):
$$F(b_1, \ldots, b_T) = \frac{1}{T} \sum_{t=1}^{T} b_t$$
- взвешенное голосование (weighted voting):
$$F(b_1, \ldots, b_T) = \sum_{t=1}^{T} \alpha_t b_t, \quad \sum_{t=1}^{T} \alpha_t = 1, \quad \alpha_t \geqslant 0$$
- смесь алгоритмов (mixture of experts)
с функциями компетентности (gating function) $g_t : X \to \mathbb{R}$
$$F(b_1, \ldots, b_T, x) = \sum_{t=1}^{T} g_t(x) b_t(x)$$

Обучение предсказательных моделей и их ансамблей

$\mathscr{L}(b, x_i)$ — функция потерь модели $b(x_i, w)$ при ответе y_i
Минимизация эмпирического риска для базовых алгоритмов:

$$\sum_{i=1}^{\ell} \mathscr{L}\big(b_t(x_i, w), y_i\big) \;\to\; \min_w$$

Минимизация эмпирического риска для добавления базового алгоритма b_T в ансамбль при фиксации предыдущих:

$$\sum_{i=1}^{\ell} \mathscr{L}\left(\sum_{t=1}^{T-1} \alpha_t b_t(x_i, w_t) + \alpha_T b_T(x_i, w_T), y_i\right) \;\to\; \min_{\alpha_T, w_T}$$

Ю.И.Журавлёв. Корректные алгебры над множествами некорректных (эвристических) алгоритмов (I, II, III). Кибернетика, Киев, 1977–1978.
M.Kearns, L.G.Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. 1989.
Y.Freund, R.E.Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. 1995.
К.В.Рудаков, К.В.Воронцов. О методах оптимизации и монотонной коррекции в алгебраическом подходе к проблеме распознавания. Доклады РАН, 1999.

История обучаемых композиций обучаемых моделей

- Простое и взвешенное голосование
  Ablow C. M., *Kaylor D. J.* Inconsistent homogeneous linear inequalities. 1965
  Мазуров В. Д. Комитеты системы неравенств и задача распознавания. 1971.
  Журавлёв Ю. И. Корректные алгебры над множествами некорректных (эвристических) алгоритмов. 1977.
  Freund Y., *Schapire R. E.* A decision-theoretic generalization of on-line learning and an application to boosting. 1995.
  Friedman G. Greedy function approximation: A gradient boosting machine. 1999.
- Случайный лес
  Breiman L. Random Forests. 2001.
- Восстановление смесей распределений, EM-алгоритм
  Шлезингер М. И. О самопроизвольном различении образов. 1965.
  Dempster A. P., *Laird N. M.*, *Rubin D. B.* Maximum likelihood from incomplete data via the EM-algorithm. 1977.
- Смеси классификаторов с областями компетентности
  Растригин Л. А., *Эренштейн Р. Х.* Коллективные правила распознавания. 1981.
  Jacobs R. A., *Jordan M. I.*, *Nowlan S. J.*, *Hinton G. E.* Adaptive mixtures of local experts. 1991.

Градиентный бустинг и *случайный лес* — универсальные и наиболее успешные методы классификации.

MatrixNet и *CatBoost* — эффективные реализации от Яндекса.

Проблема разнообразия (diversity) базовых алгоритмов

Измерение с.в. ξ по независимым наблюдениям $\{\xi_t\}$:

- $\mathsf{E}\frac{1}{T}(\xi_1 + \cdots + \xi_T) = \mathsf{E}\xi$ — матожидание среднего
- $\mathsf{D}\frac{1}{T}(\xi_1 + \cdots + \xi_T) = \frac{1}{T}\mathsf{D}\xi$ — дисперсия $\to 0$ при $T \to \infty$

Но базовые алгоритмы не являются независимыми с.в.:

- решают одну и ту же задачу
- настраиваются на один целевой вектор (y_i)
- обычно выбираются из одной и той же модели

Способы повышения *разнообразия* базовых алгоритмов:

- обучение по различным (случайным) подвыборкам
- обучение по различным (случайным) наборам признаков
- обучение из разных параметрических моделей
- обучение с использованием рандомизации
- (иногда даже) обучение по зашумлённым данным

Методы стохастического ансамблирования

Способы повышения разнообразия с помощью рандомизации:

- bagging (bootstrap aggregating) — подвыборки обучающей выборки «с возвращением», в каждую выборку попадает $1 - \left(1 - \frac{1}{\ell}\right)^{\ell} \to 1 - \frac{1}{e} \approx 63.2\%$ объектов, при $\ell \to \infty$
- pasting — случайные обучающие подвыборки
- random subspaces — случайные подмножества признаков
- random patches — случ. подмн-ва и объектов, и признаков
- cross-validated committees — выборка разбивается на k блоков (k-fold) и делается k обучений без одного блока

Пусть $\mu\colon (G, U) \mapsto b$ — *метод обучения* по подвыборке $U \subseteq X^{\ell}$, использующий только признаки из $G \subseteq F^n = \{f_1, \ldots, f_n\}$

Tin Kam Ho. The random subspace method for constructing decision forests. 1998.
Leo Breiman. Bagging predictors // Machine Learning. 1996.

Методы стохастического ансамблирования в одном псевдо-коде

Вход: обучающая выборка X^ℓ; параметры: T,
ℓ' — объём обучающих подвыборок,
n' — размерность признаковых подпространств,
ε_1 — порог качества базовых алгоритмов на обучении,
ε_2 — порог качества базовых алгоритмов на контроле;
Выход: базовые алгоритмы b_t, $t = 1, \ldots, T$;
для всех $t = 1, \ldots, T$:
$U_t :=$ случайная подвыборка объёма ℓ' из X^ℓ;
$G_t :=$ случайное подмножество мощности n' из F^n;
$b_t := \mu(G_t, U_t)$;
если $Q(b_t, U_t) > \varepsilon_1$ **то** не включать b_t в ансамбль;
если $Q(b_t, X^\ell \setminus U_t) > \varepsilon_2$ **то** не включать b_t в ансамбль;

Ансамбль — простое голосование: $b(x) = \frac{1}{T} \sum_{t=1}^{T} b_t(x)$

Несмещённая оценка ошибок

Out-of-bag — несмещённая оценка ансамбля на объекте:

$$\text{OOB}(x_i) = \frac{1}{|T_i|} \sum_{t \in T_i} b_t(x_i), \qquad T_i = \{t : x_i \notin U_t\}$$

Несмещённая оценка ошибки ансамбля на обучающей выборке:

$$\text{OOB}(X^\ell) = \sum_{i=1}^{\ell} \mathscr{L}\big(\text{OOB}(x_i), y_i\big),$$

где $\mathscr{L}\big(b(x_i), y_i\big)$ — значение функции потерь на объекте x_i.

Оценивание важности признаков f_j, $j = 1, \ldots, n$:

$$\text{importance}_j = \frac{\text{OOB}^j(X^\ell) - \text{OOB}(X^\ell)}{\text{OOB}(X^\ell)} \cdot 100\%,$$

где при вычислении $b_t(x_i)$ для OOB^j значения признака f_j случайным образом перемешиваются на всех объектах $x_i \notin U_t$.

Случайный лес (random forest)

Грубое обучение деревьев для случайного леса:

- бэггинг над решающими деревьями, без pruning
- признак в каждой вершине дерева выбирается из случайного подмножества k из n признаков. По умолчанию $k = \lfloor n/3 \rfloor$ для регрессии, $k = \lfloor \sqrt{n} \rfloor$ для классификации

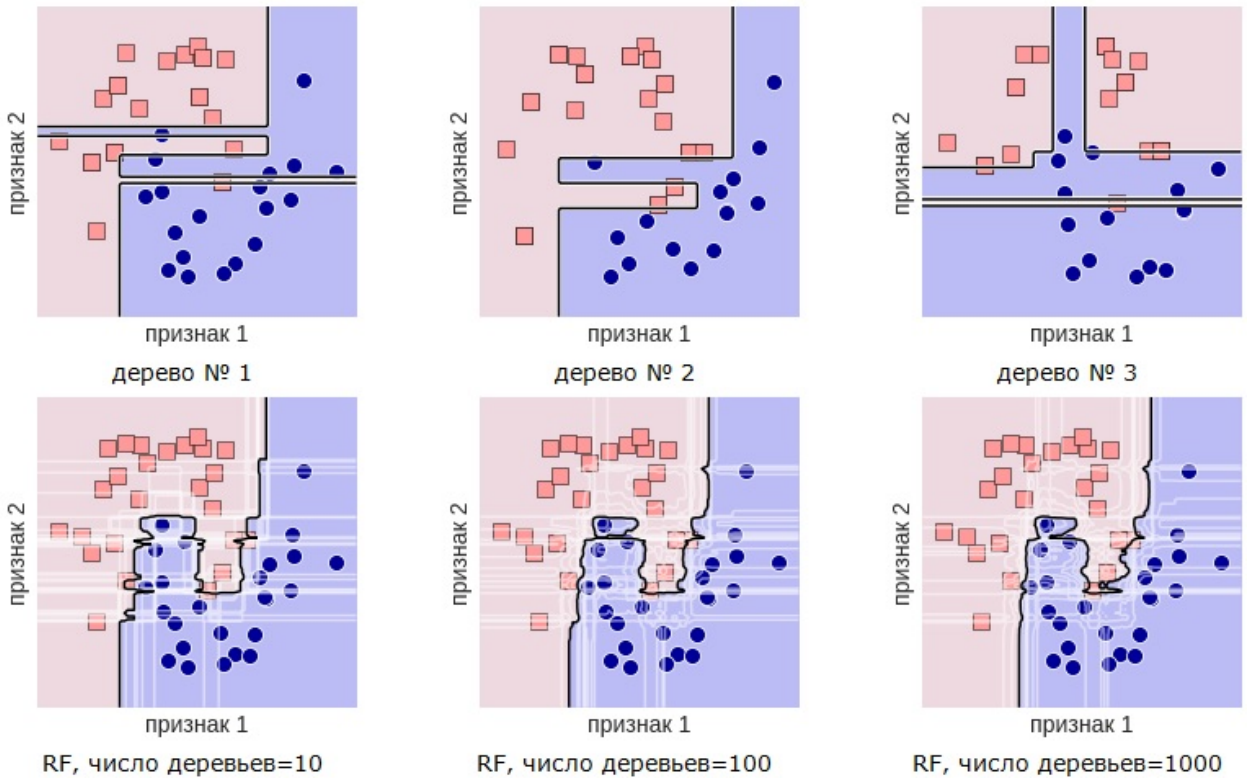
Параметры, которые можно настраивать (в частности, по OOB):

- число T деревьев
- число k случайно выбираемых признаков
- максимальная глубина деревьев
- минимальное число объектов в листьях
- минимальное число объектов для расщепления подвыборки
- критерий расщепления внутренних вершин дерева

Breiman L. Random Forests. Machine Learning, 2001.

Постепенное сглаживание разделяющей поверхности

Пример разделения выборки с помощью отдельных деревьев (показаны соответствующие бутстреп-подвыборки) и случайного леса с числом деревьев 10, 100, 1000:



https://dyakonov.org/2019/04/19/ансамбли-в-машинном-обучении

Преимущества и ограничения стохастического ансамблирования

Преимущества:

- метод-обёртка (envelop) над базовым методом обучения
- подходит для классификации, регрессии и других задач
- простая реализация и простое распараллеливание
- возможность получения несмещённых оценок OOB
- возможность оценивания важности признаков
- RF — один из лучших универсальных методов в ML

Ограничения:

- требуется ооооооочень много базовых алгоритмов
- трудно агрегировать устойчивые базовые методы обучения

---

https://dyakonov.org/2016/11/14/случайный-лес-random-forest

Градиентный бустинг с произвольной функцией потерь

Линейный ансамбль базовых алгоритмов b_t из семейства $\mathscr{B}$:

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

Эвристика: обучаем α_T, b_T при фиксированных предыдущих.

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

$$Q(\alpha, b; X^\ell) = \sum_{i=1}^{\ell} \mathscr{L}\Bigg(\underbrace{\underbrace{\sum_{t=1}^{T-1} \alpha_t b_t(x_i)}_{a_{T-1,i}} + \alpha b(x_i)}_{a_{T,i}}, y_i \Bigg) \to \min_{\alpha, b}.$$

$\big(a_{T-1,i}\big)_{i=1}^{\ell}$ — вектор текущего приближения

$\big(a_{T,i}\big)_{i=1}^{\ell}$ — вектор следующего приближения

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

Параметрическая аппроксимация градиентного шага

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

$$a_{0,i} := \text{начальное приближение};$$
$$a_{T,i} := a_{T-1,i} - \alpha g_i, \quad i = 1, \ldots, \ell;$$

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

Это очень похоже на добавление одного базового алгоритма:

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

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

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

Алгоритм градиентного бустинга (Gradient Boosting)

Вход: обучающая выборка X^ℓ; параметр T;
Выход: базовые алгоритмы и их веса $\alpha_t b_t$, $t = 1, \ldots, T$;
инициализация: $a_{0,i} := 0$, $i = 1, \ldots, \ell$;
для всех $t = 1, \ldots, T$

базовый алгоритм, приближающий антиградиент:

$$b_t := \arg\min_{b \in \mathscr{B}} \sum_{i=1}^{\ell} \big(b(x_i) + \mathscr{L}'(a_{t-1,i}, y_i)\big)^2;$$

задача одномерной минимизации:

$$\alpha_t := \arg\min_{\alpha > 0} \sum_{i=1}^{\ell} \mathscr{L}\big(a_{t-1,i} + \alpha b_t(x_i), y_i\big);$$

обновление вектора значений на объектах выборки:

$a_{t,i} := a_{t-1,i} + \alpha_t b_t(x_i)$; $i = 1, \ldots, \ell$;

Каждый следующий базовый алгоритм обучается так, чтобы по возможности исправить ошибки предыдущих алгоритмов.

Пример. Классификация синтетической выборки

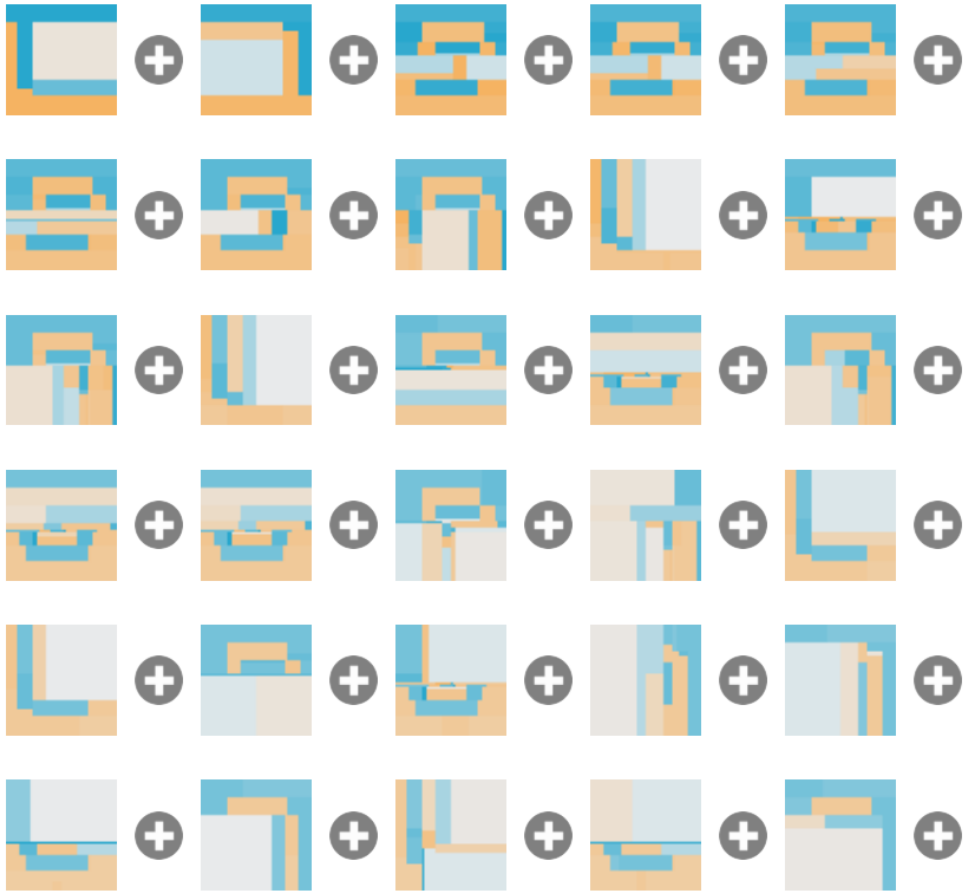
100 деревьев глубины 5

Prediction:

↑
predictions of GB (all 100 trees)

train loss: 0.022 test loss: 0.218

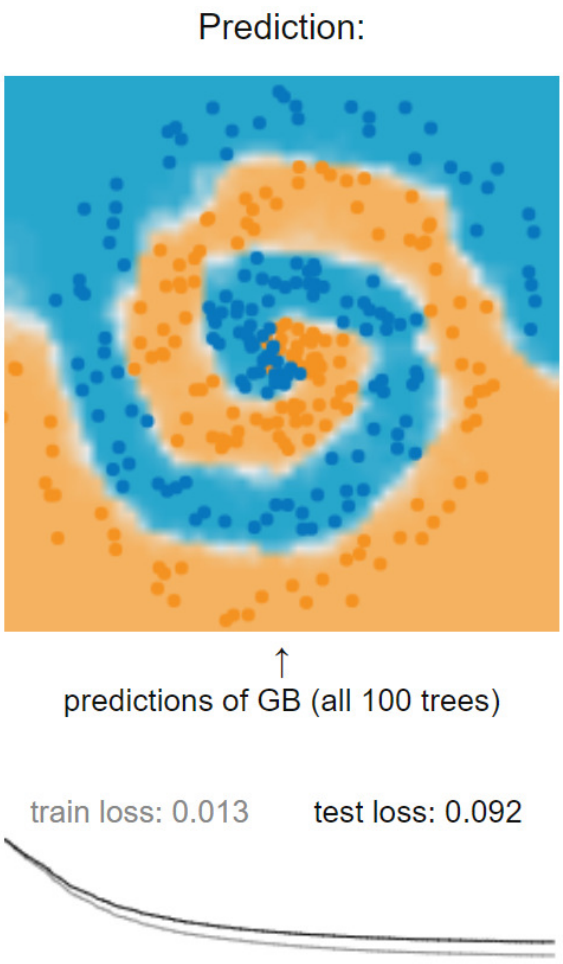
Decision functions of first 30 trees

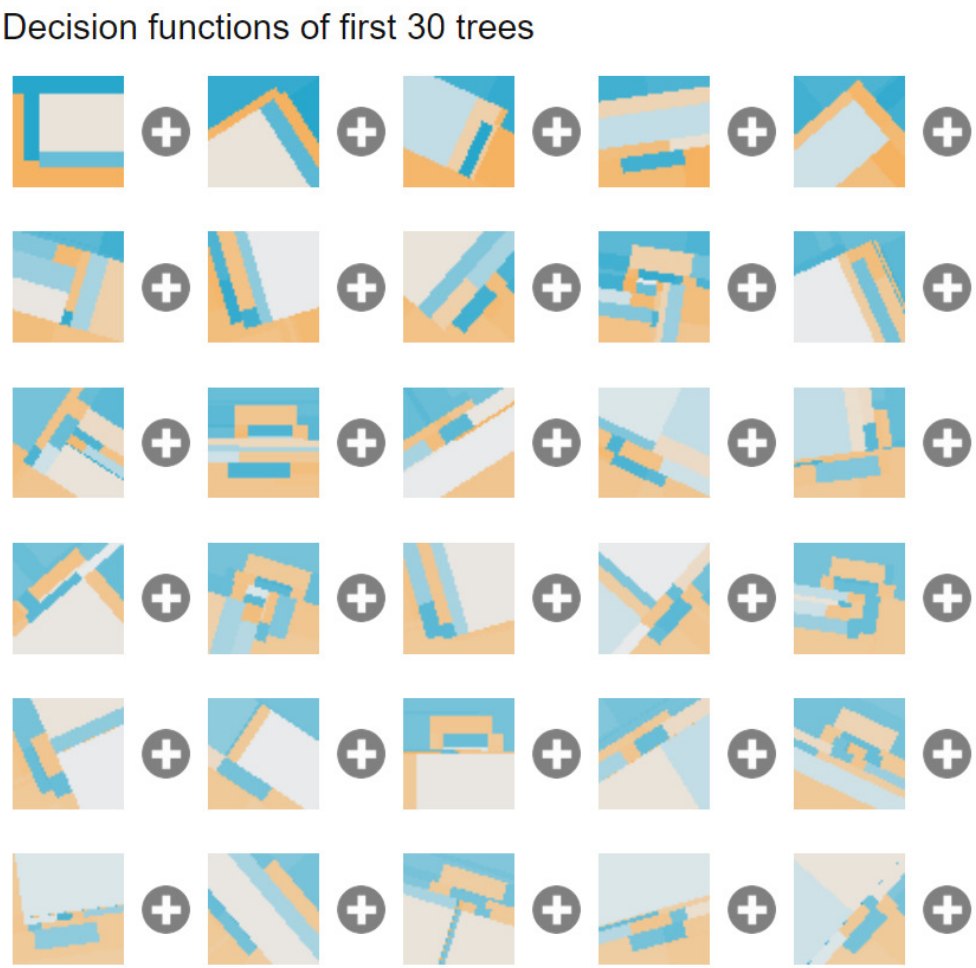


http://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html

Пример. Классификация синтетической выборки

100 деревьев глубины 5, с подбором вращения каждого дерева

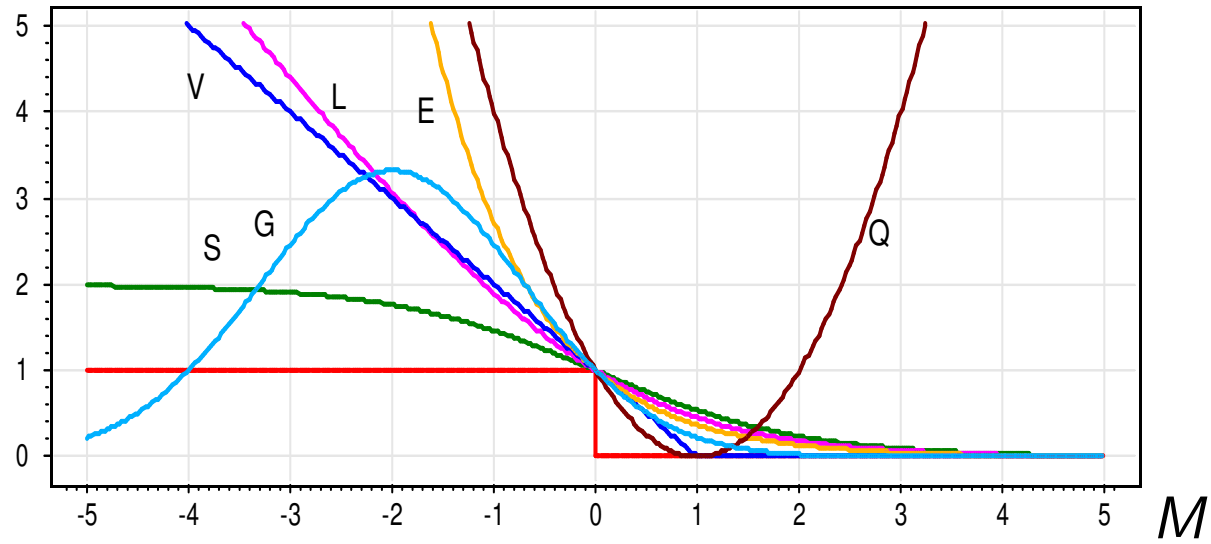




http://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html

Варианты бустинга для двухклассовой классификации

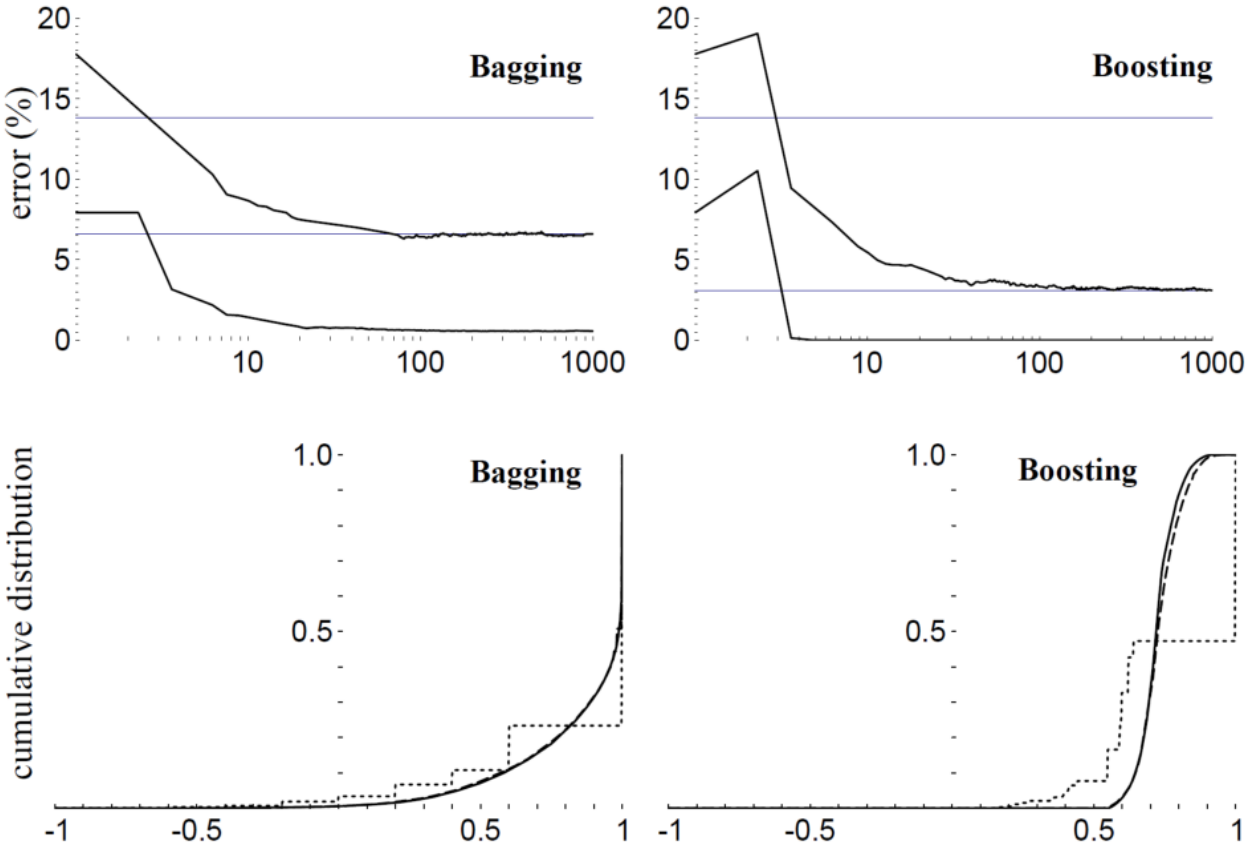
Гладкие аппроксимации пороговой функции потерь $[M < 0]$:



$E(M) = e^{-M}$ — экспоненциальная (AdaBoost);
$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LogitBoost);
$Q(M) = (1 - M)^2$ — квадратичная (GentleBoost);
$G(M) = \exp\big(-cM(M + s)\big)$ — гауссовская (BrownBoost);
$S(M) = 2(1 + e^{M})^{-1}$ — сигмоидная;
$V(M) = (1 - M)_+$ — кусочно-линейная (из SVM);

Удивительно: линейные ансамбли почти не переобучаются!

Ошибки на обучении и тесте. Снизу распределение отступов.



R.E.Schapire, Y.Freund, Wee Sun Lee, P.Bartlett. Boosting the margin: a new explanation for the effectiveness of voting methods. Annals of Statistics, 1998.

Обоснование линейных ансамблей (бинарная классификация)

Усиленная *частота ошибок* классификатора $\operatorname{sign} b(x)$, $b \in \mathscr{B}$:

$$\nu_\theta(b, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \big[b(x_i) y_i \leqslant \theta \big], \quad \theta > 0.$$

Обычная частота ошибок $\nu_0(b, X^\ell) \leqslant \nu_\theta(b, X^\ell)$ при $\theta > 0$.

Теорема (Freund, Schapire, Lee, Bartlett, 1998)

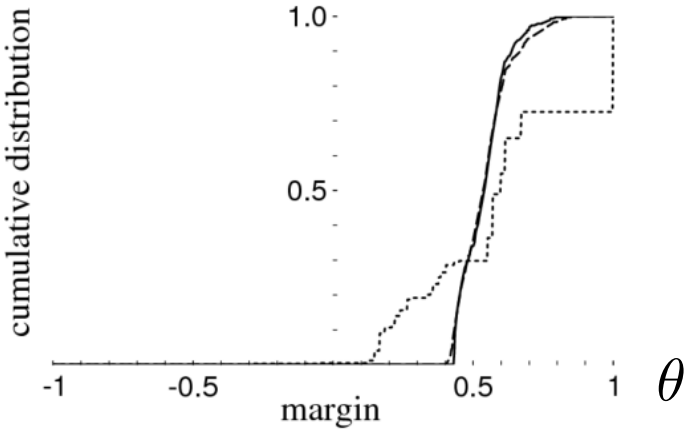
Если $|\mathscr{B}| < \infty$, *то* $\forall \theta > 0$, $\forall \eta \in (0,1)$ *с вероятностью* $1 - \eta$

$$\mathsf{P}[ya(x) < 0] \leqslant \nu_\theta(a, X^\ell) + C \sqrt{\frac{\ln |\mathscr{B}| \ln \ell}{\ell \theta^2} + \frac{1}{\ell} \ln \frac{1}{\eta}}$$

Основной вывод: оценка зависит от $|\mathscr{B}|$, но не от T. Голосование не увеличивает сложность семейства базовых алгоритмов, а лишь усредняет их ответы.

Обоснование бустинга: что же всё-таки происходит?

Распределение отступов:
доля объектов, имеющих отступ меньше заданного θ
после 5, 100, 1000 итераций
(Задача UCI:vehicle)



- С ростом T распределение отступов сдвигается вправо, то есть бустинг «раздвигает» классы в пространстве векторов растущей размерности $\big(b_1(x), \ldots, b_T(x)\big)$
- Значит, в оценке можно уменьшать второй член, увеличивая θ при неизменной $\nu_\theta(a, X^\ell) = \nu_0(a, X^\ell)$.
- Можно уменьшить второй член, если уменьшить $|\mathscr{B}|$, то есть взять простое семейство базовых алгоритмов.

R.E.Schapire, Y.Freund, Wee Sun Lee, P.Bartlett. Boosting the margin: a new explanation for the effectiveness of voting methods. Annals of Statistics, 1998.

Анализ смещения–разброса (bias–variance)

Задача регрессии: $Y = \mathbb{R}$
Квадратичная функция потерь: $\mathscr{L}(a, y) = \big(a(x) - y\big)^2$
Вероятностная постановка: $X^\ell = (x_i, y_i)_{i=1}^\ell \sim p(x, y)$
Метод обучения: $\mu\colon 2^X \to A$, т.е. выборка $\mapsto$ алгоритм

Задача минимизации среднеквадратичного риска:

$$R(a) = \mathsf{E}_{x,y}(a(x) - y)^2 = \int_X \int_Y \big(a(x) - y\big)^2 p(x, y)\, dx\, dy \to \min_a$$

Идеальный минимизатор среднеквадратичного риска:

$$a^*(x) = \mathsf{E}(y|x) = \int_Y y\, p(y|x)\, dy$$

Основная мера качества метода обучения μ:

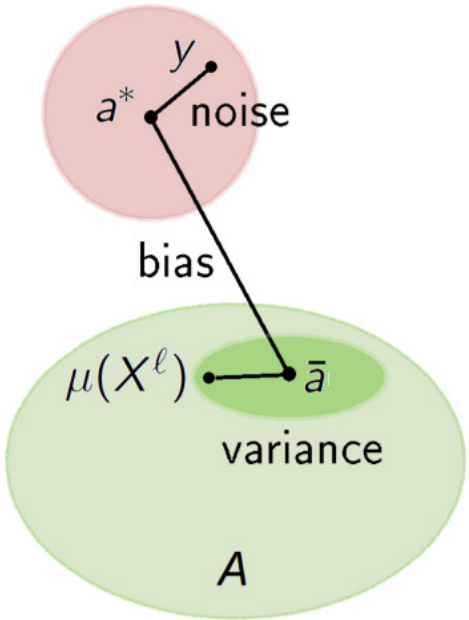
$$Q(\mu) = \mathsf{E}_{X^\ell} R\big(\mu(X^\ell)\big) = \mathsf{E}_{X^\ell} \mathsf{E}_{x,y} \big(\mu(X^\ell)(x) - y\big)^2$$

Разложение ошибки на шум, смещение и разброс

$a^*(x) = \mathsf{E}(y|x)$ — неизвестная идеальная зависимость y от x
$y(x) \sim p(y|x)$ — наблюдаемый ответ на объекте x
$a = \mu(X^\ell)$ — аппроксимация, выбранная по X^ℓ из семейства A
$\bar{a}(x) = \mathsf{E}_{X^\ell}\big(a(x)\big)$ — средний ответ обученного алгоритма

Теорема. При квадратичной функции потерь для любого μ

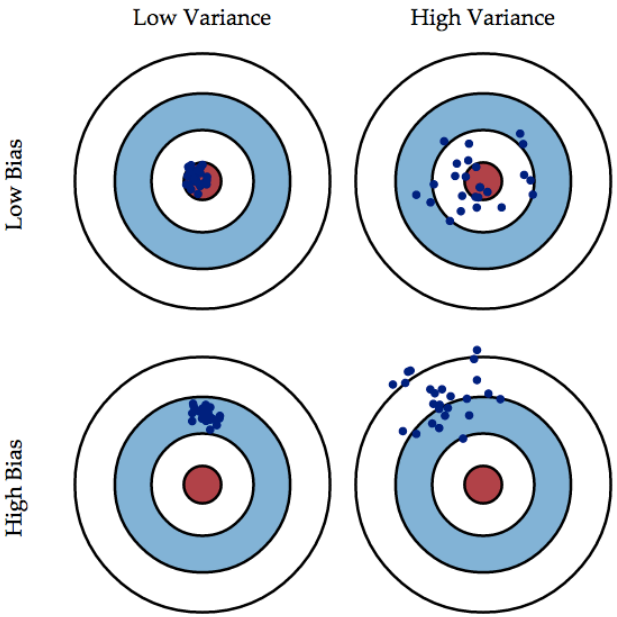
$$Q(\mu) = \underbrace{\mathsf{E}_{x,y}\big(a^*(x) - y\big)^2}_{\text{шум (noise)}} +$$
$$+ \underbrace{\mathsf{E}_{x,y}\big(\bar{a}(x) - a^*(x)\big)^2}_{\text{смещение (bias)}} +$$
$$+ \underbrace{\mathsf{E}_{x,y}\mathsf{E}_{X^\ell}\big(\mu(X^\ell)(x) - \bar{a}(x)\big)^2}_{\text{разброс (variance)}}$$

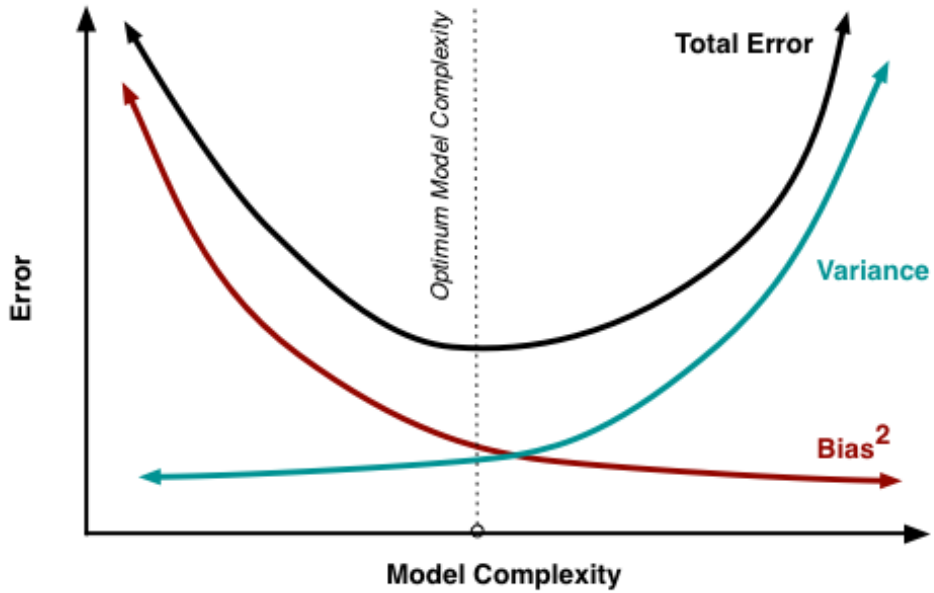


Разложение ошибки на шум, смещение и разброс

Качественное понимание: по мере роста сложности модели

- смещение (bias) уменьшается
- разброс (variance) увеличивается





Анализ смещения–разброса для простого голосования

Обучение базовых алгоритмов по случайным подвыборкам:
$b_t = \mu(X_t^k),\ X_t^k \sim X^\ell,\ t = 1, \ldots, T$

Ансамбль — простое голосование: $a_T(x) = \frac{1}{T}\sum_{t=1}^{T} b_t(x)$

Смещение ансамбля совпадает со смещением отдельного базового алгоритма:

$$\text{bias} = \mathsf{E}_{x,y}\big(a^*(x) - \mathsf{E}_{X^\ell} b_t(x)\big)^2$$

Разброс состоит из дисперсии и различности (ковариации):

$$\text{variance} = \tfrac{1}{T}\mathsf{E}_{x,y}\mathsf{E}_{X^\ell}\big(b_t(x) - \mathsf{E}_{X^\ell} b_t(x)\big)^2 +$$
$$+ \tfrac{T-1}{T}\mathsf{E}_{x,y}\mathsf{E}_{X^\ell}\big(b_t(x) - \mathsf{E}_{X^\ell} b_t(x)\big)\big(b_s(x) - \mathsf{E}_{X^\ell} b_s(x)\big)$$

Почему сложные ансамбли не переобучаются?

С позиций анализа отступов:

- ансамблирование не увеличивает сложность модели
- но с каждой итерацией увеличивает зазор между классами

С позиций анализа смещения–разброса:

- разнообразие базовых алгоритмов уменьшает разброс
- бэггинг уменьшает только разброс
- бустинг уменьшает и смещение, и разброс

Практическое сравнение: boosting / bagging / RSM

- бустинг лучше для классов с границами сложной формы
- бэггинг и RSM лучше для коротких обучающих выборок
- RSM лучше, когда много неинформативных признаков
- бэггинг параллельно обучает базовые алгоритмы b_t
- бустинг обучает каждый b_t параллельно по частям выборки

Недостатки бэггинга и бустинга

- задача минимизировать число T вообще не ставится
- композиция из сотен алгоритмов не интерпретируема
- не удаётся строить короткие композиции из «сильных» алгоритмов типа SVM (только длинные из «слабых»)

Несколько эмпирических наблюдений:

- веса алгоритмов не важны для оптимизации отступов
- веса объектов не важны для обеспечения различности

Предлагается:

- отказаться от аппроксимации пороговой функции потерь,
- оптимизировать распределение отступов композиции,
- обучать базовые алгоритмы последовательно (как бустинг),
- обучать их на подвыборках (как бэггинг), но не случайных,
- использовать простое голосование (комитет большинства)

Оптимизация распределения отступов на каждом шаге

Идея: явно управлять распределением отступов, максимизируя различность базовых алгоритмов и минимизируя их число.

Возьмём $b(x) = \frac{1}{T}\sum_{t=1}^{T} b_t(x)$, $a(x) = \text{sign}(b(x))$, $Y = \{\pm 1\}$.

Критерий качества ансамбля — число ошибок на обучении:

$$Q(a, X^\ell) = \sum_{i=1}^{\ell} \big[y_i a(x_i) < 0\big] = \sum_{i=1}^{\ell} \big[\underbrace{y_i b_1(x_i) + \cdots + y_i b_T(x_i)}_{M_{iT}} < 0\big],$$

$M_{it} = y_i b_1(x_i) + \cdots + y_i b_t(x_i)$ — *отступ* (margin) объекта x_i.

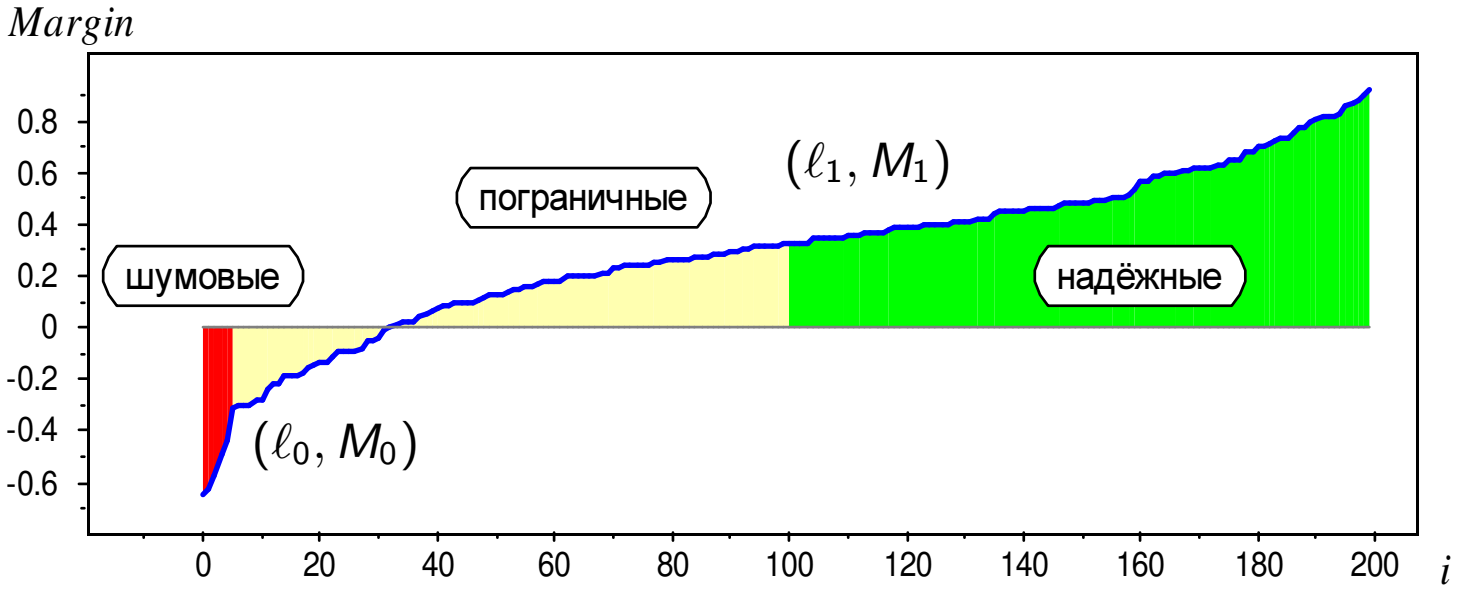
Эвристика: обучение b_t компенсирует ошибки ансамбля:

$$Q(b_t, U_t) = \sum_{x_i \in U_t} \big[y_i b_t(x_i) < 0\big] \to \min_{b_t},$$

$U_t = \big\{x_i : M_0 < M_{i,t-1} \leqslant M_1\big\}$, M_0, M_1 — параметры метода

Формирование выборки для обучения базового алгоритма

Упорядочим объекты по возрастанию отступов $M_{i,t-1}$:



Принцип выравнивания распределения отступов

два случая, когда b_t на объекте x_i обучать не надо:

$M_{i,t-1} < M_0, \quad i < \ell_0$ — объект x_i шумовой

$M_{i,t-1} > M_1, \quad i > \ell_1$ — объект x_i надёжный

Алгоритм ComBoost (Committee Boosting)

Вход: выборки X^ℓ, X^k; параметры $T, \ell_0, \ell_1, \ell_2, \Delta\ell$;
Выход: $b_1, \ldots, b_T$;

$b_1 := \arg\min_b Q(b, X^\ell)$; отступы $M_i = y_i b_1(x_i)$, $i = 1, \ldots, \ell$;
для всех $t = 2, \ldots, T$:
  упорядочить выборку X^ℓ по возрастанию отступов M_i;
  для всех $\ell' = \ell_1, \ldots, \ell_2$ с шагом $\Delta\ell$:
    $U_t = \{x_i \in X^\ell : \ell_0 \leqslant i \leqslant \ell'\}$;
    $b_{t\ell'} := \arg\min_b Q(b, U_t)$ — инкрементное обучение;
  выбрать наилучший $b_t \in \{b_{t\ell'}\}$ по критерию $Q(a, X^k)$;
  обновить отступы: $M_i := M_i + y_i b_t(x_i)$, $i = 1, \ldots, \ell$;
пока Q существенно улучшается.

Маценов А. А. Комитетный бустинг: минимизация числа базовых алгоритмов при простом голосовании. ММРО-13, 2007.

Результаты эксперимента на 4 задачах из репозитория UCI

По 50 случайным разбиениям «обучение : контроль» = 4 : 1

	ionoshere	pima	bupa	votes
SVM	12,9	24,2	42,0	4,6
ComBoost$_0$[SVM] (T)	12,6 (**4**)	23,1 (**2**)	34,2 (**5**)	4,0 (**2**)
ComBoost[SVM] (T)	**12,3** (5)	**22,5** (**2**)	30,9 (**5**)	**3,8** (3)
AdaBoost[SVM] (T)	15,0 (65)	22,7 (18)	**30,6** (15)	4,0 (8)
Parzen	6,3	25,1	41,6	6,9
ComBoost$_0$[Parzen]	6,1	25,0	38,1	6,8
ComBoost[Parzen]	**5,8**	**24,7**	30,6	**6,2**
AdaBoost[Parzen]	6,0	24,8	**30,5**	6,5

ComBoost$_0$ — без подбора длины подвыборки U_t в цикле $\ell' = \ell_1, \ldots, \ell_2$
Parzen — метод окна Парзена с подбором ширины окна по leave-one-out

Результат: ComBoost способен строить короткие ансамбли из сильных и устойчивых базовых алгоритмов

Маценов А. А. Комитетный бустинг: минимизация числа базовых алгоритмов при простом голосовании. ММРО-13, 2007.

Обобщение для задач с произвольным числом классов

$Y = \{1, \ldots, M\}$, ансамбль — простое голосование, причём каждый базовый алгоритм b_{yt} голосует только за свой класс y:

$$a(x) = \arg\max_{y \in Y} \Gamma_y(x); \qquad \Gamma_y(x) = \frac{1}{|T_y|} \sum_{t \in T_y} b_{yt}(x).$$

В алгоритме ComBoost три небольших изменения:

- обобщённое определение отступа M_i:

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

- придётся решать, для какого класса строить очередной b_{yt} (например, для того y, на котором доля ошибок больше)
- изменится пересчёт отступов в конце итерации

Allwein E. L., Schapire R. E., Singer Y. Reducing multiclass to binary: A unifying approach for margin classifiers. 2000

Смесь алгоритмов (Mixture of Experts)

$b_t : X \to \mathbb{R}$ — базовые алгоритмы, $t = 1, \ldots, T$

$g_t : X \to \mathbb{R}$ — функция компетентности (шлюз, gate) для $b_t(x)$

$$b(x) = \sum_{t=1}^{T} g_t(x) b_t(x)$$

Чем больше $g_t(x)$, тем выше доверие к ответу $b_t(x)$.

Условие нормировки: $\sum_{t=1}^{T} g_t(x) = 1$ для любого $x \in X$.

Нормировка «мягкого максимума» SoftMax: $\mathbb{R}^T \to \mathbb{R}^T$:

$$\tilde{g}_t(x) = \mathrm{SoftMax}_t\big(g_1(x), \ldots, g_T(x); \gamma\big) = \frac{e^{\gamma g_t(x)}}{e^{\gamma g_1(x)} + \ldots + e^{\gamma g_T(x)}}$$

При $\gamma \to \infty$ SoftMax выделяет максимальную из T величин.

Растригин Л. А., Эренштейн Р. Х. Коллективные правила распознавания. 1981.
Hien D. Nguyen, Faicel Chamroukhi. Practical and theoretical aspects of mixture-of-experts modeling: An overview. 2018

Вид функций компетентности

Функции компетентности определяются из практических соображений, в зависимости от особенностей задачи, например:

- по признаку $f(x)$:
$$g(x;\alpha,\beta) = \sigma(\alpha f(x) + \beta), \quad \alpha,\beta \in \mathbb{R};$$
- по неизвестному направлению $\alpha \in \mathbb{R}^n$:
$$g(x;\alpha,\beta) = \sigma(x^{\mathsf{T}}\alpha + \beta), \quad \alpha \in \mathbb{R}^n,\ \beta \in \mathbb{R};$$
- по расстоянию до неизвестной точки $\alpha \in \mathbb{R}^n$:
$$g(x;\alpha,\beta) = \exp(-\beta\|x-\alpha\|^2), \quad \alpha \in \mathbb{R}^n,\ \beta \in \mathbb{R};$$

где параметры α, β фиксируются или обучаются по выборке, $\sigma(z) = \frac{1}{1+e^{-z}}$ — сигмоидная функция.

Выпуклые функции потерь

Функция потерь $\mathscr{L}(b, y)$ называется *выпуклой* по b, если $\forall\, y \in Y,\ \forall\, b_1, b_2 \in R,\ \forall\, g_1, g_2 \geqslant 0$: $g_1 + g_2 = 1$, выполняется

$$\mathscr{L}(g_1 b_1 + g_2 b_2, y) \leqslant g_1 \mathscr{L}(b_1, y) + g_2 \mathscr{L}(b_2, y).$$

Интерпретация: потери растут не медленнее, чем величина отклонения от правильного ответа y.

Примеры выпуклых функций потерь:

$$\mathscr{L}(b, y) = \begin{cases} (b - y)^2 & \text{— квадратичная (МНК-регрессия);} \\ e^{-by} & \text{— экспоненциальная (AdaBoost);} \\ \log_2(1 + e^{-by}) & \text{— логарифмическая (LR);} \\ (1 - by)_+ & \text{— кусочно-линейная (SVM).} \end{cases}$$

Пример невыпуклой функции потерь: $\mathscr{L}(b, y) = [by < 0]$.

Основная идея применения выпуклых функций потерь

Пусть $\forall x \sum_{t=1}^{T} g_t(x) = 1$ и функция потерь $\mathscr{L}$ выпукла.

Тогда $Q(a)$ распадается на T независимых критериев Q_t:

$$Q(a) = \sum_{i=1}^{\ell} \mathscr{L}\left(\sum_{t=1}^{T} g_t(x_i) b_t(x_i), y_i\right) \leqslant \sum_{t=1}^{T} \underbrace{\sum_{i=1}^{\ell} g_t(x_i) \mathscr{L}\big(b_t(x_i), y_i\big)}_{Q_t(g_t, b_t)}$$

Итерационный процесс, два шага на каждой итерации:

начальное приближение функций компетентности g_t;

повторять

  обучить все $b_t := \arg\min_b Q_t(g_t, b)$ при фиксированных g_t;

  обучить все g_t при фиксированных b_t;

пока значения компетентностей $g_t(x_i)$ не стабилизируются;

Философия ансамблирования

Ансамблировать можно только нечто гомогенное.

1. **Декомпозиция** — разделение модели алгоритма a_t на алгоритмический оператор b_t и решающее правило C:
$$a_t = C \circ b_t$$
2. **Гомогенизация** — разнородные модели имеют общее пространство оценок R и общую структуру алгоритмического оператора b_t как отображения
$$b_t \colon X \to R$$
3. **Ансамблирование** — совместное обучение базовых алгоритмических операторов для решения общей задачи:
$$a = C \circ F(b_1, \ldots, b_T)$$

Ю.И.Журавлёв. Об алгебраическом подходе к решению задач распознавания или классификации. Проблемы кибернетики, 1978.

Философия многозадачного обучения

1. **Декомпозиция** — разделение моделей $y_t : X \to Y_t$ на векторизатор $z = f(x, \alpha)$ и предиктор $y_t = g_t(z, \beta)$:

$$y_t(x) = g_t(f(x, \alpha), \beta_t)$$

2. **Гомогенизация** — разнородные модели имеют общий векторизатор $z = f(x, \alpha)$ и общее векторное пространство представлений (эмбедингов) Z:

$$f : X \to Z$$

3. **Ансамблирование** — совместное обучение общего векторизатора для решения разнородных задач:

$$\sum_{t \in T} \sum_{i \in X^t} \mathscr{L}_{ti}\big(g_t(f(x_{ti}, \alpha), \beta_t)\big) \to \min_{\alpha, \{\beta_t\}}$$

Yu Zhang, Qiang Yang. A survey on multi-task learning. 2021

M.Crawshaw. Multi-task learning with deep neural networks: a survey. 2020

Философия фундаментальных моделей

1. **Декомпозиция** — разделение моделей $y_t : X_t \to Y_t$ на векторизатор $z = f(x_t, \alpha_t)$ и предиктор $y_t = g(z_t, \beta_t)$:

$$y_t(x) = g_t(f(x_t, \alpha_t), \beta_t)$$

2. **Гомогенизация** — разнородные модели в разнородных задачах имеют общее пространство эмбедингов Z:

$$f_t : X_t \to Z$$

3. **Ансамблирование** — совместное обучение эмбедингов в едином семантическом пространстве для решения разнородных задач:

$$\sum_{t \in T} \sum_{i \in X^t} \mathscr{L}_{ti}\big(g_t(f(x_{ti}, \alpha_t), \beta_t)\big) \;\to\; \min_{\{\alpha_t, \beta_t\}}$$

R.Bommasani et al. (Center for Research on Foundation Models, Stanford University) On the opportunities and risks of foundation models // CoRR, 20 August 2021.

Философия аддитивной регуляризации (ARTM)

1. **Декомпозиция** — разделение критерия обучения модели на основной (log-правдоподобие) и регуляризатор R:

$$\sum_i \ln p(x_i|\Phi,\Theta) + R(\Phi,\Theta) \;\to\; \max_{\Phi,\Theta}$$

2. **Гомогенизация** — разнородные модели имеют общую структуру векторизатора (матричное разложение) и общий основной критерий (log-правдоподобие):

$$f_\Phi : X \to \Theta, \qquad \sum_i \ln p(x_i|\Phi,\Theta) \;\to\; \max_{\Phi,\Theta}$$

3. **Ансамблирование** — совместное использование регуляризаторов R_k, взятых от разнородных моделей:

$$\sum_i \ln p(x_i|\Phi,\Theta) + \sum_k \lambda_k R_k(\Phi,\Theta) \;\to\; \max_{\Phi,\Theta}$$

Vorontsov K. V. Rethinking probabilistic topic modeling from the point of view of classical non-Bayesian regularization. 2023.

Резюме

- Ансамбли позволяют решать сложные задачи, которые плохо решаются отдельными базовыми алгоритмами
- Обычно ансамбль строится *алгоритмом-обёрткой* (envelop): базовые алгоритмы обучаются готовыми методами
- 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
- Базовые алгоритмы: компромисс качество/различность
- Чаще всего GB применяется к решающим деревьям
- Для смешивания нужна адекватная модель компетентности