Композиция методов

Леонович.Р.А

Санкт-Петербургский государственный университет
Прикладная математика и информатика
Кафедра Статистического Моделирования

СПб, 2021

План

- Дано $\mathbf{X} \in \mathbb{R}^{n \times p}$ — набор данных, $\mathbf{Y} \in \mathbb{R}^{n}$ — зависимые переменные, $X = (x_i, y_i)$.
- Возьмем l объектов с возвращениями — X_1
- Повторим N раз — $X_1, \ldots, X_N$
- Обучим по каждой выборке модель линейной регрессии и получим базовые алгоритмы $b_1(x), \ldots, b_N(x)$
- Предположим, что существует модель $y(x) = \sum \beta_i x_i + \epsilon_i$ и $p(x)$ — распределение $\mathbf{X}$.
- Ошибка регрессии: $\epsilon_j(x) = b_j(x) - y(x), \qquad j = 1, ..., N.$
- $\mathbb{E}_x \epsilon_j^2(x) = \mathbb{E}_x \left(b_j(x) - y(x)\right)^2$

Средняя ошибка построенных функций регрессии:

$E_1 = \dfrac{1}{N} \sum_{j=1}^{N} \mathbb{E}_x \epsilon_j^2(x)$

Пусть $\mathbb{E}_x \epsilon_j(x) = 0$ и $\mathbb{E}_x \epsilon_i(x) \epsilon_j(x) = 0$, $i \neq j$

и $a(x) = \frac{1}{N} \sum_{j=1}^{N} b_j(x)$

Тогда

$E_N = \mathbb{E}_x \left(\frac{1}{N} \sum_{j=1}^{N} b_j(x) - y(x) \right)^2 = \mathbb{E}_x \left(\frac{1}{N} \sum_{j=1}^{N} \epsilon_j(x) \right)^2 =$

$= \frac{1}{N^2} \mathbb{E}_x \left(\sum_{j=1}^{N} \epsilon_j^2(x) + \sum_{i \neq j} \epsilon_i(x) \epsilon_j(x) \right) = \frac{1}{N} E_1$

Пусть задана выборка $X = (x_i, y_i)_{i=1}^l$ с ответами $y_i \in \mathbb{R}$ и $\exists p(x, y)$
Рассмотрим $L(y, a) = (y - a(x))^2$ — функция потерь,
и $R(a) = \mathbb{E}_{x,y}\left[(y - a(x))^2\right] \int_{\mathbb{X}} \int_{\mathbb{Y}} p(x, y)(y - a(x))^2 dxdy$ — ее среднеквадратичный риск.

Метод обучения: $\mu : (\mathbb{X} \times \mathbb{Y})^l \to \mathbf{A}$

$$L(\mu) = \mathbb{E}_X \left[\mathbb{E}_{x,y}\left[(y - \mu(X)(x)))^2\right]\right] = \\ = \int_{(\mathbb{X}\times\mathbb{Y})^l} \int_{\mathbb{X}\times\mathbb{Y}} (y - \mu(X)(x))^2 \, p(x,y) \times \\ \times \prod_{i=1}^{l} p(x_i, y_i) dx dy dx_1 dy_1, \dots dx_l dy_l \tag{1}$$

Здесь, $\mathbb{E}_X[\cdot]$ берется по всем возможным выборкам $(x_1, y_1), \ldots, (x_l, y_l)$ из распределения $\prod_{i=1}^{l} p(x_i, y_i)$
$\mathbb{E}_{x,y} = \left[(y - \mu(X))^2\right] =$
$\mathbb{E}_{x,y}\left[(y - \mathbb{E}[y|x])^2\right] + \mathbb{E}_{x,y}\left[(\mathbb{E}[y|x] - \mu(X))^2\right]$ —
Среднеквадратичный риск на фиксированной выборке X
Подставим это в формулу (1)

Ошибка метода обучения (продолжение)

$$L(\mu) = \mathbb{E}_X\left[\underbrace{\mathbb{E}_{x,y}\left[(y-\mathbb{E}[y|x])^2\right]}_{\text{не зависит от X}} + \mathbb{E}_{x,y}\left[(\mathbb{E}[y|x]-\mu(X))^2\right]\right] =$$

$$= \mathbb{E}_{x,y}\left[(y-\mathbb{E}[y|x])^2\right] + \mathbb{E}_{x,y}\left[\mathbb{E}_X\left[(\mathbb{E}[y|x]-\mu(X))^2\right]\right] \tag{2}$$

Преобразовываем второе слагаемое:

$$\mathbb{E}_{x,y}\left[\mathbb{E}_X\left[(\mathbb{E}[y|x]-\mu(X))^2\right]\right] =$$

$$= \mathbb{E}_{x,y}\left[\mathbb{E}_X\left[(\mathbb{E}[y|x]-\mathbb{E}_X[\mu(X)]+\mathbb{E}_X[\mu(X)]-\mu(X))^2\right]\right] =$$

$$= \mathbb{E}_{x,y}\left[\mathbb{E}_X\left[\underbrace{(\mathbb{E}[y|x]-\mathbb{E}_X\mu(X))^2}_{\text{не зависит от X}}\right]\right] + \mathbb{E}_{x,y}\left[\mathbb{E}_X\left[(\mathbb{E}_X\mu(X)-\mu(X))^2\right]\right]$$

$$+ 2\mathbb{E}_{x,y}\left[\mathbb{E}_X\left[(\mathbb{E}[y|x]-\mathbb{E}_X[\mu(X)])(\mathbb{E}_X[\mu(X)]-\mu(X))\right]\right] \tag{3}$$

Ошибка метода обучения (продолжение)

Подставим (3) в (2).

$$L(\mu) = \underbrace{\mathbb{E}_{x,y}\left[(y - \mathbb{E}[y|x]^2)\right]}_{\text{шум}} + \tag{4}$$

$$+ \underbrace{\mathbb{E}_x\left[\mathbb{E}_X[\mu(X)] - \mathbb{E}[y|x]\right]}_{\text{смещение}} + \underbrace{\mathbb{E}_x\left[\mathbb{E}_X\left[(\mu(X) - \mathbb{E}_X[\mu(X)])^2\right]\right]}_{\text{разброс}} \tag{5}$$

Пример

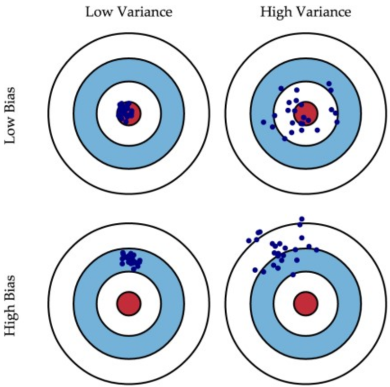

Рис.: Сдвиг и разброс разных моделей

Bagging

Возьмем некоторый метод обучения $\mu(X)$. Построим на его основе метод $\hat{\mu}(X)$, который гененирует случайную подвыборку $\hat{X}$ с помощью бутстрапа и подает ее на вход метода $\mu : \hat{\mu}(X) = \mu(\hat{X})$

$$a_N(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x) = \frac{1}{N}\sum_{n=1}^{N} \hat{\mu}(x) \tag{6}$$

Смещение и разброс для бэггинга

Из (5), смещение будет равно:

$$\mathbb{E}_{x,y}\left[\left(\mathbb{E}_X\left[\frac{1}{N}\sum_{b=1}^{N}\hat{\mu}(X)(x)\right]-\mathbb{E}[y|x]\right)^2\right]=$$
$$=\mathbb{E}_{x,y}\left[\left(\frac{1}{N}\sum_{b=1}^{N}\mathbb{E}_X\hat{\mu}(X)(x)\right)\right]=\mathbb{E}_{x,y}\left[\left(\mathbb{E}_X\left[\hat{\mu}(X)(x)\right]-\mathbb{E}[y|x]\right)^2\right] \tag{7}$$

Разброс:

$$\frac{1}{N}\mathbb{E}_{x,y}\left[\mathbb{E}_X\left[\left(\hat{\mu}(X)(x)-\mathbb{E}_X[\hat{\mu}(X)(x)]\right)^2\right]\right]+$$
$$+\frac{N(N-1)}{N^2}\mathbb{E}_{x,y}[\mathbb{E}_X[(\hat{\mu}(X)(x)-\mathbb{E}_X[\hat{\mu}(X)(x)])\times$$
$$\times(\hat{\mu}(X)(x)-\mathbb{E}_X[\hat{\mu}(X)(x)])]] \tag{8}$$

Out-of-Bag Error Estimation

$$OOB = \sum_{i=1}^{l} L\left(y_i, \frac{1}{\sum_{n=1}^{N}\left[x_i \notin X_n\right]} \sum_{n=1}^{N} [x_i \notin X_n] b_n(x_i)\right)$$

Алгоритм: Для $n = 1, \dots, N$

1. Сгенерировать выборку $\hat{X}_n$ с помощью бутстрапа.
2. Построить решающее дерево $b_n(x)$ по выборке $\hat{X}_n$.
   - дерево строится, пока в каждом листе не окажется не более n_{min} объектов
   - при каждом разбиении сначала выбирается m случайных признаков из p и оптимальное разделение ищется только среди них
3. Вернуть композицию $a_N(x) = \dfrac{1}{N} \sum_{n=1}^{N} b_n(x)$

В случайных лесах признак, по которому производится разбиение, выбирается из их случайного подмножества размера m.

Рекомендуется в задачах классификации брать $m = \sqrt{p}$, а в задачах регрессии — $m = p/3$

Сравнение бэггинга и случайного леса

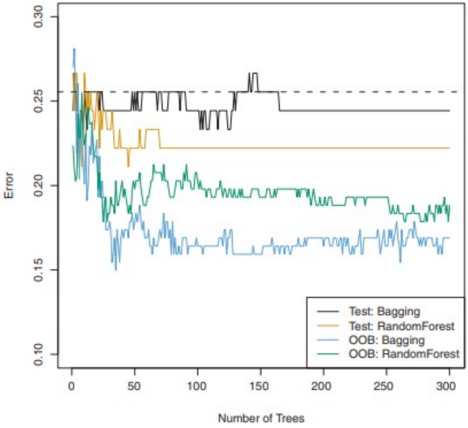

Рис.: График изменения ошибки моделей

Минимизация квадратичного функционала:
$\frac{1}{2}\sum_{i=1}^{l}(a(x_i) - y_i)^2 \to \min\limits_{a}$
Будем искать итоговый алгоритм в виде суммы базовых моделей $b_n(x)$:
$a_N(x) = \sum_{n=1}^{N} b_n(x)$, где базовые алгоритмы $b_n \in \mathbf{A}$.
Первый базовый алгоритм:
$b_1(x) := \underset{b \in A}{argmin} \frac{1}{2}\sum_{i=1}^{l}(b(x_i) - y_i)^2$.
Остатки на каждом объекте: $s_i^{(1)} = y_i - b_i(x)$
$b_2(x) := \underset{b \in A}{argmin} \frac{1}{2}\sum_{i=1}^{l}(b(x_i) - s^{(1)})^2$

Таким образом, каждый следующий алгоритм тоже будем настраивать на остатки предыдущих:

$s_i^{(N)} = y_i - \sum_{n=1} N - 1 b_n(x_i) = y_i - a_{N-1}(x_i)$,

$i = 1, \dots, l$

$b_N(x) := \underset{b \in A}{argmin} \frac{1}{2} \sum_{i=1}^{l} (b(x_i) - s_i^{(N)})^2$

Также, остатки могут быть найдены как антиградиент функции потерь по ответу модели, посчитанный в точке ответа уже построенной композиции:

$s_i^{(N)} = y_i - a_{N-1}(x_i) = -\frac{\mathrm{d}p}{\mathrm{d}z} \frac{1}{2} (z - y_i)^2 \Big|_{z = a_{N-1}(x_i)}$

Пусть дана некоторая дифференцируемая функция потерь $L(y, z)$.

Будем строить взвешенную сумму базовых алгоритмов:

$a_N(x) = \sum_{n=0}^{N} \gamma_n b_n(x)$

Примеры выбора алгоритма $b_0(x)$:

1. Нулевой: $b_0(x) = 0$.
2. Возвращающий самый популярный класс (в задачах классификации):

   $b_0(x) = \underset{y \in Y}{argmax} \sum_{i=1}^{l} [y_i = y]$
3. Возвращающий средний ответ (в задачах регрессии):

   $b_0(x) = \frac{1}{l} \sum_{i=1} l y_i$

Допустим, мы построили композицию $a_{N-1}(x)$ из $N-1$ алгоритма, и хотим выбрать следующий абзовый алгоритм $b_N(x)$ так, чтобы как можно сильнее уменьшить ошибку:
$\sum_{i=1}^{l} L\left(y_i, a_{N-1}(x_i) + \gamma_N b_N(x_i)\right) \to \underset{b_N, \gamma_N}{min}$
Какие числа $s_1, \ldots, s_l$ надо выбрать для решения следующей задачи:
$\sum_{i=1}^{l} L\left(y_i, a_{N-1}(x_i) + s_i\right) \to \underset{s_1, \ldots, s_l}{min}$

- $s_i = y_i - a_{N-1}(x_i)$ **?**
- $s_i = - \left. \frac{\partial L}{\partial z} \right|_{z=a_{N-1}(x_i)}$
  В этом случае сдвиг s_i будет противоположен производной функции потерь в точке $z = a_{N-1}(x_i)$:

Заметим, что вектор сдвигов $s = s_1, \ldots, s_l$ совпадает с антиградиентом:

$$\left(- \left. \frac{\partial L}{\partial z} \right|_{z=a_{N-1}(x_i)} \right)_{i=1}^{l} = - \bigtriangledown_z \sum_{i=1}^{l} L(y_i, z_i)|_{z=a_{N-1}(x_i)}$$

По данным значениям в конечном числе точек необходимо построить функцию, заданную на всем пространстве объектов.

$b_N(x) = \underset{b \in A}{argmin} \sum_{i=1}^{l} (b(x_i) - s_i)^2$ — среднеквадратичная ошибка

$\gamma_N = \underset{\gamma \in R}{argmin} \sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + \gamma b_N(x_i))$ — подбор коэффициента

Регуляризация

- Сокращение шага
  $a_N(x) = a_{N-1}(x) + \eta\gamma_N b_N(x)$, где $\eta \in (0, 1]$ — темп обучения.
- Стохастический градиентный бустинг

- Регрессия
  - Квадратичная $\sum_{i=1}^{l}(a(x_i) - y_i)^2 \to \min\limits_{a}$
  - Модуль отклонения $L(y, z) = |y - z|$, для которого антиградиент вычисляется по формуле $s_i^{(N)} = -sign(a_{N-1}(x_i) - y_i)$
- Классификация

  $L(y, z) = log(1 + exp(-yz))$

  Задача поиска базового алгоритма с ней принимает вид

  $b_N = \underset{b \in A}{argmin} \sum_{i=1}^{l}(b(x_i) - \frac{y_i}{1+exp(y_i a_{N-1}(x_i))})^2$

Особенность логистической функции

Ошибка на N-ой итерации:

$Q(a_N) = \sum_{i=1}^{l} log(1 + exp(-y_i a_N(x_i))) =$
$\sum_{i=1}^{l} log(1 + exp(-y_i a_{N-1}(x_i)) exp(-y_i \gamma_n b_N(x_i)))$

Таким образом, величина $w_i^{(N)} = exp(-y_i a_{N-1} x(i))$ может случить мерой важности объекта x_i на N-й итерации градиентного бустинга.

$b_n(x) = \sum_{j=1}^{J_n} b_{nj}[x \in R_j]$,
где $j = 1, \dots, J_n$ — индексы листьев, R_j — соответствующие области разбиения, b_{nj} — значения в листьях.

В N-й итерации бустинга композиция обновляется как
$a_N(x) = a_{N-1}(x) + \gamma_N \sum_{j=1}^{J_N} b_{Nj}[x \in R_j]$

Можно улучшить качество композиции:
$\sum_{i=1}^{l} L\left(y_i, a_{N-1}(x_i) + \sum_{j=1}^{J_N} \gamma_{Nj}[x \in R_j]\right) \to \min\limits_{\{\gamma_{Nj}\}_{j=1}^{J_N}}$

Так как области разбиения R_j не пересекаются, данная задача распадается на J_N независимых подзадач:

$$\gamma_{Nj} = \underset{\gamma}{argmin} \sum_{x_i \in R_j} L(y_i, a_{N-1}(x_i) + \gamma), \qquad j = 1, \dots, J_N$$

Логистическая функция потерь.

$$F_j^{(N)}(\gamma) = \sum_{x_i \in R_j} log(1 + exp(-y_i(a_{N-1}(x_i) + \gamma))) \to \underset{\gamma}{min}.$$

Смещение и разброс

В случайных лесах:

- Используются глубокие деревья, поскольку от базовых алгоритмов требуется низкое смещение.
- Разброс устраняется за счёт усреднения ответов различных деревьев.

В бустинге:

- Каждый следующий алгоритм понижает ошибку композиции.
- Переобучение при большом количестве базовых моделей.
- Можно понизить смещение моделей, а разброс либо останется таким же, либо увеличится.
- Используются неглубокие решающие деревья.

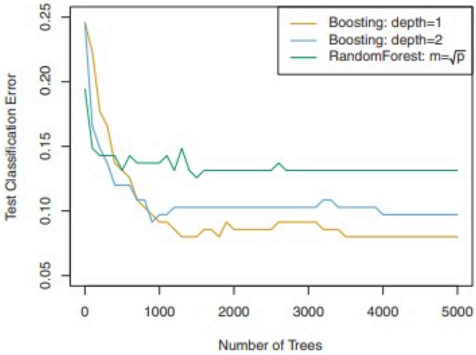

Рис.: График изменения ошибки моделей

Взвешивание объектов

AdaBoost: $L(y, z) = exp(-yz)$

$L(a, X) = \sum_{i=1}^{l} exp\left(-y_i \sum_{n=1}^{N} \gamma_n b_n(x_i)\right)$

Компоненты ее антиградиента после $N-1$ итерации:

$$s_i = -\frac{\partial L(y_i, z)}{\partial z}\Bigg|_z = a_{N-1}(x_i) = y_i \underbrace{exp\left(-y_i \sum_{n=1}^{N-1} \gamma_n b_n(x_i)\right)}_{w_i}$$

Рассмотрим теперь логистическую функцию потерь, которая также может использоваться в задачах классификации:
$L(a, X^l) = \sum_{i=1}^{l} log(1 + exp(-y_i a(x_i)))$

Ее антиградиент после $N - 1$ шага:
$$s_i = y_i \underbrace{\frac{1}{1 + exp(y_i a_{N-1}(x_i))}}_{w_i^{(N)}}$$