Композиции классификаторов (часть 1)

К. В. Воронцов, А.В. Зухба
vokov@forecsys.ru
a__l@mail.ru

сентябрь 2016

Содержание

Определение композиции

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

$a(x) = C(b(x))$ — алгоритм, где

$b\colon X \to R$ — *базовый алгоритм* (алгоритмический оператор),

$C\colon R \to Y$ — *решающее правило*,

R — *пространство оценок*;

Определение

Композиция базовых алгоритмов $b_1, \ldots, b_T$

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

где $F\colon R^T \to R$ — *корректирующая операция*.

Зачем вводится R?

В задачах классификации множество отображений $\{F\colon R^T \to R\}$ существенно шире, чем $\{F\colon Y^T \to Y\}$.

Примеры пространств оценок и решающих правил

- **Пример 1:** классификация на 2 класса, $Y = \{-1, +1\}$:

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

где $R = \mathbb{R}$, $b\colon X \to \mathbb{R}$, $C(b) \equiv \text{sign}(b)$.

- **Пример 2:** классификация на M классов $Y = \{1, \ldots, M\}$:

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

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

- **Пример 3:** регрессия, $Y = R = \mathbb{R}$:
$C(b) \equiv b$ — решающее правило не нужно.

Примеры корректирующих операций

- **Пример 1:** Простое голосование (Simple Voting):

$$F\big(b_1(x),\ldots,b_T(x)\big) = \frac{1}{T}\sum_{t=1}^{T} b_t(x), \quad x \in X.$$

- **Пример 2:** Взвешенное голосование (Weighted Voting):

$$F\big(b_1(x),\ldots,b_T(x)\big) = \sum_{t=1}^{T} \alpha_t b_t(x), \quad x \in X, \quad \alpha_t \in \mathbb{R}.$$

- **Пример 3:** Смесь алгоритмов (Mixture of Experts)

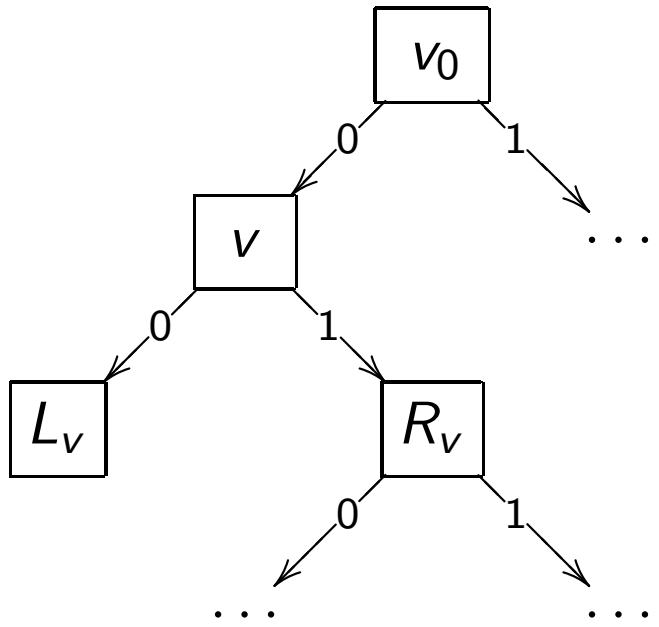
$$F\big(b_1(x),\ldots,b_T(x)\big) = \sum_{t=1}^{T} g_t(x) b_t(x), \quad x \in X, \quad g_t \colon X \to \mathbb{R}.$$

Определение бинарного решающего дерева

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

1) $\forall v \in V_{\text{внутр}} \;\to\;$ предикат $\beta_v : X \to \{0,1\}$, $\beta_v \in \mathscr{B}$
2) $\forall v \in V_{\text{лист}} \;\to\;$ имя класса $c_v \in Y$.

1: $v := v_0$;
2: **пока** $v \in V_{\text{внутр}}$
3: **если** $\beta_v(x) = 1$ **то**
4: переход вправо:
$v := R_v$;
5: **иначе**
6: переход влево:
$v := L_v$;
7: **вернуть** c_v.



Линейные алгоритмы

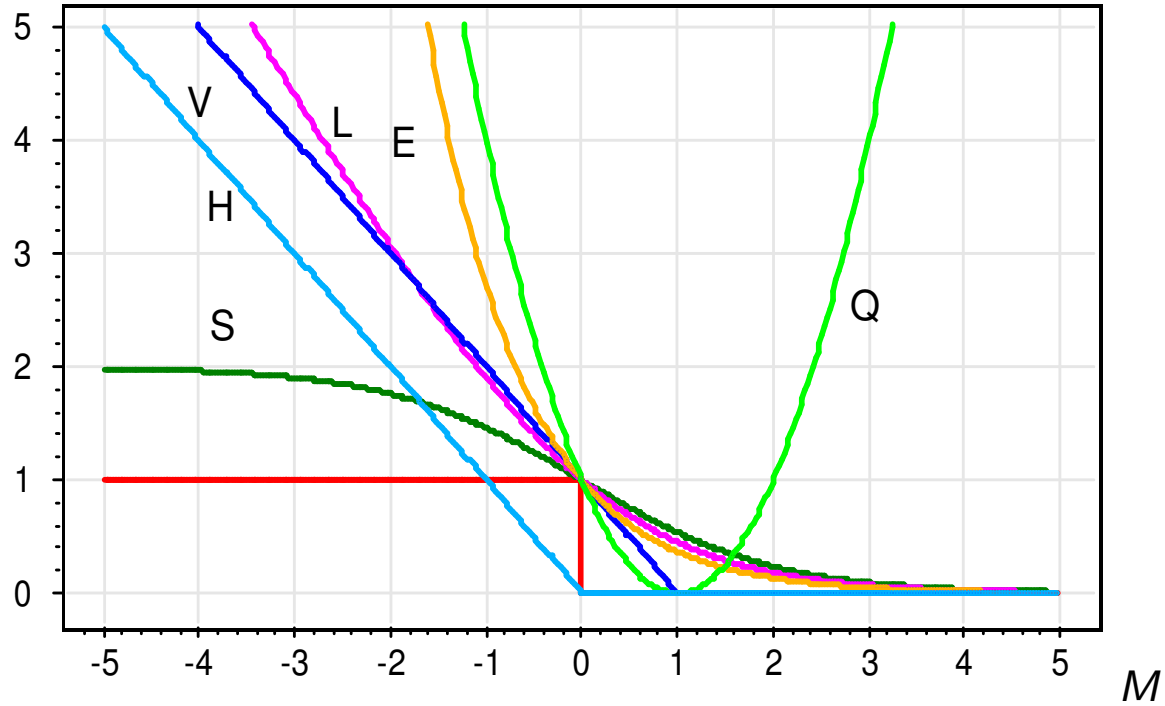
- Задача классификации с двумя классами, $Y = \{-1, +1\}$;
  обучающая выборка $X^\ell = (x_i, y_i)_{i=1}^\ell$;
  алгоритм классификации $a(x, w) = \operatorname{sign} f(x, w)$,
  $f(x, w)$ — *дискриминантная функция*,
  w — вектор параметров.
- $f(x, w) = 0$ — разделяющая поверхность;
  $M_i(w) = y_i f(x_i, w)$ — *отступ* (margin) объекта x_i;
  $M_i(w) < 0 \iff$ алгоритм $a(x, w)$ ошибается на x_i.
- *Минимизация эмпирического риска*:

$$Q(w) = \sum_{i=1}^{\ell} \big[M_i(w) < 0\big] \quad \leqslant \quad \widetilde{Q}(w) = \sum_{i=1}^{\ell} \mathscr{L}\big(M_i(w)\big) \to \min_w;$$

функция потерь $\mathscr{L}(M)$ невозрастающая, неотрицательная.

Непрерывные аппроксимации пороговой функции потерь

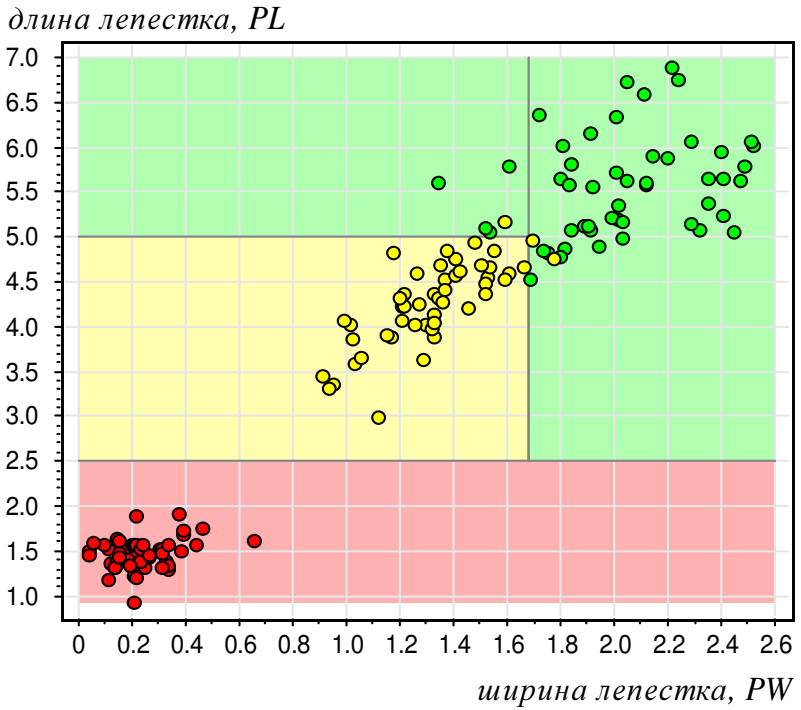
Часто используемые функции потерь $\mathscr{L}(M)$:

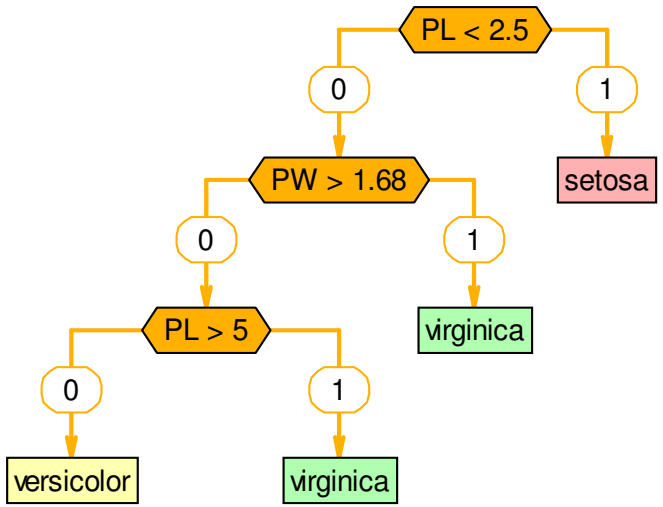


$V(M) = (1 - M)_+$ — кусочно-линейная (SVM);
$S(M) = 2(1 + e^M)^{-1}$ — сигмоидная (ANN);
$L(M) = \log_2(1 + e^{-M})$ — логарифмическая (LR);
$E(M) = e^{-M}$ — экспоненциальная (AdaBoost);
$Q(M) = (1 - M)^2$ — квадратичная (FLD).

Пример решающего дерева

Задача Фишера о классификации цветков ириса на 3 класса, в выборке по 50 объектов каждого класса, 4 признака.





На графике: в осях двух самых информативных признаков (из 4) два класса разделились без ошибок, на третьем 3 ошибки.

Стохастические методы построения композиций

Чтобы алгоритмы в композиции были различными

- их обучают по (случайным) подвыборкам,
- либо по (случайным) подмножествам признаков.

Первую идею реализует bagging (bootstrap aggregation) [Breiman, 1996], причём подвыборки берутся длины ℓ с возвращениями, как в методе bootstrap.

Вторую идею реализует RSM (random subspace method) [Duin, 2002].

Совместим обе идеи в одном алгоритме.

$\mathscr{F} = \{f_1, \ldots, f_n\}$ — признаки,
$\mu(\mathscr{G}, U)$ — метод обучения алгоритма по подвыборке $U \subseteq X^\ell$, использующий только признаки из $\mathscr{G} \subseteq \mathscr{F}$.

Бэггинг и метод случайных подпространств

Вход: обучающая выборка X^ℓ; параметры: T

ℓ' — длина обучающих подвыборок;

n' — длина признакового подописания;

ε_1 — порог качества базовых алгоритмов на обучении;

ε_2 — порог качества базовых алгоритмов на контроле;

Выход: базовые алгоритмы b_t, $t = 1, \ldots, T$;

1: **для всех** $t = 1, \ldots, T$
2: $U :=$ случайное подмножество X^ℓ длины ℓ';
3: $\mathscr{G} :=$ случайное подмножество $\mathscr{F}$ длины n';
4: $b_t := \mu(\mathscr{G}, U)$;
5: **если** $Q(b_t, U) > \varepsilon_1$ или $Q(b_t, X^\ell \setminus U) > \varepsilon_2$ **то**
6: не включать b_t в композицию;

Композиция — простое голосование: $a(x) = C\Big(\sum_{t=1}^{T} b_t(x)\Big)$.

Сравнение: boosting — bagging — RSM

- Бустинг лучше для больших обучающих выборок и для классов с границами сложной формы.
- Бэггинг и RSM лучше для коротких обучающих выборок.
- RSM лучше в тех случаях, когда признаков больше, чем объектов, или когда много неинформативных признаков.
- Бэггинг и RSM эффективно распараллеливаются, бустинг выполняется строго последовательно.

И ещё несколько эмпирических наблюдений:

- Веса алгоритмов не столь важны для выравнивания отступов.
- Веса объектов не столь важны для обеспечения различности.
- Не удаётся строить короткие композиции из «сильных» алгоритмов типа SVM (только длинные из слабых).

Возможно ли строить композиции проще и аккуратнее?

Простое голосование в задаче классификации

Возьмём $Y = \{\pm 1\}$, $F(b_1, \ldots, b_T) = \frac{1}{T}\sum_{t=1}^{T} b_t$, $C(b) = \text{sign}(b)$.

Функционал качества композиции — 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.

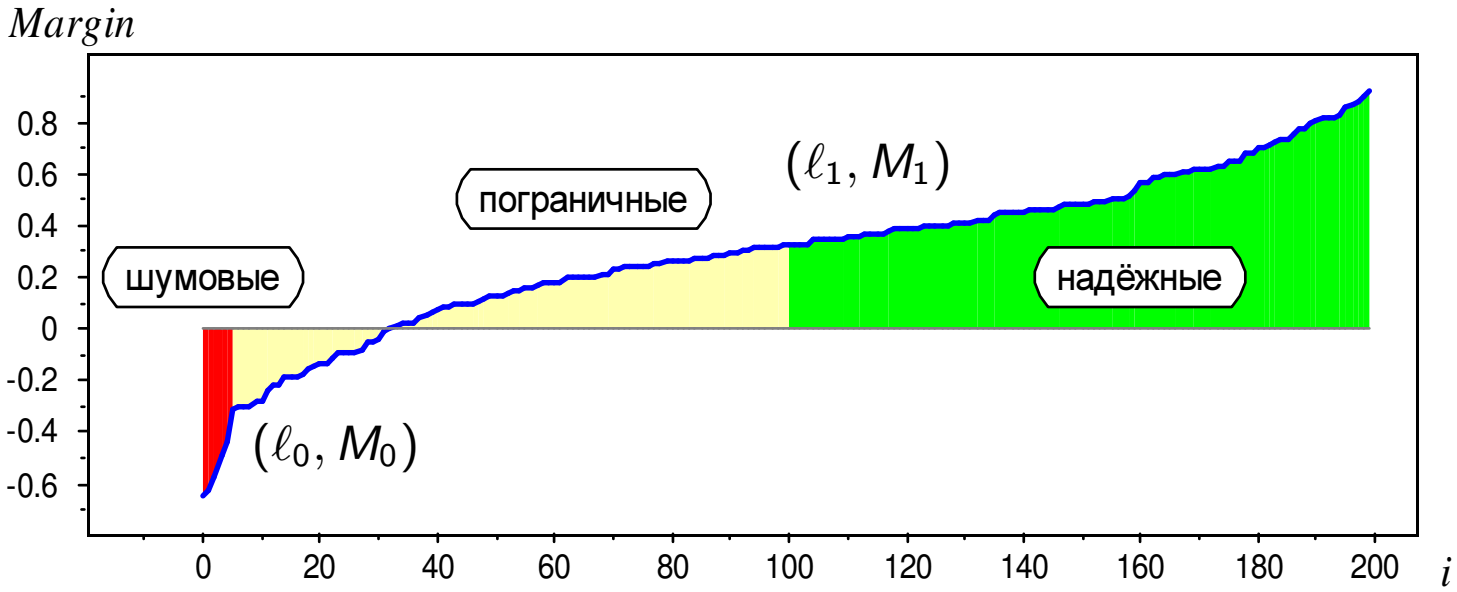
Эвристика: чтобы b_{t+1} компенсировал ошибки композиции,

$$Q(b, U) = \sum_{x_i \in U} \big[y_i b(x_i) < 0\big] \to \min_b,$$

где $U = \big\{x_i : M_0 < M_{it} \leqslant M_1\big\}$,
M_0, M_1 — параметры метода обучения.

Подбор параметров M_0 и M_1

Упорядочим объекты по возрастанию отступов M_{it}:



Принцип максимизации и выравнивания отступов.

Два случая, когда b_{t+1} на объекте x_i обучать не надо:

$M_{it} < M_0, \;\; i < \ell_0$ — объект x_i шумовой;

$M_{it} > M_1, \;\; i > \ell_1$ — объект x_i уже надёжно классифицируется.

Основные понятия и определения

Задача регрессии: $Y = \mathbb{R}$
Квадратичная функция потерь: $L(y, a) = \big(a(x) - y\big)^2$
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 $\to$ алгоритм

Среднеквадратичный риск:

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

Минимум среднеквадратичного риска, «недостижимый идеал»:

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

Основная мера качества метода обучения μ:

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

Разложение ошибки на шум, вариацию и смещение

Теорема

В случае квадратичной функции потерь для любого μ

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

$\bar{a}(x) = \mathsf{E}_{X^\ell}\big(\mu(X^\ell)(x)\big)$ — средний ответ обученного алгоритма

Метод k ближайших соседей

Вероятностная модель данных: $p(y|x) = f(x) + \mathcal{N}(0, \sigma^2)$
Метод k ближайших соседей:

$$a(x) = \frac{1}{k} \sum_{j=1}^{k} y(x^{(j)}),$$

где $x^{(j)}$ — j-й сосед объекта x
$a^*(x) = f(x)$ — истинная зависимость
$\bar{a}(x) = \frac{1}{k} \sum_{j=1}^{k} f(x^{(j)})$ — средний ответ

Разложение bias-variance:

$$Q(\mu) = \underbrace{\sigma^2}_{\text{шум}} + \underbrace{\mathsf{E}_{x,y}\Big(\bar{a}(x) - f(x)\Big)^2}_{\text{смещение}} + \underbrace{\tfrac{1}{k}\sigma^2}_{\text{разброс}}$$

Простое голосование

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

Композиция — простое голосование: $a_T(x) = \frac{1}{T}\sum\limits_{t=1}^{T} b_t(x)$

Смещение композиции 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 ковариации:

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