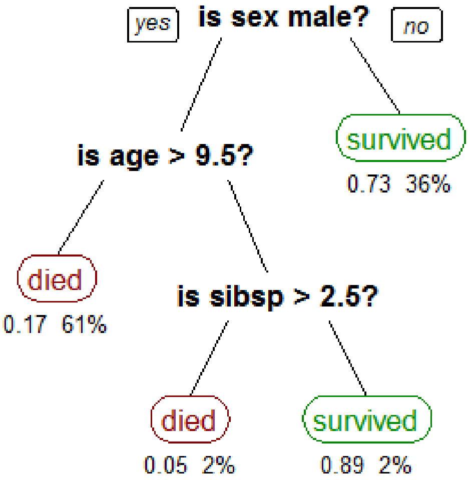
Лекция 4
Бустинг

Напоминание: решающие деревья и решающие леса

Решающее дерево



Критерий информативности

$\{x_i, y_i\}$ — выборка
$A = \{i_1, \dots, i_n\}$ — индексы подвыборки

Классификация (k классов)

Энтропийный критерий

$$H(A) = \sum_{i=1}^{k} p_i \log p_i, \quad p_k = \frac{1}{|A|} \sum_{i \in A} [y_i = k]$$

Регрессия

Критерий дисперсии

$$H(A) = \frac{1}{|A|} \sum_{i \in A} (y_i - \overline{y})^2, \quad \overline{y} = \frac{1}{|A|} \sum_{i \in A} y_i$$

Обучение деревьев

Выбор разбиения для подвыборки A

- Рассматриваем критерии вида $[x^j < t]$ для $j = 1, \dots, d$ (признаки) и $t \in \mathbb{R}$.
- Качество разбиения A на A_l и A_r:

$$Q(A, j, t) = H(A) - \frac{|A_l|}{|A|} H(A_l) - \frac{|A_r|}{|A|} H(A_r)$$

- Наилучшее разбиение:

$$Q(A, j, t) \to \max_{j,t}$$

Критерии остановки разбиения

- Глубина
- Размер подвыборки

Решающий лес

$a_1(x), \ldots, a_n(x)$ — набор решающих деревьев

$a(x) =$ композиция$(a_1(x), \ldots, a_n(x))$ — решающий лес

Композиция для классификации

$$a(x) = \arg\max_y \sum_{i=1}^{n} [a_i(x) = y]$$

Компопзиция для регрессии

$$a(x) = \frac{1}{n} \sum_{i=1}^{n} a_i(x)$$

Обучение лесов

Как строить набор деревьев $a_1(x), \ldots, a_n(x)$?

- **Бэггинг**
  Каждое дерево обучается на случайной подвыборке
- **Метод случайных подпространств**
  Каждое дерево обучается на случайном подмножестве признаков
- **Случайный лес**
  Каждое разбиение каждого дерева рассматривает случайное подмножество признаков

Бустинг — введение

Общая идея (регрессия)

- **Композиция с помощью суммы**

$$a(x) = a_1(x) + a_2(x) + \ldots + a_K(x)$$

- **Обучение на ошибках**
  $a_k(x)$ обучается на ошибках $a_1(x) + \ldots + a_{k-1}(x)$.

Бустинг (регрессия)

$L = \{x_i, y_i\}_{i=1}^N$ — обучающая выборка

K — количество деревьев

- Инициализация

$$a(x) := 0, \quad r_i = y_i \text{ для } i = 1, \ldots, N$$

- Для $k = 1, \ldots, K$
  - Новая обучающая выборка

$$L' = \{x_i, r_i\}_{i=1}^N$$

  - Обучение решающего дерева a_k на L'
  - Обновление алгоритма и ошибки

$$a(x) := a(x) + a_k(x)$$

$$r_i := y_i - a(x_i) \text{ для } i = 1, \ldots N$$

Бустинг (регрессия)

$L = \{x_i, y_i\}_{i=1}^N$ — обучающая выборка

K — количество деревьев, λ — скорость обучения

- Инициализация

$$a(x) := 0, \quad r_i = y_i \text{ для } i = 1, \ldots, N$$

- Для $k = 1, \ldots, K$
  - Новая обучающая выборка

$$L' = \{x_i, r_i\}_{i=1}^N$$

  - Обучение решающего дерева a_k на L'
  - Обновление алгоритма и ошибки

$$a(x) := a(x) + \lambda a_k(x)$$

$$r_i := y_i - a(x_i) \text{ для } i = 1, \ldots N$$

Особенности бустинга

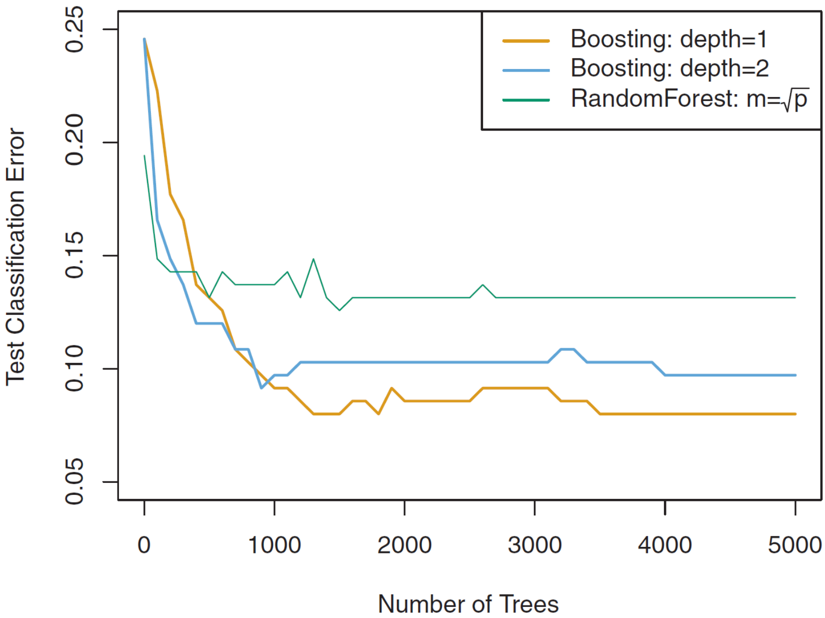
- **Переобучение**
  Нужна сильная регуляризация деревьев.
  Например: глубина 1 или 2.

Особенности бустинга

- **Переобучение**
  Нужна сильная регуляризация деревьев.
  Например: глубина 1 или 2.
- **Медленное обучение**
  Сильная регуляризация $\rightarrow$ медленная сходимость

Особенности бустинга

- **Переобучение**
  Нужна сильная регуляризация деревьев.
  Например: глубина 1 или 2.
- **Медленное обучение**
  Сильная регуляризация $\rightarrow$ медленная сходимость
- **Высокая эффективность**
  (сильная регуляризация + много итераций)


Boosting: depth=1
Boosting: depth=2
RandomForest: m=√p
Test Classification Error
0.05
0.10
0.15
0.20
0.25
0
1000
2000
3000
4000
5000
Number of Trees

AdaBoost

Бустинг для классификации v.1.0

$\{x_i, y_i\}_{i=1}^N$ — обучающая выборка

$y_i \in \{-1, +1\}$

$$a_i(x) \in \{-1, +1\} \text{ для } i = 1, \ldots, K$$

$$a(x) = a_1(x) + \ldots + a_K(x)$$

На итерации k: $\quad a_k(x) \sim \{x_i, r_i\}$

$$r_i = y_i - a(x_i) = y_i - (a_1(x_i) + \ldots + a_{k-1}(x_i))$$

Бустинг для классификации v.1.0

$\{x_i, y_i\}_{i=1}^N$ — обучающая выборка

$y_i \in \{-1, +1\}$

$$a_i(x) \in \{-1, +1\} \text{ для } i = 1, \ldots, K$$

$$a(x) = a_1(x) + \ldots + a_K(x)$$

На итерации k: $\quad a_k(x) \sim \{x_i, r_i\}$

$$r_i = y_i - a(x_i) = y_i - (a_1(x_i) + \ldots + a_{k-1}(x_i))$$

Проблемы

- Как обучаться на метках 0, 2, -2 и т.п.?

Бустинг для классификации v.2.0

$\{x_i, y_i\}_{i=1}^N$ — обучающая выборка

$y_i \in \{-1, +1\}$

$$a_i(x) \in \mathbb{R} \text{ для } i = 1, \ldots, K$$

$$a(x) = \operatorname{sign}(a_1(x) + \ldots + a_K(x))$$

На итерации k: $\quad a_k(x) \sim \{x_i, r_i\}$

$$r_i = y_i - a(x_i) = y_i - (a_1(x_i) + \ldots + a_{k-1}(x_i))$$

Бустинг для классификации v.2.0

$\{x_i, y_i\}_{i=1}^N$ — обучающая выборка

$y_i \in \{-1, +1\}$

$$a_i(x) \in \mathbb{R} \text{ для } i = 1, \ldots, K$$

$$a(x) = \operatorname{sign}(a_1(x) + \ldots + a_K(x))$$

На итерации k: $\quad a_k(x) \sim \{x_i, r_i\}$

$$r_i = y_i - a(x_i) = y_i - (a_1(x_i) + \ldots + a_{k-1}(x_i))$$

Проблемы

- Оптимизируется непонятно что
- Нужно $a_i(x) \in \mathbb{R}$

Оптимизация числа ошибок

$$a(x) = \text{sign}(a_1(x) + \ldots + a_k(x))$$

$$Q(a) = \sum_{i=1}^{N} [y_i \neq a(x_i)]$$

Оптимизация числа ошибок

$$a(x) = \operatorname{sign}(a_1(x) + \ldots + a_k(x))$$

$$Q(a) = \sum_{i=1}^{N} [y_i \neq a(x_i)]$$

Добавляем компонент:

$$Q(a, a_{k+1}) = \sum_{i=1}^{N} [y_i \neq \operatorname{sign}(a(x_i) + a_{k+1}(x_i))]$$

$$Q(a, a_{k+1}) \to \min_{a_{k+1}}$$

Оптимизация числа ошибок

$$a(x) = \operatorname{sign}(a_1(x) + \ldots + a_k(x))$$

$$Q(a) = \sum_{i=1}^{N} [y_i \neq a(x_i)]$$

Добавляем компонент:

$$Q(a, a_{k+1}) = \sum_{i=1}^{N} [y_i \neq \operatorname{sign}(a(x_i) + a_{k+1}(x_i))]$$

$$Q(a, a_{k+1}) \to \min_{a_{k+1}}$$

Дискретная функция $\Rightarrow$ непонятно как минимизировать

Аппроксимации пороговой функции потерь

$z = yf(x)$
y — ответ, $f(x)$ — решающая функция

$$T(z) = [z < 0]$$

Квадратичная

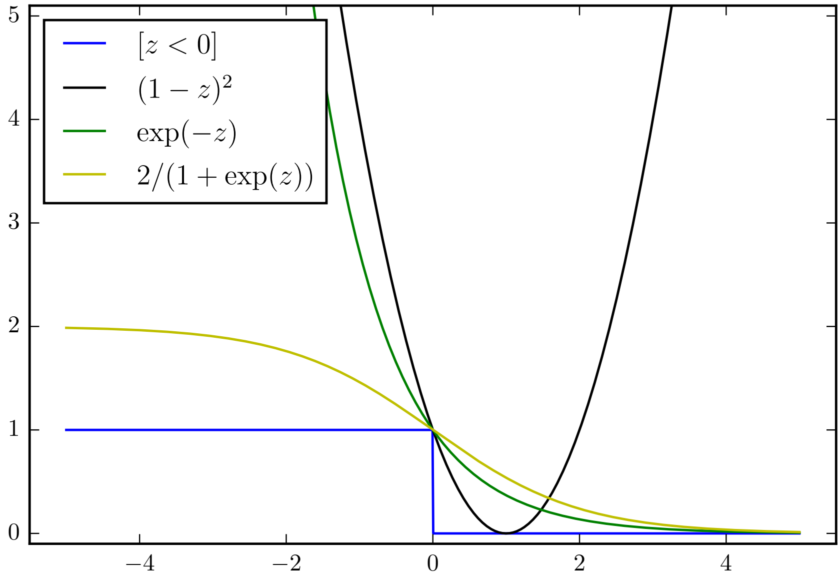
$$T(z) \leqslant (1 - z)^2$$

Экспоненциальная

$$T(z) \leqslant e^{-z}$$

Сигмоидная

$$T(z) \leqslant \frac{2}{1 + e^z}$$


$[z < 0]$
$(1 - z)^2$
$\exp(-z)$
$2/(1 + \exp(z))$
5
4
3
2
1
0
−4
−2
0
2
4

AdaBoost

$$a(x) = \operatorname{sign}(\alpha_1 t_1(x) + \ldots + \alpha_k t_k(x))$$

$$\alpha_j \in \mathbb{R}, \quad t_j(x) \in \{-1, +1\} \text{ для } j = 1, \ldots, k$$

$$Q(a) = \sum_{i=1}^{N} \exp\left(-y_i \sum_{j=1}^{k} \alpha_j t_j(x_i)\right)$$

AdaBoost

$$a(x) = \operatorname{sign}(\alpha_1 t_1(x) + \ldots + \alpha_k t_k(x))$$

$$\alpha_j \in \mathbb{R}, \quad t_j(x) \in \{-1, +1\} \text{ для } j = 1, \ldots, k$$

$$Q(a) = \sum_{i=1}^{N} \exp\left(-y_i \sum_{j=1}^{k} \alpha_j t_j(x_i)\right)$$

Добавляем компонент:

$$Q(a, \alpha_{k+1}, t_{k+1}) = \sum_{i=1}^{N} \exp\left(-y_i \left(\sum_{j=1}^{k} \alpha_j t_j(x_i) + \alpha_{k+1} t_{k+1}(x_i)\right)\right)$$

$$Q(a, \alpha_{k+1}, t_{k+1}) \to \min_{\alpha_{k+1}, t_{k+1}}$$

Преобразование функции потерь

$$
\begin{aligned}
&Q(a, \alpha_{k+1}, t_{k+1}) = \\
&\quad = \sum_{i=1}^{N} \exp\left(-y_i \left(\sum_{j=1}^{k} \alpha_j t_j(x_i) + \alpha_{k+1} t_{k+1}(x_i)\right)\right) = \\
&= \sum_{i=1}^{N} \exp\left(-y_i \left(\sum_{j=1}^{k} \alpha_j t_j(x_i)\right)\right) \exp\bigl(-y_i \alpha_{k+1} t_{k+1}(x_i)\bigr) = \\
&\qquad\qquad = \sum_{i=1}^{N} w_i \exp\bigl(-y_i \alpha_{k+1} t_{k+1}(x_i)\bigr).
\end{aligned}
$$

$$
w_i = \exp\left(-y_i \left(\sum_{j=1}^{k} \alpha_j t_j(x_i)\right)\right)
$$

Новый компонент

$$Q(\alpha_{k+1}, t_{k+1}) = \sum_{i=1}^{N} w_i \exp\bigl(-y_i \alpha_{k+1} t_{k+1}(x_i)\bigr)$$

Обучение t_{k+1}

- Взвешенные объекты: w_i
- Исходные метки: y_i
- Ошибка классификации: $w_i[y_i \neq t_{k+1}(x_i)]$

Новый компонент

$$Q(\alpha_{k+1}, t_{k+1}) = \sum_{i=1}^{N} w_i \exp\bigl(-y_i \alpha_{k+1} t_{k+1}(x_i)\bigr)$$

Обучение t_{k+1}

- Взвешенные объекты: w_i
- Исходные метки: y_i
- Ошибка классификации: $w_i[y_i \neq t_{k+1}(x_i)]$

Выбор α_{k+1} (без доказательства)

$$\varepsilon = \frac{\sum_{i=1}^{N} w_i[y_i \neq t_{k+1}(x_i)]}{\sum_{i=1}^{N} w_i}$$

$$\alpha = \ln\left((1-\varepsilon)/\varepsilon\right)$$

Особенности AdaBoost

Достоинства

- Высокая обобщающая способность

Особенности AdaBoost

Достоинства

- Высокая обобщающая способность

Недостатки

- Неустойчивость к шуму
  (экспоненциальная функция потерь)
- Плохая интерпретируемость

Источники

- James, Witten, Hastie, Tibshirani. Introduction to Statistical Learning. Глава 8.
- Bishop C. Pattern Recognition and Machine Learning. Глава 14.
- Воронцов К. Лекции по алгоритмическим композициям.