Лекция 5
Бустинг (продолжение)
Нейронные сети (начало)

Взвешенная классификация

Задача взвешенной классификации

Обучающая выборка:

- $x_i \in \mathbb{R}^n$ — объекты
- $y_i \in \{1, \dots, k\}$ — ответы
- $w_i \in (0, +\infty)$ — веса

Функция потерь алгоритма $a(x)$:

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

i-й объект имеет вес w_i $\approx$ i-й объект повторяется w_i раз

Взвешенная линейная классификация

Обычная линейная регрессия:

$$\|y - AX\|_2^2 \to \min_A$$

$$A^* = (X^T X)^{-1} X^T y$$

Взвешенная линейная классификация

Обычная линейная регрессия:

$$\|y - AX\|_2^2 \to \min_A$$

$$A^* = (X^T X)^{-1} X^T y$$

Взвешенный вариант:

$$W = \mathrm{diag}(w_1, \dots, w_N)$$

$$\|W(y - AX)\|_2^2 \to \min_A$$

$$A^* = (X^T W^{-1} X)^{-1} X^T W^{-1} y$$

Взвешенные решающие деревья

Взвешенные доли классов:

$$p_k = \frac{1}{|A|} \sum_{i \in A} w_i [y_i = k]$$

Энтропия:

$$H(A) = \sum_{i=1}^{k} p_i \log p_i,$$

Качество разбиения A на A_l и A_r:

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

AdaBoost

Композиция для классификации

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

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

$t_1(x), \dots, t_n(x)$ — набор простых классификаторов

Голосование по большинству

$$a(x) = \operatorname{sign}\left(t_1(x) + \ldots + t_n(x)\right)$$

Бэггинг, случайный лес и т.д.

Взвешенная композиция для классификации

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

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

$t_1(x), \ldots, t_n(x)$ — набор простых классификаторов
$\alpha_1, \ldots, \alpha_n$ — веса классификаторов

Взвешенное голосование по большинству

$$a(x) = \operatorname{sign}\left(\alpha_1 t_1(x) + \ldots + \alpha_n t_n(x)\right)$$

Взвешенная композиция для классификации

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

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

$t_1(x), \ldots, t_n(x)$ — набор простых классификаторов
$\alpha_1, \ldots, \alpha_n$ — веса классификаторов

Взвешенное голосование по большинству

$$a(x) = \operatorname{sign}\left(\alpha_1 t_1(x) + \ldots + \alpha_n t_n(x)\right)$$

Как строить $\alpha_i, t_i(x)$?

Приближенное решение

Идея

- Будем строить $\alpha_i, t_i(x)$ последовательно
- Каждый из них строим оптимально
- Веса пока одинаковые

Приближенное решение

Идея

- Будем строить $\alpha_i, t_i(x)$ последовательно
- Каждый из них строим оптимально
- Веса пока одинаковые

План алгоритма

- Обучим алгоритм t_1 на исходной выборке
- Найдем, где алгоритм ошибается
- Увеличим вес этих объектов
- Обучим алгоритм t_2 на взвешенной выборке
- Найдем, где алгоритм ошибается
- $\ldots$

Выбор весов объектов

Из каких соображений выбирать веса объектов для t_{k+1}?

- Текущий алгоритм: $t_1(x) + \ldots + t_k(x)$
- Добавка: $t_{k+1}(x)$

Минимизация функции потерь

$$Q = \sum_{i=1}^{N} [y_i \neq \text{sign}\big(t_1(x_i) + \ldots + t_k(x_i) + t_{k+1}(x_i)\big)]$$

$$Q \to \min_{t_{k+1}}$$

(дискретную функцию тяжело оптимизировать)

Аппроксимация функции потерь

$$Q = \sum_{i=1}^{N} L(y_i f(x_i))$$

$$f(x_i) = t_1(x_i) + \ldots + t_n(x_i)$$

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

Аппроксимация функции потерь

$$Q = \sum_{i=1}^{N} L(y_i f(x_i))$$

$$f(x_i) = t_1(x_i) + \ldots + t_n(x_i)$$

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

$$\tilde{L}(z) = e^{-z}$$

$$L(z) \leqslant \tilde{L}(z)$$

$$Q \leqslant \tilde{Q} = \sum_{i=1}^{N} \tilde{L}(y_i f(x_i))$$

Минимизация аппроксимации

- Текущий алгоритм: $t_1(x) + \ldots + t_k(x)$
- Добавка: $t_{k+1}(x)$

Минимизация функции потерь

$$\tilde{Q} = \sum_{i=1}^{N} e^{-y_i(t_1(x_i)+\ldots+t_k(x_i)+t_{k+1}(x_i))}$$

$$\tilde{Q} \to \min_{t_{k+1}}$$

Преобразование функции потерь

$$\tilde{Q} \to \min_{t_{k+1}}$$

$$\tilde{Q} = \sum_{i=1}^{N} \exp\big(-y_i(t_1(x_i) + \ldots + t_k(x_i) + t_{k+1}(x_i))\big) =$$

$$= \sum_{i=1}^{N} \exp\big(-y_i(t_1(x_i) + \ldots + t_k(x_i))\big) \exp\big(-y_i t_{k+1}(x_i)\big) =$$

$$= \sum_{i=1}^{N} w_i \exp\big(-y_i t_{k+1}(x_i)\big).$$

$$w_i = \exp\big(-y_i(t_1(x_i) + \ldots + t_k(x_i))\big)$$

Выбор новой компоненты

- Текущий алгоритм: $t_1(x) + \ldots + t_k(x)$
- Добавка: $t_{k+1}(x)$

Минимизация функции потерь

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

Выбор новой компоненты

- Текущий алгоритм: $t_1(x) + \ldots + t_k(x)$
- Добавка: $t_{k+1}(x)$

Минимизация функции потерь

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

$t_{k+1}(x) \in \{-1, +1\}$, поэтому

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

Веса алгоритмов

$$a(x) = \operatorname{sign}\left(\alpha_1 t_1(x) + \ldots + \alpha_n t_n(x)\right)$$

$\alpha_j \sim$ правильность классификатора $t_j(x)$.

Веса алгоритмов

$$a(x) = \operatorname{sign}\left(\alpha_1 t_1(x) + \ldots + \alpha_n t_n(x)\right)$$

$\alpha_j \sim$ правильность классификатора $t_j(x)$.

Выбор α_{k+1}:

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

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

(без доказательства)

Алгоритм AdaBoost

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

- Инициализация решающей функции

$$f(x) := 0, \quad w_i = 1/N \text{ для } i = 1, \ldots, N$$

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

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

  - Обучение решающего дерева t_k на L'
  - Вычисление α_k по величине ошибки t_k на L'
  - Обновление алгоритма и весов

$$f(x) := f(x) + \alpha_k t_k(x)$$

$$w_i = \exp(-y_i f(x_i))$$

- Итоговый классификатор: $a(x) = \operatorname{sign}(f(x))$

Градиентный бустинг

Постановка задачи

L — произвольная функция потерь (дифференцируемая)

$$Q(a) = \sum_{i=1}^{N} L(y_i f(x_i))$$

$f(x)$ — решающая функция

$$f(x) = \sum_{j=1}^{k} \alpha_j t_j(x)$$

Новый компонент

Добавляем в композицию $\alpha_{k+1}t_{k+1}$

$$L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \to \min_{\alpha_{k+1}, t_{k+1}}$$

Новый компонент

Добавляем в композицию $\alpha_{k+1} t_{k+1}$

$$L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \to \min_{\alpha_{k+1}, t_{k+1}}$$

Рассмотрим L как функцию от α_{k+1}

$$\lambda(\alpha_{k+1}) = L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i))$$

Новый компонент

Добавляем в композицию $\alpha_{k+1} t_{k+1}$

$$L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \to \min_{\alpha_{k+1}, t_{k+1}}$$

Рассмотрим L как функцию от α_{k+1}

$$\lambda(\alpha_{k+1}) = L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i))$$

Ряд Тейлора по α_{k+1} в нуле:

$$\lambda(\alpha_{k+1}) \approx \lambda(0) + \alpha_{k+1} \lambda'(0)$$

Новый компонент

Добавляем в композицию $\alpha_{k+1} t_{k+1}$

$$L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \to \min_{\alpha_{k+1}, t_{k+1}}$$

Рассмотрим L как функцию от α_{k+1}

$$\lambda(\alpha_{k+1}) = L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i))$$

Ряд Тейлора по α_{k+1} в нуле:

$$\lambda(\alpha_{k+1}) \approx \lambda(0) + \alpha_{k+1} \lambda'(0)$$

Возвращаемся к L:

$$\begin{aligned} L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \approx \\ \approx L(y_i f(x_i)) + \alpha_{k+1} L'(y_i f(x_i)) y_i t_{k+1}(x_i) \end{aligned}$$

Новый компонент

Добавляем в композицию $\alpha_{k+1} t_{k+1}$

$$L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \to \min_{\alpha_{k+1}, t_{k+1}}$$

$$\begin{aligned} L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \approx \\ \approx L(y_i f(x_i)) + \alpha_{k+1} L'(y_i f(x_i)) y_i t_{k+1}(x_i) = \\ = L(y_i f(x_i)) + \alpha_{k+1}(-L'(y_i f(x_i)))(-y_i t_{k+1}(x_i)). \end{aligned}$$

Новый компонент

Добавляем в композицию $\alpha_{k+1} t_{k+1}$

$$L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \to \min_{\alpha_{k+1}, t_{k+1}}$$

$$\begin{aligned} L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) &\approx \\ &\approx L(y_i f(x_i)) + \alpha_{k+1} L'(y_i f(x_i)) y_i t_{k+1}(x_i) = \\ &= L(y_i f(x_i)) + \alpha_{k+1}(-L'(y_i f(x_i)))(-y_i t_{k+1}(x_i)). \end{aligned}$$

$$w_i = (-L'(y_i f(x_i)))$$

$$L \to \min_{t_{k+1}} \quad \Rightarrow \quad w_i [y_i t_{k+1} < 0] \to \min_{t_{k+1}}$$

Новый компонент

Добавляем в композицию $\alpha_{k+1} t_{k+1}$

$$L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \to \min_{\alpha_{k+1}, t_{k+1}}$$

$$\begin{aligned} L(y_i f(x_i) &+ y_i \alpha_{k+1} t_{k+1}(x_i)) \approx \\ &\approx L(y_i f(x_i)) + \alpha_{k+1} L'(y_i f(x_i)) y_i t_{k+1}(x_i) = \\ &= L(y_i f(x_i)) + \alpha_{k+1}(-L'(y_i f(x_i)))(-y_i t_{k+1}(x_i)). \end{aligned}$$

$$w_i = (-L'(y_i f(x_i)))$$

$$L \to \min_{t_{k+1}} \quad \Rightarrow \quad w_i [y_i t_{k+1} < 0] \to \min_{t_{k+1}}$$

- **Минимизация по t_{k+1}**
  Решаем задачу на взвешенной выборке

Новый компонент

Добавляем в композицию $\alpha_{k+1} t_{k+1}$

$$L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \to \min_{\alpha_{k+1}, t_{k+1}}$$

$$\begin{aligned} L(y_i f(x_i) + y_i \alpha_{k+1} t_{k+1}(x_i)) \approx \\ &\approx L(y_i f(x_i)) + \alpha_{k+1} L'(y_i f(x_i)) y_i t_{k+1}(x_i) = \\ &= L(y_i f(x_i)) + \alpha_{k+1}(-L'(y_i f(x_i)))(-y_i t_{k+1}(x_i)). \end{aligned}$$

$$w_i = (-L'(y_i f(x_i)))$$

$$L \to \min_{t_{k+1}} \quad \Rightarrow \quad w_i [y_i t_{k+1} < 0] \to \min_{t_{k+1}}$$

- **Минимизация по** t_{k+1}
  Решаем задачу на взвешенной выборке
- **Минимизация по** α_{k+1}
  Линейный поиск

Нейронные сети

Модели переменной емкости

Модели фиксированной емкости

- Линейная регрессия
- SVM
- Метрические методы

Модели с переменной емкостью
(сложность можно подстраивать под задачу)

- Решающее дерево
- SVM с ядрами
- Метрические методы с ядрами
- Композиции

Композиции

Простая композиция

$$\alpha_1 f_1(x) + \alpha_2 f_2(x) + \ldots + \alpha_n f_n(x)$$

Композиции

Простая композиция

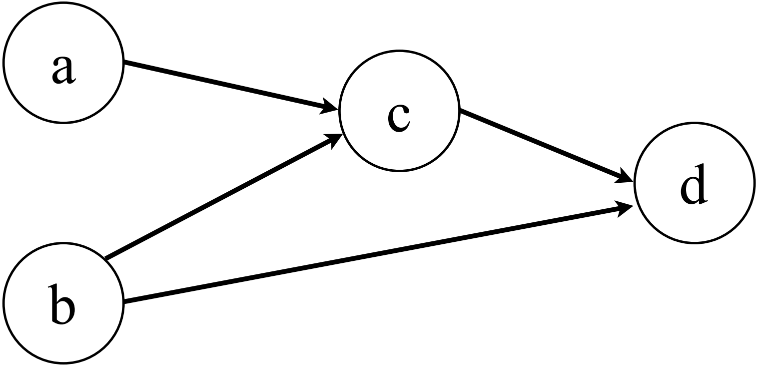
$$\alpha_1 f_1(x) + \alpha_2 f_2(x) + \ldots + \alpha_n f_n(x)$$

Более сложная композиция

$$\alpha_1 f_1(g_1(x), g_2(x)) + \alpha_2 f_2(g_3(x), g_4(x))$$

Композиции в виде графов

$$d(c(a, b), b)$$



Композиции линейных функций

Задача

- Сложные композиции
- Простые функции в узле
- Дифференцируемость

Композиции линейных функций

Задача

- Сложные композиции
- Простые функции в узле
- Дифференцируемость

Идея: ставить в узел линейную функцию.

$$f(x_1, \ldots, x_k) = \sum_{i=1}^{k} w_i x_i + w_0$$

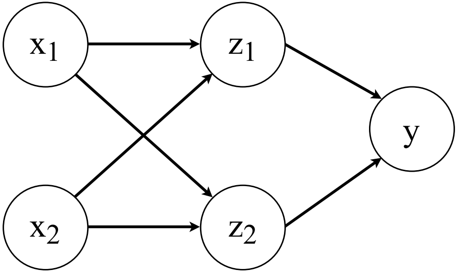
(в каждом узле свой набор w_i)

Композиции линейных функций

$$z_1 = w_{11}x_1 + w_{12}x_2$$

$$z_2 = w_{21}x_1 + w_{22}x_2$$

$$y = w_{31}z_1 + w_{32}z_2$$

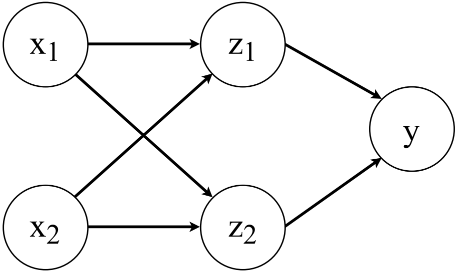


Композиции линейных функций

$$z_1 = w_{11}x_1 + w_{12}x_2$$

$$z_2 = w_{21}x_1 + w_{22}x_2$$

$$y = w_{31}z_1 + w_{32}z_2$$



Проблема: сохраняется линейность

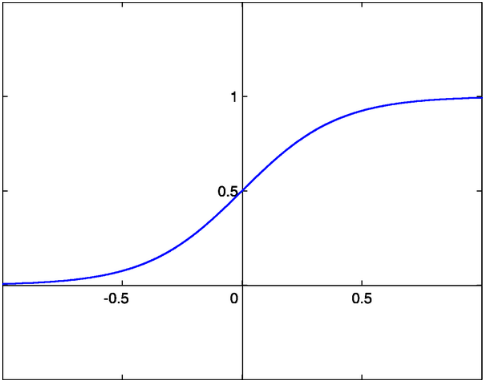
$$y = (w_{31}w_{11} + w_{32}w_{21})x_1 + (w_{31}w_{12}x_2 + w_{32}w_{22})x_2$$

Композиция нелинейных функций

$h : \mathbb{R} \to \mathbb{R}$ — некоторая нелинейная функция

Пример: сигмоида

$$h(x) = \frac{1}{1 + e^{-x}}$$



В каждом узле:

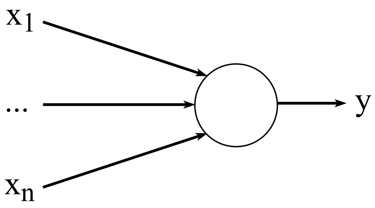
$$f(x_1, \ldots, x_k) = h\left(\sum_{i=1}^{k} w_i x_i + w_0\right)$$

Искусственная нейронная сеть

$$y = h\left(\sum_{i=1}^{k} w_i x_i + w_0\right)$$

$w_0, w_1, \ldots, w_k$ — параметры нейрона

$h(\cdot)$ — гиперпараметр нейрона

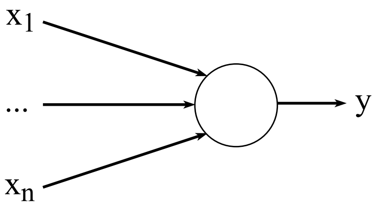


Искусственная нейронная сеть

$$y = h\left(\sum_{i=1}^{k} w_i x_i + w_0\right)$$

$w_0, w_1, \ldots, w_k$ — параметры нейрона

$h(\cdot)$ — гиперпараметр нейрона



Нейронная сеть = сеть (композиция) нейронов

Нейронная сеть — пример

$x_1, \ldots, x_D$ — входы
$x_0 = 1$ — фиктивный вход

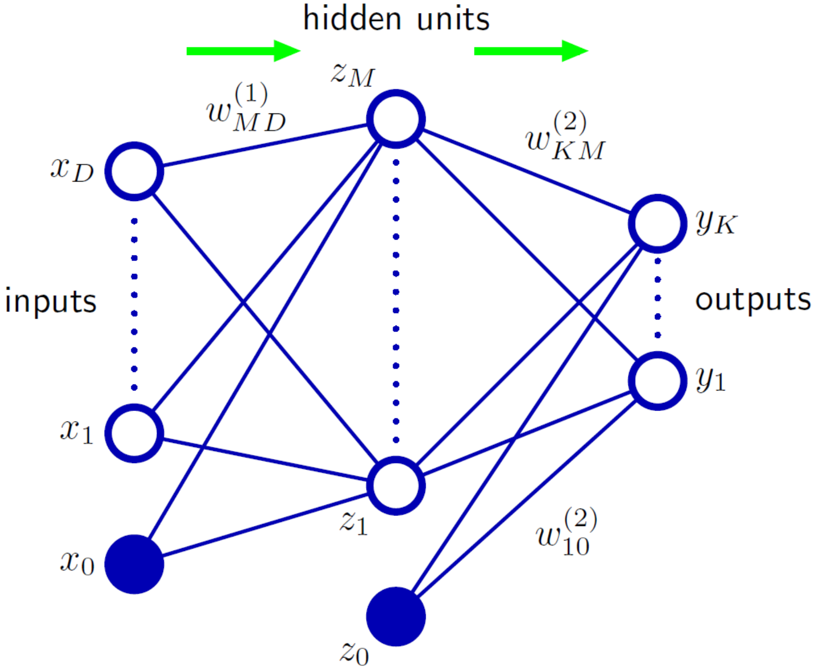
Скрытый слой

$$z_j = h\left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i\right), \quad j = 1, \ldots M$$

$$z_0 = 1 \text{ — фиктивный элемент}$$

Выходной слой

$$y_k = h\left(\sum_{j=0}^{M} w_{kj}^{(2)} z_i\right), \quad k = 1, \ldots K$$


hidden units
z_M
$w^{(1)}_{MD}$
$w^{(2)}_{KM}$
x_D
y_K
inputs
outputs
y_1
x_1
z_1
$w^{(2)}_{10}$
x_0
z_0

Аппроксимации с помощью нейронных сетей

Универсальные аппроксиматоры

Двухслойная нейронная сеть $\longrightarrow$ любая функция
(при условии произвольного числа нейронов)

Аппроксимации с помощью нейронных сетей

Универсальные аппроксиматоры

Двухслойная нейронная сеть $\longrightarrow$ любая функция
(при условии произвольного числа нейронов)

Дальнейшие вопросы

- Как обучать нейронные сети?
- Как добиться высокой обобщающей способности?

Источники

Бустинг

- 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.
  (Градиентный бустинг = AnyBoost)
- Википедия, Gradient boosting [ссылка]

Нейронные сети

- Bishop C. Pattern Recognition and Machine Learning. Глава 5.