Методы машинного обучения. Линейные ансамбли

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 • 11 ноября 2025

Содержание

1. **Простое голосование**
   - Ансамблирование моделей
   - Бэггинг и случайные подпространства
   - Случайные леса
2. **Взвешенное голосование**
   - Адаптивный бустинг AdaBoost
   - Основная теорема AdaBoost
   - Алгоритм AdaBoost
3. **Эксперименты с бустингом**
   - Вид разделяющей поверхности
   - Переобучение бустинга
   - Сравнение бэггинга и бустинга

Задача обучения ансамбля (композиции) моделей

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

$a_t(x, w) = C(b_t(x, w))$ — «слабые» обучаемые *базовые модели*

$b_t \colon X \to R$ — *алгоритмические операторы* с параметрами w

$C \colon R \to Y$ — *решающее правило* простого вида (без параметров)

R — удобное *пространство оценок*

Найти: ансамбль $a(x) = C\big(F(b_1(x, w_1), \ldots, b_T(x, w_T), x, \alpha)\big)$

$F \colon R^T \times X \to R$ — *корректирующая функция* с параметрами α

Критерий обучения «сильного» алгоритма как ансамбля из T по-отдельности «слабых» базовых алгоритмов:

$$Q(a, X^\ell) = \sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i), y_i\big) \;\to\; \min_{w_1, \ldots, w_T, \alpha}$$

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

M.Kearns, L.G.Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. 1989.

Пространства оценок R и решающие правила C

- **Пример 1:** классификация, Y — конечное множество, $R = Y$, $C(b) \equiv b$ — решающее правило не используется.
- **Пример 2:** бинарная классификация, $Y = \{-1, +1\}$,

$$a(x, w) = \text{sign}\big(b(x, w)\big),$$

$R = \mathbb{R}$, $b\colon X \to \mathbb{R}$ — *real-valued classifier*, $C(b) \equiv \text{sign}(b)$.

- **Пример 3:** классификация на M классов $Y = \{1, \ldots, M\}$,

$$a(x, w) = \arg\max_{y \in Y} b_y(x, w_y),$$

$R = \mathbb{R}^M$, $b\colon X \to \mathbb{R}^M$, $C(b_1, \ldots, b_M) \equiv \arg\max_{y \in Y} b_y$.

- **Пример 4:** регрессия, $Y = R = \mathbb{R}$, $C(b) \equiv b$ — решающее правило не используется.

Корректирующие (агрегирующие) функции

Общие требования к агрегирующей функции $F(b_1, \ldots, b_T, x, \alpha)$:

- $F \in \left[\min_t b_t, \max_t b_t\right]$ — среднее по Коши $\forall x$
- F 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):
$$F(b_1, \ldots, b_T, \alpha) = \sum_{t=1}^{T} \alpha_t b_t, \quad \sum_{t=1}^{T} \alpha_t = 1, \quad \alpha_t \geqslant 0$$
- смесь моделей-экспертов (mixture of experts)
с функциями компетентности (gating function) $g_t : X \to \mathbb{R}$
$$F(b_1, \ldots, b_T, x, \alpha) = \sum_{t=1}^{T} g_t(x, \alpha_t) b_t(x)$$

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 b_t 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 подвыборки «без возвращения»
- random subspace method, RSM — случ. подмн-ва признаков
- random patches — случ. подмн-ва и объектов, и признаков
- cross-validated committee — 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 $G \subseteq F^n = \{f_1, \ldots, f_n\}$

$$Q(b, U) = \sum_{x_i \in U} \mathscr{L}\big(b(x_i, w), y_i\big) \to \min_w \text{ — миним. эмпирич. риска}$$

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 ℓ' объектов из X^ℓ;
  $G_t :=$ случайное подмножество n' признаков 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 $x_i \notin U_t$.

Преобразование простого голосования во взвешенное

- **Линейная модель** над готовыми признаками $b_t(x)$:

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

- **Обучение:** МНК для регрессии, LR для классификации:

$$Q(\alpha, X^\ell) = \sum_{i=1}^{\ell} \mathscr{L}\big(b(x_i), y_i\big) \to \min_{\alpha}.$$

Регуляризация: $\alpha_t \geqslant 0$ либо LASSO: $\sum_{t=1}^{T} |\alpha_t| \leqslant \varkappa$.

- **Наивный байесовский классификатор** предполагает независимость с.в. $b_t(x)$ и даёт аналитическое решение:

$$\alpha_t = \ln \frac{1 - p_t}{p_t}, \quad t = 1, \ldots, T,$$

p_t — оценка вероятности ошибки базового алгоритма b_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

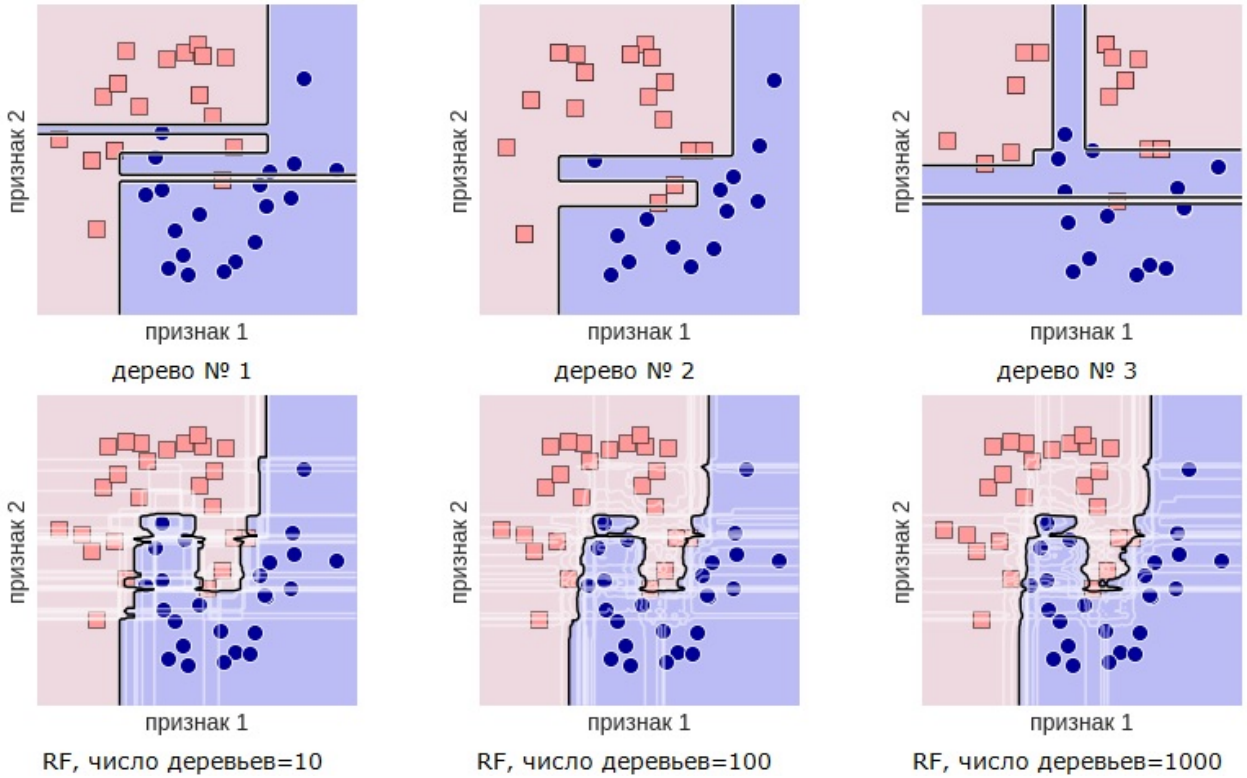
Параметры, которые можно настраивать (в частности, по OOB):

- число T деревьев
- число k случайно выбираемых признаков
- максимальная глубина деревьев
- минимальное число объектов в расщепляемой подвыборке
- минимальное число объектов в листьях
- 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 объединения bagging + RSM
- 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

Бустинг для задачи бинарной классификации

Дано: X^ℓ, $Y = \{\pm 1\}$, $b_t \colon X \to \{-1, 0, +1\}$, $C(b) = \text{sign}(b)$
$b_t(x) = 0$ означает отказ от классификации
(в частности, b_t могут быть логическими закономерностями)

Найти ансамбль со взвешенным голосованием:

$$a(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t b_t(x)\right), \quad x \in X$$

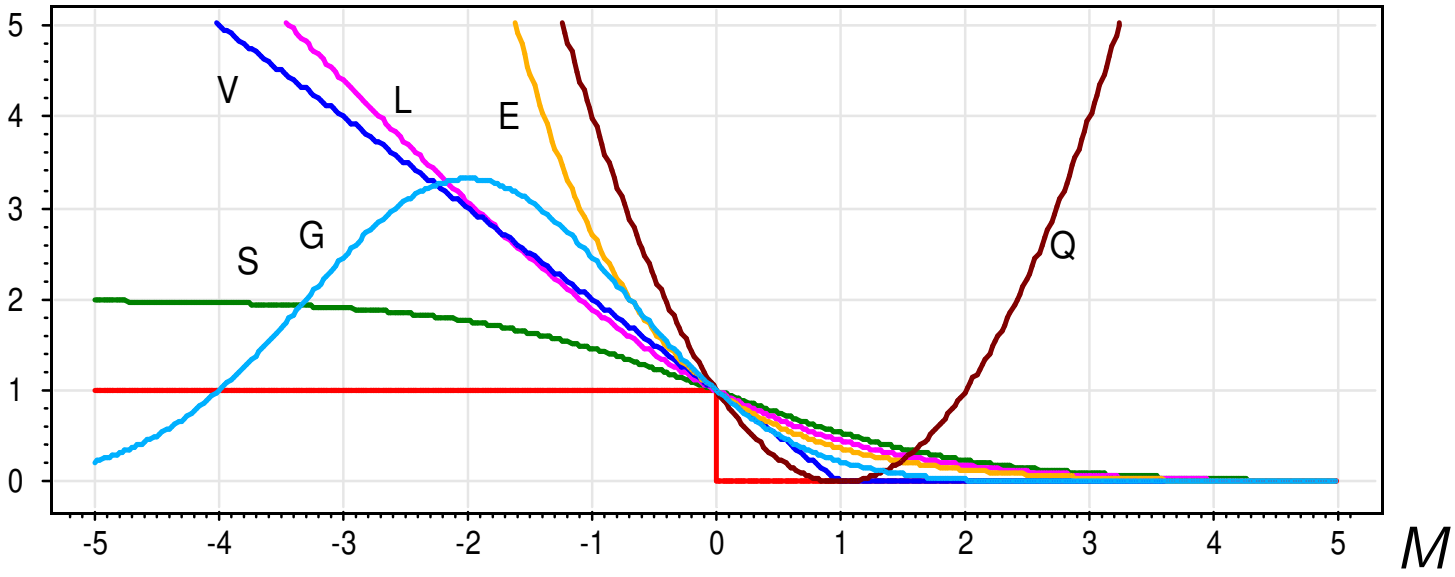
Критерий — минимум числа ошибок ансамбля на обучении:

$$Q_T = \sum_{i=1}^{\ell} \left[y_i \sum_{t=1}^{T} \alpha_t b_t(x_i) < 0 \right] \to \min_{\{\alpha_t, b_t\}}$$

Две основные эвристики бустинга:

- фиксация $\alpha_1 b_1(x), \ldots, \alpha_{t-1} b_{t-1}(x)$ при добавлении $\alpha_t b_t(x)$
- гладкая аппроксимация пороговой функции потерь $[M < 0]$

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);

Экспоненциальная аппроксимация пороговой функции потерь

Оценка функционала качества Q_T сверху:

$$Q_T \leqslant \widetilde{Q}_T = \sum_{i=1}^{\ell} \underbrace{\exp\left(-y_i \sum_{t=1}^{T-1} \alpha_t b_t(x_i)\right)}_{w_i} \exp\left(-y_i \alpha_T b_T(x_i)\right)$$

Нормированные веса: $\widetilde{W}^\ell = (\tilde{w}_1, \ldots, \tilde{w}_\ell)$, $\tilde{w}_i = w_i \,/\, \sum_{j=1}^{\ell} w_j$.

Взвешенная доля ошибочных (negative) и правильных (positive) классификаций с нормированными весами $\tilde{w}_1, \ldots, \tilde{w}_\ell$:

$$N(b, \widetilde{W}^\ell) = \sum_{i=1}^{\ell} \tilde{w}_i \left[b(x_i) = -y_i\right]; \quad P(b, \widetilde{W}^\ell) = \sum_{i=1}^{\ell} \tilde{w}_i \left[b(x_i) = y_i\right]$$

$1 - N - P$ — взвешенная доля отказов от классификации.

Y.Freund, R.E.Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. 1995

Основная теорема бустинга (для AdaBoost)

Если базовая модель $\mathscr{B}$ достаточно богатая, то, начиная с некоторого T, ансамбль не допускает ошибок на выборке X^ℓ

Теорема (Freund, Schapire, 1996)

Пусть для любого нормированного вектора весов U^ℓ существует алгоритм $b \in \mathscr{B}$, классифицирующий выборку хотя бы немного лучше, чем наугад: $P(b; U^\ell) > N(b; U^\ell)$.

Тогда минимум функционала $\widetilde{Q}_T$ достигается при

$$b_T = \arg\max_{b \in \mathscr{B}} \sqrt{P(b; \widetilde{W}^\ell)} - \sqrt{N(b; \widetilde{W}^\ell)}.$$

$$\alpha_T = \frac{1}{2} \ln \frac{P(b_T; \widetilde{W}^\ell)}{N(b_T; \widetilde{W}^\ell)}.$$

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

Доказательство (шаг 1 из 2)

Воспользуемся тождеством $\forall \alpha \in \mathbb{R},\ \forall b \in \{-1, 0, +1\}$:
$e^{-\alpha b} = e^{-\alpha}[b=1] + e^{\alpha}[b=-1] + [b=0]$.

Положим для краткости $\alpha = \alpha_T$ и $b_i = b_T(x_i)$. Тогда

$$\widetilde{Q}_T = \Bigg(e^{-\alpha}\underbrace{\sum_{i=1}^{\ell}\tilde{w}_i[b_i=y_i]}_{P} + e^{\alpha}\underbrace{\sum_{i=1}^{\ell}\tilde{w}_i[b_i=-y_i]}_{N} + \underbrace{\sum_{i=1}^{\ell}\tilde{w}_i[b_i=0]}_{1-P-N}\Bigg)\underbrace{\sum_{i=1}^{\ell} w_i}_{\widetilde{Q}_{T-1}}$$

$$= \big(e^{-\alpha}P + e^{\alpha}N + (1-P-N)\big)\widetilde{Q}_{T-1} \to \min_{\alpha,b}.$$

$$\frac{\partial}{\partial\alpha}\widetilde{Q}_T = \big(-e^{-\alpha}P + e^{\alpha}N\big)\widetilde{Q}_{T-1} = 0 \ \Rightarrow\ e^{-\alpha}P = e^{\alpha}N \ \Rightarrow\ e^{2\alpha} = \frac{P}{N}.$$

Получили требуемое: $\boxed{\alpha_T = \frac{1}{2}\ln\frac{P}{N}}$.

Доказательство (шаг 2 из 2)

Подставим оптимальное значение $\alpha = \frac{1}{2}\ln\frac{P}{N}$ обратно в $\widetilde{Q}_T$:

$$\begin{aligned}\widetilde{Q}_T &= \left(e^{-\alpha}P + e^{\alpha}N + (1-P-N)\right)\widetilde{Q}_{T-1} = \\ &= \left(1 + \sqrt{\tfrac{N}{P}}P + \sqrt{\tfrac{P}{N}}N - P - N\right)\widetilde{Q}_{T-1} = \\ &= \left(1 - \left(\sqrt{P} - \sqrt{N}\right)^2\right)\widetilde{Q}_{T-1} \to \min_b .\end{aligned}$$

Поскольку $\widetilde{Q}_{T-1}$ не зависит от α_T и b_T, минимизация $\widetilde{Q}_T$ эквивалентна либо максимизации $\sqrt{P} - \sqrt{N}$ при $P > N$, либо максимизации $\sqrt{N} - \sqrt{P}$ при $P < N$, однако второй случай исключён условием теоремы.

Получили $\boxed{b_T = \arg\max_b \sqrt{P} - \sqrt{N}}$. Теорема доказана.

Следствие 1. Сходимость

Теорема

Если на каждом шаге семейство $\mathscr{B}$ и метод обучения обеспечивают построение базового алгоритма b_t такого, что

$$\sqrt{P(b_t; \widetilde{W}^\ell)} - \sqrt{N(b_t; \widetilde{W}^\ell)} = \gamma_t > \gamma$$

при некотором $\gamma > 0$, то за конечное число шагов будет построен алгоритм $a(x)$, не допускающий ошибок на X^ℓ.

Доказательство. Q_T сходится к нулю со скоростью геометрической прогрессии:

$$Q_{T+1} \leqslant \widetilde{Q}_{T+1} = \widetilde{Q}_T(1-\gamma^2) \leqslant \cdots \leqslant \widetilde{Q}_1(1-\gamma^2)^T.$$

Наступит момент, когда $\widetilde{Q}_T < 1$.
Но тогда $Q_T = 0$, поскольку $Q_T \in \{0, 1, \ldots, \ell\}$.

Следствие 2. Исходный (классический) вариант AdaBoost

Пусть отказов нет, $b_t : X \to \{\pm 1\}$. Тогда $P = 1 - N$.

Теорема (Freund, Schapire, 1995)

Пусть для любого нормированного вектора весов U^ℓ существует алгоритм $b \in \mathscr{B}$, классифицирующий выборку хотя бы немного лучше, чем наугад: $N(b; U^\ell) < \frac{1}{2}$.

Тогда минимум функционала $\widetilde{Q}_T$ достигается при

$$b_T = \arg\min_{b \in \mathscr{B}} N(b; \widetilde{W}^\ell).$$

$$\alpha_T = \frac{1}{2} \ln \frac{1 - N(b_T; \widetilde{W}^\ell)}{N(b_T; \widetilde{W}^\ell)}.$$

Y.Freund, R.E.Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. 1995

Алгоритм AdaBoost (в исходном варианте)

Вход: обучающая выборка X^ℓ; параметр T;

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

инициализировать веса объектов: $w_i := 1/\ell$, $i = 1, \ldots, \ell$;

для всех $t = 1, \ldots, T$:

обучить b_t, минимизируя взвешенный эмпирический риск:

$$b_t := \arg\min_b N(b; W^\ell);$$

$$\alpha_t := \frac{1}{2} \ln \frac{1 - N(b_t; W^\ell)}{N(b_t; W^\ell)};$$

обновить веса объектов:

$$w_i := w_i \exp\bigl(-\alpha_t y_i b_t(x_i)\bigr), \quad i = 1, \ldots, \ell;$$

нормировать веса объектов:

$$w_0 := \sum_{j=1}^{\ell} w_j;$$

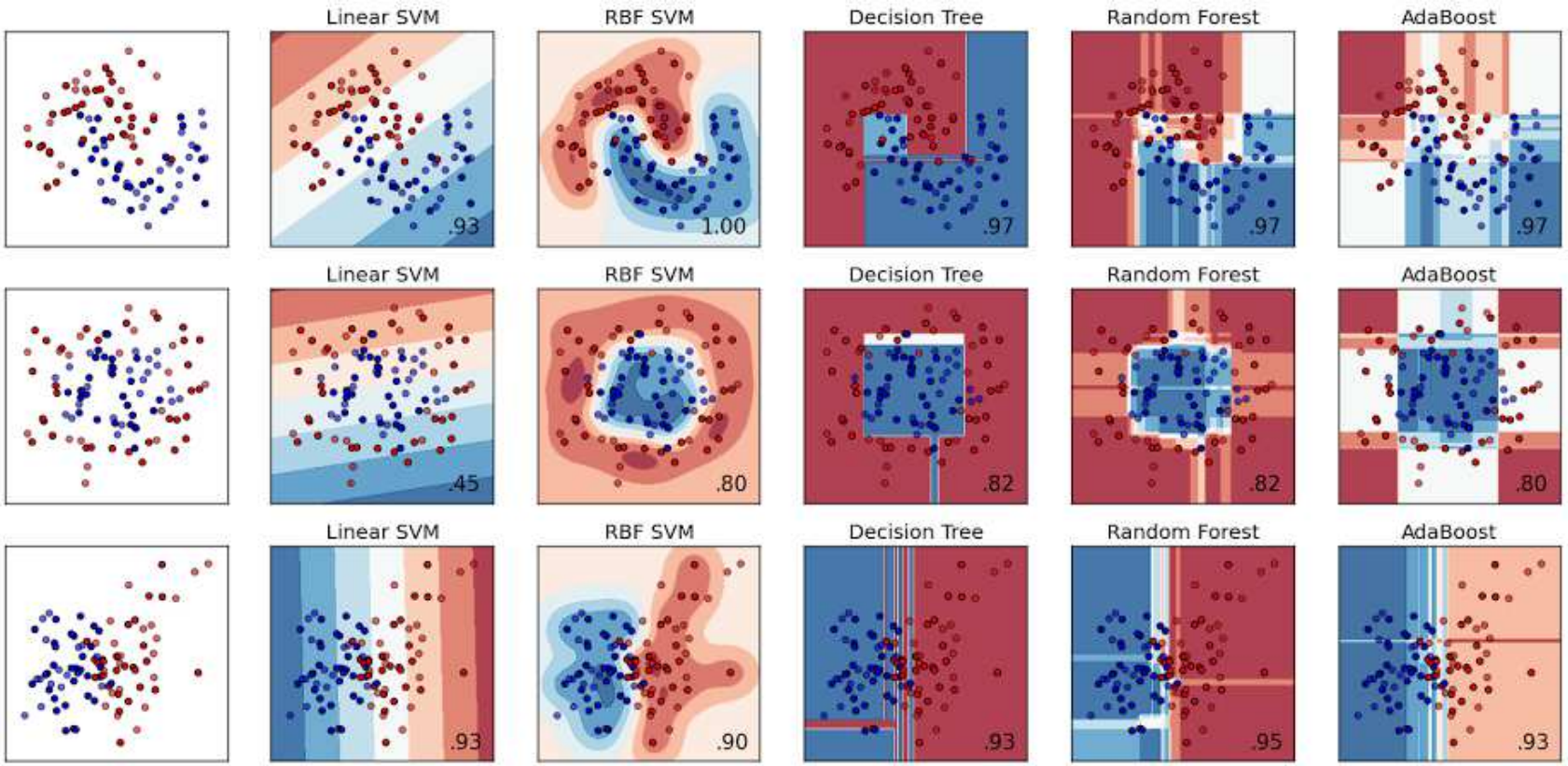
$$w_i := w_i / w_0, \quad i = 1, \ldots, \ell;$$

Эвристики и рекомендации

- **Базовые классификаторы (weak classifiers):**
  — решающие деревья — используются чаще всего
  — пороговые правила, т.н. «решающие пни» (data stumps)
  $$\mathscr{B} = \left\{ b(x) = \left[f_j(x) \leqslant \theta \right] \,\middle|\, j = 1, \ldots, n,\ \theta \in \mathbb{R} \right\}$$
  — логические закономерности: $b_t(x) \in \{0, +1\}$ или $\{0, -1\}$
  — для сильных и устойчивых базовых моделей (например, SVM) бустинг не эффективен
- **Отсев шума:** отбросить объекты с наибольшими w_i
- **Модификация формулы для α_t на случай $N = 0$:**
  $$\alpha_t := \frac{1}{2} \ln \frac{1 - N(b_t; W^\ell) + \frac{1}{\ell}}{N(b_t; W^\ell) + \frac{1}{\ell}}$$
- **Критерий ранней остановки** (early stop):
  увеличение частоты ошибок на контрольной выборке

Случайный лес и бустинг в сравнении с другими методами

Эксперименты на трёх двумерных синтетических выборках:

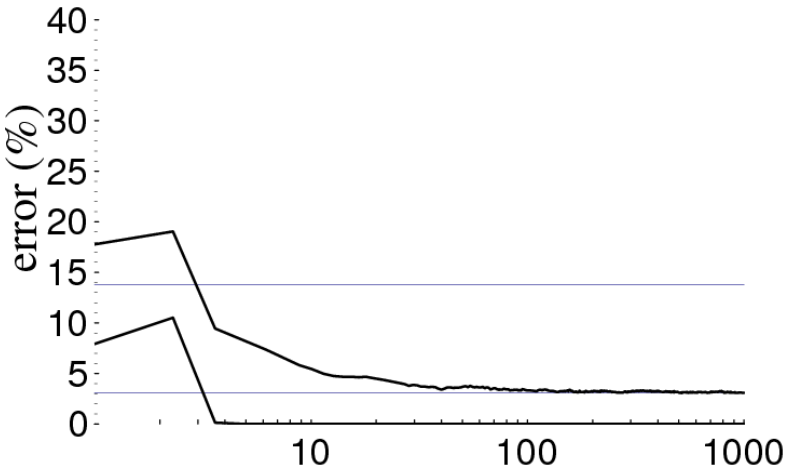


Решения могут выглядеть странно... тем не менее, RF и бустинг считаются наиболее сильными универсальными методами в ML

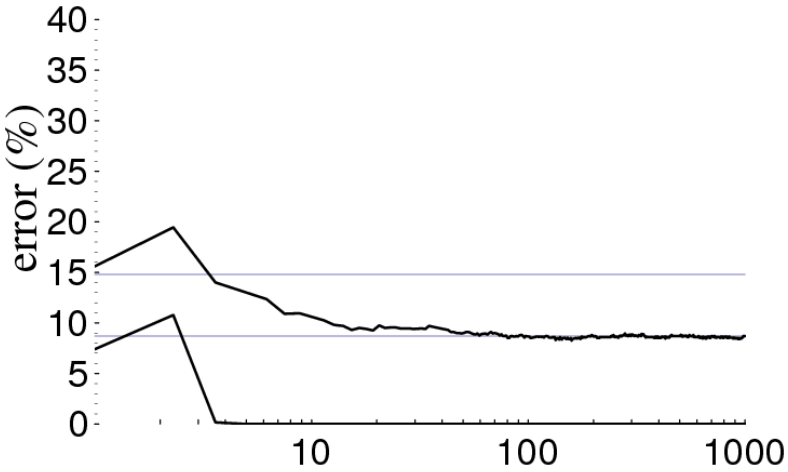
Эксперименты с алгоритмом классификации AdaBoost

Удивительное отсутствие переобучения вплоть до $T = 1000$ (нижняя кривая — обучение, верхняя – тест):

Задача UCI:letter



Задача UCI:satimage

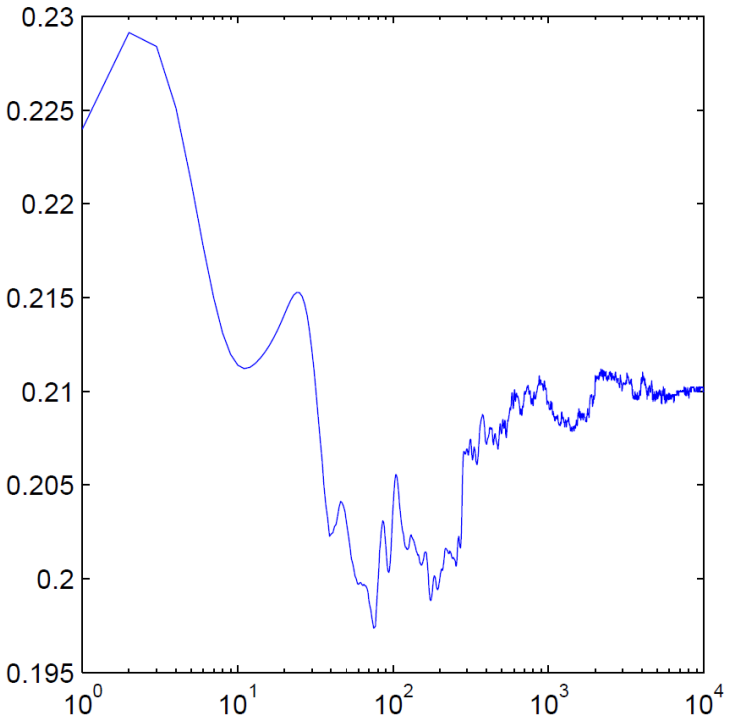


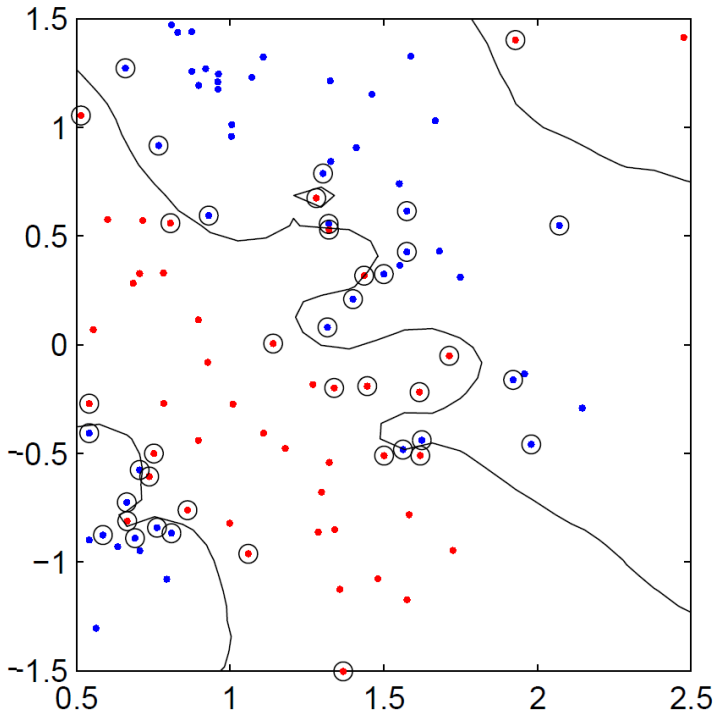
До этих экспериментов считалось, что увеличение сложности модели (числа параметров) неизбежно ведёт к переобучению

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.

Иногда AdaBoost всё же переобучается...

... но не сильно, и на тысячах базовых классификаторах.
Слева: зависимость ошибки на тестовой выборке от $|T|$.
Справа: разделяющая поверхность при переобучении.





G.Rätsch, T.Onoda, K.R.Müller. An improvement of AdaBoost to avoid overfitting. 1998.

Сравнение бэггинга и бустинга

Что общего:

- усиление различности базовых алгоритмов + голосование
- простая обёртка над алгоритмом обучения базовой модели
- оба метода примерно одинаково успешны

В чём отличия boosting / bagging / RSM:

- бустинг взвешивает как объекты, так и базовые алгоритмы
- бустинг лучше для классов с границами сложной формы
- бэггинг и RSM лучше для коротких обучающих выборок
- RSM лучше для зависимых и неинформативных признаков

Распараллеливание:

- бэггинг может параллельно обучать базовые алгоритмы b_t
- бустинг вынужден параллельно обучать каждый b_t

Недостатки AdaBoost

- чрезмерная чувствительность к выбросам из-за e^{-M}
- неинтерпретируемое нагромождение из сотен алгоритмов
- не удаётся строить короткие композиции из «сильных» алгоритмов типа SVM (только длинные из «слабых»)
- требуются достаточно большие обучающие выборки (бэггинг обходится более короткими)

Способы устранения:

- отсев выбросов по критерию увеличения веса w_i
- обучение b_t по случайным подвыборкам как в бэггинге
- другие функции потерь (градиентный бустинг, см. далее)
- явная оптимизация распределения отступов (комитетный бустинг, direct optimization of margin, см. далее)

Резюме

- Ансамбли позволяют решать сложные задачи, которые плохо решаются отдельными базовыми алгоритмами
- Обычно ансамбль строится *алгоритмом-обёрткой* (envelop): базовые алгоритмы обучаются готовыми методами
- Базовые алгоритмы: компромисс качество/различность
- Две основные эвристики бустинга (и не только AdaBoost):
  — обучать базовые алгоритмы по одному
  — использовать гладкую замену пороговой функции потерь
- Важное открытие середины 90-х: обобщающая способность бустинга не ухудшается с ростом сложности T
- Практическое сравнение бустинга и бэггинга:
  — бустинг лучше для классов с границами сложной формы
  — бэггинг лучше для коротких обучающих выборок
  — бэггинг легче распараллеливается