Машинное обучение ФИВТ

DS-поток

Лекция пятая

Бустинг

Ансамбли

Подходы к построению композиций:

- Беггинг
- Случайный лес
- Бустинг
- Блендинг
- Стекинг
- StackNet

Сегодня о бустинге!

Бустинг

Бустинг в задаче регрессии

Общий случай градиентного бустинга

Регуляризация

Вывод для разных функций потерь

Градиентный бустинг над деревьями

Смещение и разброс

Взвешивание объектов для задачи классификации

Частные случаи для задачи классификации

Методы оптимизации второго порядка

Бустинг в задаче регрессии

Пусть $(x_1, Y_1), \ldots, (x_n, Y_n)$ — обучающая выборка.

$\widehat{y}$ — модель.

Рассмотрим задачу минимизации MSE:

$$Q(Y, \widehat{y}) = \frac{1}{2}\sum_{i=1}^{n}(\widehat{y}(x_i) - Y_i)^2 \longrightarrow \min_{\widehat{y}}$$

Ищем итоговую модель в виде суммы *базовых моделей* $b_t(x)$,

где b_t принадлежат некоторому семейству $\mathscr{F}$:

$$\widehat{y}_T(x) = \sum_{t=1}^{T} b_t(x)$$

Замечание: под $\widehat{y}_T$ будем обозначать модель, состоящую из T базовых моделей.

Бустинг в задаче регрессии

Сначала построим первую базовую модель:

$$b_1 = \arg\min_{b \in \mathscr{F}} \frac{1}{2} \sum_{i=1}^{n} (b(x_i) - Y_i)^2$$

Обучив первую базовую модель, можем посчитать остатки :

$$e_i^1 = Y_i - b_1(x_i)$$

Построим вторую базовую модель так, чтобы ее ответы как можно лучше приближали остатки e_i^1 :

$$b_2 = \arg\min_{b \in \mathscr{F}} \frac{1}{2} \sum_{i=1}^{n} \left(b(x_i) - e_i^1\right)^2$$

Каждую следующую модель тоже будем обучать на остатки предыдущих:

$$e_i^{t-1} = Y_i - \sum_{k=1}^{t-1} b_k(x_i) = Y_i - \widehat{y}_{t-1}(x_i)$$

$$b_t(x) = \arg\min_{b \in \mathscr{F}} \frac{1}{2} \sum_{i=1}^{n} \left(b(x_i) - e_i^{t-1}\right)^2$$

Бустинг в задаче регрессии

Задача построения следующей модели:

$$e_i^{t-1} = Y_i - \sum_{k=1}^{t-1} b_k(x_i) = Y_i - \widehat{y}_{t-1}(x_i)$$

$$b_t(x) = \arg\min_{b \in \mathscr{F}} \frac{1}{2} \sum_{i=1}^{n} \left(b(x_i) - e_i^{t-1}\right)^2$$

Таким образом:

- b_1 обучается на выборке $\{(x_i, Y_i)\}$.
- b_2 обучается на выборке $\{(x_i, e_i^1)\}$.
- b_t обучается на выборке $\{(x_i, e_i^t)\}$.

Бустинг в задаче регрессии

Посчитаем производную функции потерь по ответу модели:

$$\frac{\partial}{\partial z}\mathcal{L}(Y_i, z)\Big|_{z=\widehat{y}_{t-\mathbf{1}}(x_i)} = \frac{\partial}{\partial z}\frac{1}{2}(Y_i - z)^2\Big|_{z=\widehat{y}_{t-\mathbf{1}}(x_i)} = \widehat{y}_{t-1}(x_i) - Y_i$$

Заметим, что производная со знаком минус равна e_i^t:

$$e_i^t = -\frac{\partial}{\partial z}\mathcal{L}(Y_i, z)\Big|_{z=\widehat{y}_{t-\mathbf{1}}(x_i)} = Y_i - \widehat{y}_{t-1}(x_i)$$

$$\Rightarrow e^t = (e_1^t, ..., e_n^t) = -\nabla\mathcal{L}(Y, z)|_{z=\widehat{y}_{t-\mathbf{1}}(x)}$$

Напоминание:

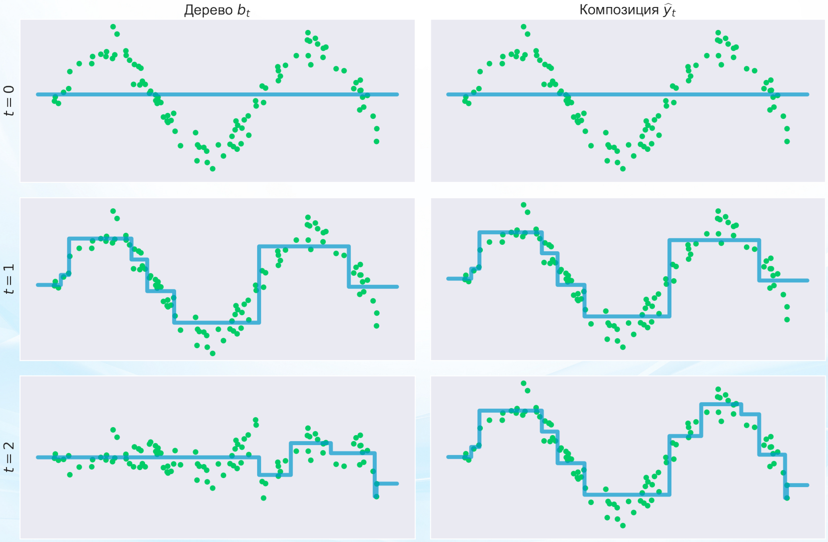
$\nabla F(x_0)$ — направление наискорейшего роста функции F в точке x_0.

$-\nabla F(x_0)$ — направление наискорейшего убывания F в точке x_0.

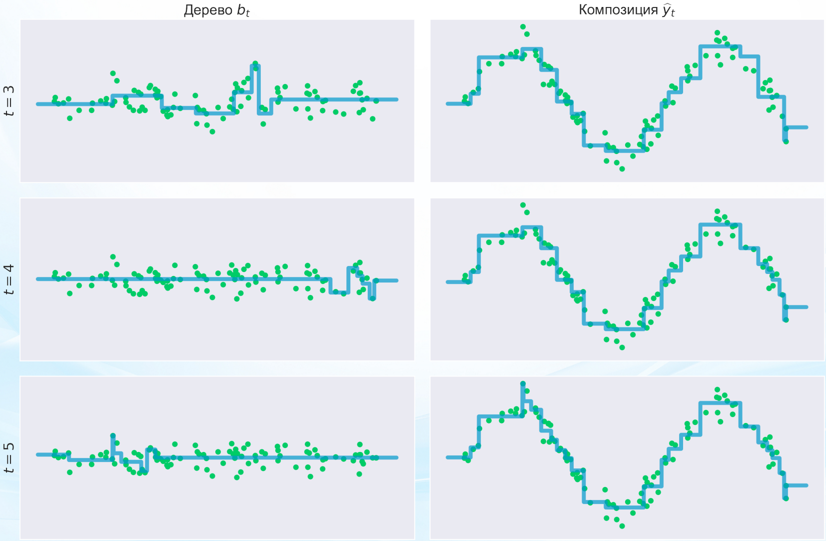
$\Rightarrow$ Модель шагает в сторону антиградиента,
т.е. направления наискорейшего спуска.

$\Rightarrow$ Выбирается такая базовая модель,
которая как можно сильнее уменьшит ошибку композиции.

Пример



Пример



Бустинг

Бустинг в задаче регрессии

Общий случай градиентного бустинга

Регуляризация

Вывод для разных функций потерь

Градиентный бустинг над деревьями

Смещение и разброс

Взвешивание объектов для задачи классификации

Частные случаи для задачи классификации

Методы оптимизации второго порядка

Градиентный бустинг

Пусть дана некоторая дифференцируемая функция потерь $\mathcal{L}(y, z)$.

Будем строить взвешенную сумму базовых моделей:

$$\widehat{y}_T(x) = \sum_{t=1}^{T} \gamma_t b_t(x)$$

В композиции имеется начальная модель $b_0(x)$.

Обычно берут $\gamma_0 = 1$.

Базовую модель выбирают очень простой:

- нулевой $b_0(x) = 0$
- возвращающую самый популярный класс (для классификации):

$$b_0(x) = \arg\max_{y \in \mathscr{Y}} \sum_{i=1}^{n} I\{Y_i = y\}$$

- возвращающую средний ответ (для регрессии):

$$b_0(x) = \frac{1}{n} \sum_{i=1}^{n} Y_i$$

Построение новой базовой модели

Пусть построили композицию $\widehat{y}_{t-1}$ из $t-1$ моделей.

Хотим выбрать b_t так, чтобы как можно больше уменьшить ошибку:

$$\sum_{i=1}^{n} \mathcal{L}(Y_i,\ \widehat{y}_{t-1}(x_i) + \gamma_t b_t(x_i)) \longrightarrow \min_{b_t, \gamma_t}$$

Поймем какие значения нужно принять модели b_t на трейне.

Т.е. хотим понять какие числа $s_1, ..., s_n$ в идеале нужно выбрать

для решения задачи

$$\sum_{i=1}^{n} \mathcal{L}(Y_i, \widehat{y}_{t-1}(x_i) + s_i) \to \min_{s_1, \dots, s_n}$$

Замечание:

Понятно, что можно выбрать $s_i = Y_i - \widehat{y}_{t-1}(x_i)$, но такой подход никак не учитывает особенностей функции потерь $\mathcal{L}(y, z)$
и требует лишь точного совпадения предсказаний и истинных ответов.
Также нужно учитывать, что такой модели в $\mathscr{F}$ может не быть.

Построение новой базовой модели

$$Q(s) = Q(s_1, \dots, s_n) = \sum_{i=1}^{n} \mathcal{L}(Y_i, \widehat{y}_{t-1}(x_i) + s_i) \to \min_{s_1, \dots, s_n}$$

Для нахождения минимума Q рассмотрим направление наискорейшего убывания Q в точке $\overline{0}$.

$\overline{0}$ соответствует отсутствию модели b_t.

$$-\nabla_s Q\big|_{s=\overline{0}} = -\nabla_z \sum_{i=1}^{n} \mathcal{L}(Y_i, z_i)\Big|_{z_i=\widehat{y}_{t-1}(x_i)}$$

Тогда возьмем вектор сдвигов равный антиградиенту:

$$s = -\nabla_z \sum_{i=1}^{n} \mathcal{L}(Y_i, z_i)\Big|_{z_i=\widehat{y}_{t-1}(x_i)} = \left(- \left.\frac{\partial \mathcal{L}}{\partial z}\right|_{z=\widehat{y}_{t-1}(x_i)} \right)_{i=1}^{n}$$

$$\Rightarrow s_i = - \left.\frac{\partial \mathcal{L}}{\partial z}\right|_{z=\widehat{y}_{t-1}(x_i)}$$

То есть хотим сдвинуться в сторону наискорейшего убывания ошибки на обучающей выборке.

Построение новой базовой модели

$\Rightarrow$ Поняли какие значения новая модель должна в идеале принимать на обучающей выборке : $s_1, \dots, s_n$.

Будем обучать новую модель на полученные сдвиги.

Один из самых простых функционалов для обучения — среднеквадратичная ошибка, воспользуемся ей для поиска b_t:

$$b_t(x) = \underset{b \in \mathscr{F}}{\arg\min} \sum_{i=1}^{n} (b(x_i) - s_i)^2$$

Заметим, что в случае регресии $\widehat{y}$ возвращает действительные числа,
а в случае классификации — вероятности классов.
И то, и другое можно настраивать по MSE.

Замечание: Здесь мы оптимизируем с/к функцию потерь независимо от функционала исходной задачи — вся информация о функции потерь $\mathcal{L}$ находится в векторе $s = (s_1, \dots, s_n)$.

Можно использовать и другие функционалы, но с/к ошибки обычно достаточно.
Еще одна причина : модель должна приблизить направление наискорейшего убывания, совпадение направлений можно оценивать через косинус угла между ними, который напрямую связан с с/к ошибкой.

Аналогия с градиентным спуском

Итак, для выбора b_t решаем задачу:

$$Q(s) = Q(s_1, \ldots, s_n) = \sum_{i=1}^{n} \mathcal{L}(Y_i, \widehat{y}_{T-1}(x_i) + s_i) \longrightarrow \min_{s_1, \ldots, s_n}$$

$$s_i = -\left.\frac{\partial \mathcal{L}}{\partial z}\right|_{z=\widehat{y}_{t-1}(x_i)}$$

$$b_t(x) = \arg\min_{b \in \mathscr{F}} \sum_{i=1}^{n} (b(x_i) - s_i)^2$$

Реализуется приближение градиентного спуска

по вектору предсказаний модели $z = (z_i)_{i=1\ldots n} = \widehat{y}(x_i)_{i=1\ldots n}$

для минимизации функционала $\sum\limits_{i=1}^{n} \mathcal{L}(Y_i, z_i)$

Если бы могли получить $b_t \in \mathscr{F}$ т.ч. $b_t(x_i) = s_i$,

то это был бы в точности градиентный спуск.

Замечание: Здесь речь о градиентном спуске в n-мерном пространстве предсказаний модели на объектах обучающей выборки.

Выбор коэффициента при базовой модели

Теперь можно подобрать коэффициент при b_t

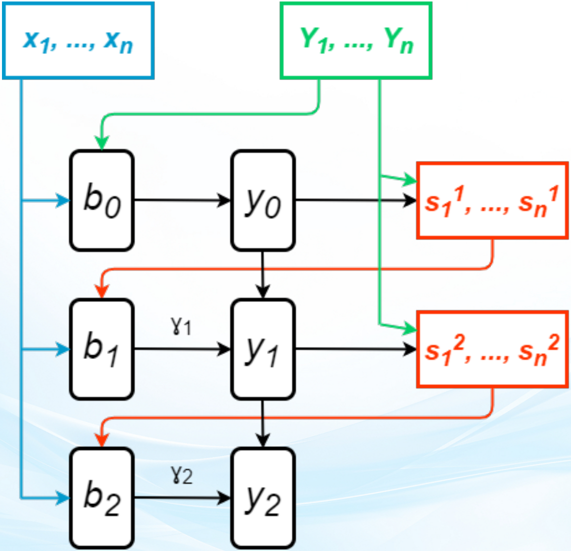
по аналогии с градиентным спуском:

$$\gamma_t = \arg\min_{\gamma \in \mathbb{R}} \sum_{i=1}^{n} \mathcal{L}(Y_i, \widehat{y}_{t-1}(x_i) + \gamma b_t(x_i))$$

В итоге:

- Каждый шаг делается вдоль направления, задаваемого некоторой базовой моделью.
- Базовая модель выбирается так, чтобы как можно лучше приближать антиградиент ошибки на обучающей выборке.

Схема градиентного бустинга



Бустинг

Бустинг в задаче регрессии

Общий случай градиентного бустинга

Регуляризация

Вывод для разных функций потерь

Градиентный бустинг над деревьями

Смещение и разброс

Взвешивание объектов для задачи классификации

Частные случаи для задачи классификации

Методы оптимизации второго порядка

Регуляризация

Проблемы:

- Если базовые модели очень простые,

  то они плохо приближают вектор антиградиента.

  $\Rightarrow$ Шаг выполняется вдоль направления, сильно отличного от направления антиградиента.

  $\Rightarrow$ Градиентный бустинг может свестись к случайному блужданию в пространстве.

- Если базовые модели сложные, то они способны за несколько шагов бустинга идеально подогнаться под обучающую выборку $\Rightarrow$ композиция переобучится.

1. Сокращение шага

Вместо перехода в оптимальную точку в направлении антиградиента делаем укороченный шаг:

$$\widehat{y}_t(x) = \widehat{y}_{t-1}(x) + \eta \cdot \gamma_t b_t(x)$$

где $\eta \in (0, 1]$ — темп обучения (learning rate).

Смысл:

Понижаем доверие к направлению, предсказан. базовой моделью.

Обычно, чем меньше η, тем лучше качество итоговой композиции, но требуется больше итераций для сходимости.

2. Контроль количества итераций

С увеличением кол-ва итераций:

- Ошибка на обучающей выборке стремится к 0.
- Ошибка на контроле обычно начинает увеличиваться после определенной итерации.

Идея: Можно контролировать число итераций бустинга.

Зададим гиперпараметр T — максимальное число итераций.

Оптимальное число итераций можно выбирать по валидационной выборке или с помощью кросс-валидации.

3. Стохастический градиентный бустинг

Модель b_t обучается не по всей выборке X,

а лишь по ее случайному подмножеству $X_t^* \subset X$.

Подмножество X_t^* выбирается для каждой итерации заново.

Плюсы:

- Понижается уровень шума в обучении
- Повышается эффективность вычислений
- Повышается обощающая способность

Рекомендация :

Брать подвыборки, размер которых вдвое меньше исходной выборки.

Бустинг

Функции потерь : Регрессия

- MSE : $\mathcal{L}(Y_i, \widehat{y}(x_i)) = \frac{1}{2}\left(\widehat{y}(x_i) - Y_i\right)^2$

$$s_i^t = -\frac{\partial}{\partial z}\frac{1}{2}(z - Y_i)^2\Big|_{z=\widehat{y}_{t-\mathbf{1}}(x_i)} = Y_i - \widehat{y}_{t-1}(x_i)$$

Модель b_t обучается на выборке $\{(x_i,\, Y_i - \widehat{y}_{t-1}(x_i))\}$.

- MAE : $\mathcal{L}(Y_i, \widehat{y}(x_i)) = |\widehat{y}(x_i) - Y_i|$

$$s_i^t = -\frac{\partial}{\partial z}|z - Y_i|\Big|_{z=\widehat{y}_{t-\mathbf{1}}(x_i)} = -sign(\widehat{y}_{t-1}(x_i) - Y_i)$$

Модель b_t обучается на выборке $\{(x_i, -sign(\widehat{y}_{t-1}(x_i) - Y_i))\}$.

Функции потерь : Классификация

Рассмотрим задачу бинарной классификации: $Y_i \in \{-1, +1\}$

Тогда решающее правило принимает вид $f(x) = sign(\widehat{y}(x))$.

Экспоненциальная функция потерь:

$$\mathcal{L}(Y_i, \widehat{y}(x_i)) = \exp(-Y_i \cdot \widehat{y}(x_i))$$

Найдем компоненты ее антиградиента после $(T-1)$-й итерации:

$$s_i = -\left.\frac{\partial \mathcal{L}(Y_i, z)}{\partial z}\right|_{z=\widehat{y}_{T-1}(x_i)} = -\left.\frac{\partial}{\partial z}\exp(-Y_i \cdot z)\right|_{z=\widehat{y}_{T-1}(x_i)} = \\ = Y_i \cdot \exp(-Y_i \cdot \widehat{y}_{T-1}(x_i))$$

Задача поиска базовой модели принимает вид:

$$b_T = \arg\min_{b \in \mathscr{F}} \sum_{i=1}^{n} \Big(b(x_i) - Y_i \cdot \exp\left(Y_i \cdot \widehat{y}_{T-1}(x_i)\right)\Big)^2$$

Бустинг

Градиентный бустинг над деревьями

Решающее дерево разбивает все пространство на непересекающиеся области, в которых его ответ равен константе:

$$b_T(x) = \sum_{j=1}^{J_T} b_{Tj} \cdot I\{x \in R_j\}$$

где $j = 1, \ldots, J_T$ - индексы листьев,

R_j - соответствующие области разбиения : $\bigcup_{j=1}^{J_T} R_j = \mathscr{X}$

b_{Tj} - значения в листьях.

На T-й итерации композиция обновляется как

$$\widehat{y}_T(x) = \widehat{y}_{T-1}(x) + \gamma_T \sum_{j=1}^{J_T} b_{Tj} I\{x \in R_j\} = \widehat{y}_{T-1}(x) + \sum_{j=1}^{J_T} \gamma_T b_{Tj}\, I\{x \in R_j\}$$

Все R_j не пересекаются : $R_{j_1} \cap R_{j_2} = \varnothing$.

$\Rightarrow$ Добавление в композицию дерева с J_T листьями равносильно добавлению J_T базовых моделей, представляющих собой предикаты вида $I\{x \in R_j\}$.

Перенастройка в листьях

$$\widehat{y}_T(x) = \widehat{y}_{T-1}(x) + \sum_{j=1}^{J_T} \gamma_T b_{Tj}\, I\{x \in R_j\}$$

Если вместо общего γ_T будет свой γ_{Tj} при каждом предикате, то можем его подобрать так, чтобы повысить качество композиции.

- Обучим дерево $b_T \Rightarrow$ структура дерева задана.
- Сделаем перенастройку в листьях обученнного дерева.

Тогда потребность в b_{Tj} отпадает:

$$\sum_{i=1}^{n} \mathcal{L}\left(Y_i,\ \ \widehat{y}_{T-1}(x_i) + \sum_{j=1}^{J_T} \gamma_{Tj} \cdot I\{x \in R_j\} \right) \longrightarrow \min_{\{\gamma_{Tj}\}_{j=1}^{J_T}}$$

Т.к. области разбиения R_j не пересекаются, задача распадается на J_T независимых подзадач:

$$\gamma_{Tj} = \arg\min_{\gamma} \sum_{x_i \in R_j} \mathcal{L}(y_i, \widehat{y}_{T-1}(x_i) + \gamma), \qquad j = 1, \ldots, J_T$$

В некоторых случаях оптимальные γ_{Tj} можно найти аналитически - например, для квадратичной и абсолютной ошибки.

Перенастройка в листья

Рассмотрим экспоненциальную функцию потерь.

$$\sum_{i=1}^{n} e^{-Y_i \cdot \widehat{y}(x_i)} = \sum_{i=1}^{n} \exp\Big(- Y_i \cdot \big[\widehat{y}_{T-1}(x_i) + \gamma_T b_T(x_i)\big]\Big) \longrightarrow \min_{b_T}$$

В этом случае нужно решить задачу

$$F_j^T(\gamma) = \sum_{x_i \in R_j} \exp\Big(- Y_i \cdot \big[\widehat{y}_{T-1}(x_i) + \gamma\big]\Big) \longrightarrow \min_{\gamma}$$

Аналитической записи нет, только итерационные методы.

На практике обычно не нужно искать точное решение — достаточно сделать 1 шаг метода Ньютона из нач. приближения $\gamma_{Tj} = 0$.

Можно показать, что в этом случае

$$\gamma_{Tj} = -\frac{\partial F_j^T(0)}{\partial \gamma} \Bigg/ \frac{\partial^2 F_j^T(0)}{\partial \gamma^2} = -\sum_{x_i \in R_j} s_i^T \Bigg/ \sum_{x_i \in R_j} Y_i \cdot s_i^T$$

Бустинг

Смещение и разброс

Какие деревья используются в случайных лесах?

Глубокие

Почему?
Базовые модели должны иметь низкое смещение,
разброс устраняется за счёт усреднения ответов.

Какие деревья используются в бустинге?

Неглубокие

Почему?
Бустинг понижает смещение моделей,
а разброс либо останется таким же, либо увеличится.
$\Rightarrow$ Нужны модели с большим смещением и низким разбросом.

Обычно используются неглубокие решающие деревья (3-6 уровней).

Бустинг

Бустинг в задаче регрессии

Общий случай градиентного бустинга

Регуляризация

Вывод для разных функций потерь

Градиентный бустинг над деревьями

Смещение и разброс

Взвешивание объектов для задачи классификации

Частные случаи для задачи классификации

Методы оптимизации второго порядка

Отступ на объекте

Решаем задачу бинарной классификации : $Y_i \in \{-1, +1\}$

Решающее правило:

$$f(x) = sign(\widehat{y}(x))$$

Введем понятие отступа на объекте:

$$M_i = Y_i \cdot \widehat{y}(x_i)$$

Свойства:

- $M_i > 0 \Leftrightarrow$ объект x_i классифицируется верно.
- $M_i < 0 \Leftrightarrow$ объект x_i классифицируется неверно.
- Чем больше $|M_i|$, тем больше классификатор уверен в своем ответе.

Взвешивание объектов для задачи классификации

Экспоненциальная функция потерь.

Заметим, что ошибка на T-ой итерации выражается как:

$$Q(Y, \widehat{y}_T) = \sum_{i=1}^{n} \exp\Big(-Y_i \cdot \widehat{y}_T(x_i)\Big) = \sum_{i=1}^{n} \exp\left(-Y_i \cdot \left[\widehat{y}_{T-1}(x_i) + \gamma_T b_T(x_i)\right]\right)$$

$$= \sum_{i=1}^{n} \exp\Big(-Y_i\, \widehat{y}_{T-1}(x_i)\Big) \cdot \exp\Big(-Y_i\, \gamma_T b_T(x_i)\Big)$$

Если отступ $Y_i \cdot \widehat{y}_{T-1}(x_i)$ на i-ом объекте большой и положительный, то данный объект не вносит особого вклада в ошибку.

$\Rightarrow$ Его можно исключить из вычислений на текущей итерации.

Поэтому величина $\exp\big(-Y_i \cdot \widehat{y}_{T-1}(x_i)\big)$ может служить

мерой важности объекта x_i на T-ой итерации.

Взвешивание объектов для задачи классификации

Экспоненциальная функция потерь.

$$Q(Y, \widehat{y}_T) = \sum_{i=1}^{n} \exp\left(-Y_i \sum_{t=1}^{T} \gamma_t b_t(x_i)\right)$$

Компоненты ее антиградиента:

$$s_i^T = -\left.\frac{\partial \mathcal{L}(Y_i, z)}{\partial z}\right|_{z=\widehat{y}_{T-1}(x_i)} = Y_i \cdot \underbrace{\exp\left(-Y_i \sum_{t=1}^{T-1} \gamma_t b_t(x_i)\right)}_{w_i}$$

Сдвиг s_i^T равен ответу на объекте, умноженный на его вес.

Отступ на x_i : $M_i = Y_i \cdot \widehat{y}_{T-1}(x_i) = Y_i \sum\limits_{t=1}^{T-1} \gamma_t b_t(x_i)$

- Объект имеет большой положительный отступ $\to$ вес близок к 0.
- Отступ большой отрицательный $\to$ вес очень большой и не ограничен сверху. Обычно наблюдается на выбросах.

$\Rightarrow$ Базовый классификатор настраивается только на шумовые объекты, что приводит к неустойчивости ответов и переобучению.

Взвешивание объектов для задачи классификации

Многие функционалы ошибки классификации выражаются через отступы объектов:

$$Q = \sum_{i=1}^{n} \mathcal{L}(Y_i, \widehat{y}_{T-1}(x_i)) = \sum_{i=1}^{n} \tilde{\mathcal{L}}(Y_i \cdot \widehat{y}_{T-1}(x_i))$$

В этом случае антиградиент принимает вид

$$s_i = Y_i \underbrace{\left(-\frac{\partial \tilde{L}(Y_i \cdot \widehat{y}_{T-1}(x_i))}{\partial (Y_i \cdot \widehat{y}_{T-1}(x_i))} \right)}_{w_i},$$

то есть происходит взвешивание ответов объектов с помощью ошибки на них.

Бустинг

Бустинг для задачи бинарной классификации

Композиция:

$$f(x) = sign(\widehat{y}_T(x)) = sign\left(\sum_{t=1}^{T} \gamma_t b_t(x)\right)$$

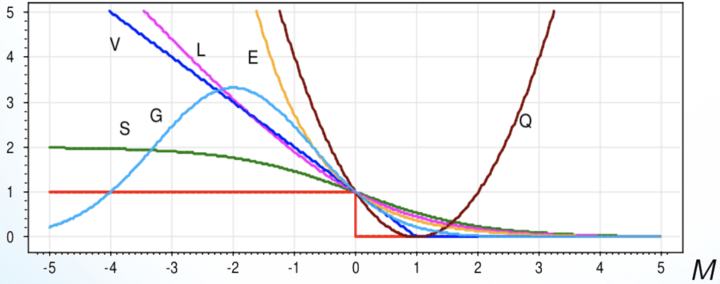
Функционал качества композиции — число ошибок на обучении:

$$Q = \sum_{i=1}^{n} I\{M_i < 0\} = \sum_{i=1}^{n} I\{Y_i \cdot \widehat{y}_T(x_i) < 0\}$$

где M_i — отступ на объекте x_i.

Разные функции потерь для классификации являются апроксимацией пороговой функции потерь $I\{M < 0\}$.

Бустинг для задачи бинарной классификации



$E(M) = e^{-M}$ — экспоненциальная (AdaBoost)

$L(M) = log(1 + e^{-M})$ — логарифмическая (LogitBoost)

$Q(M) = (1 - M)^2$ — квадратичная (GentleBoost)

$G(M) = exp(-cM(M + s))$ — гауссовская (BrownBoost)

$S(M) = 2(1 + e^M)^{-1}$ — сигмоидальная

$V(M) = (1 + M)_+$ — кусочно-линейная

AdaBoost

AdaBoost — экспоненциальная функция потерь.

Оценка исходного функционала Q сверху:

$$Q < \widetilde{Q} = \sum_{i=1}^{n} \exp\Big(- Y_i \sum_{t=1}^{T} \gamma_t b_t(x_i)\Big) =$$

$$= \underbrace{\sum_{i=1}^{n} \exp\Big(- Y_i \sum_{t=1}^{T-1} \gamma_t b_t(x_i)\Big)}_{w_i} \cdot \exp(-Y_i\, \gamma_T b_T(x_i))$$

Нормировочные веса : $\widetilde{W} = (\widetilde{w_1}, .., \widetilde{w_n})$, $\quad \widetilde{w}_i = \frac{w_i}{\sum\limits_{j=1}^{n} w_j}$

Взвешенное число ошибочных и правильных классификаций:

$$N(b, \widetilde{W}) = \sum_{i=1}^{n} \widetilde{w}_i \cdot I\{b(x_i) = -Y_i\} \qquad P(b, \widetilde{W}) = \sum_{i=1}^{n} \widetilde{w}_i \cdot I\{b(x_i) = Y_i\}$$

Можно заметить, что $P(b, \widetilde{W}) = 1 - N(b, \widetilde{W})$

AdaBoost

Теорема (Freund, Schapire, 1995)

Пусть для любого нормированного вектора весов U

существует базовая модель $b \in \mathscr{F}$, классифицирующая выборку

хотя бы немного лучше, чем наугад : $N(b, U) < \frac{1}{2}$.

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

$$b_T = \arg\min_{b \in \mathscr{F}} N(b, \widetilde{W}) =$$

$$= \arg\min_{b \in \mathscr{F}} \sum_{i=1}^{n} \widetilde{w}_i \cdot I\{b(x_i) = -Y_i\}$$

$$\gamma_T = \frac{1}{2} \log \frac{1 - N(b_T, \widetilde{W})}{N(b_T, \tilde{W})}$$

AdaBoost

Теорема (Freund, Schapire, 1995)

Пусть для любого нормированного вектора весов U

существует базовая модель $b \in \mathscr{F}$, классифицирующая выборку

хотя бы немного лучше, чем наугад : $N(b, U) < \frac{1}{2}$.

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

$$b_T = \arg\min_{b \in \mathscr{F}} N(b, \widetilde{W}) = \quad \leftarrow \text{обучение модели}$$

$$= \arg\min_{b \in \mathscr{F}} \sum_{i=1}^{n} \widetilde{w}_i \cdot I\{b(x_i) = -Y_i\}$$

$$\gamma_T = \frac{1}{2} \log \frac{1 - N(b_T, \widetilde{W})}{N(b_T, \tilde{W})}$$

Минимизируем взвешенное число ошибочных классификаций.

AdaBoost

Алгоритм

1. Инициализировать веса объектов : $\widetilde{w}_i = \frac{1}{n}$
2. Для всех t от 1 до T:
3. Обучить базовую модель : $b_t = \arg\min\limits_{b \in \mathscr{F}} N(b, \widetilde{W})$
4. $\gamma_t = \frac{1}{2} \log \frac{1 - N(b_t, \tilde{W})}{T(b_t, \widetilde{W})}$
5. Обновить веса объектов : $\widetilde{w}_i = \tilde{w}_i \cdot \exp(-y_i \gamma_t b_t(x_i))$
6. Нормировать веса: $\widetilde{w}_i = \widetilde{w}_i / \sum\limits_{j=1}^{n} \tilde{w}_j$
7. Отсев шума: отбросить объекты с наибольшими w_i (опционально).

AdaBoost был придуман из соображений взвешивания объектов, хотя по сути является частным случаем градиентного бустинга.

Бустинг

Методы оптимизации второго порядка

Градиентный бустинг осуществляет спуск в пространстве прогнозов модели на обучающей выборке.

Почему бы не воспользоваться методами второго порядка?

Рассмотрим метод Ньютона.

При оптимизации числовой функции $Q(\theta)$ шаг выглядит так:

$$\theta^t = \theta^{t-1} - H^{-1}(\theta^{t-1}) \cdot \nabla_\theta Q(\theta^{t-1})$$

где $H(\theta)$ - матрица вторых производных или матрица Гессе.

Применим его в бустинге.

Уменьшаем следующую функцию:

$$Q(s) = \sum_{i=1}^{n} \mathcal{L}(Y_i, a_{T-1}(x_i) + s_i) \to \min_{s_1, \dots, s_n}$$

Методы оптимизации второго порядка

Вектор градиента:

$$\nabla_s Q(s) = g = \left(\left. \frac{\partial \mathcal{L}}{\partial z} \right|_{z=\widehat{y}_{T-\mathbf{1}}(x_i)} \right)_{i=1}^{n}$$

Матрица вторых производных.

Она будет диагональной, т.к. каждая переменная s_i входит

лишь в одно отдельное слагаемое.

$$H = \operatorname{diag} \left(\left. \frac{\partial^2 \mathcal{L}}{\partial z^2} \right|_{z=\widehat{y}_{T-\mathbf{1}}(x_\mathbf{1})}, \ldots, \left. \frac{\partial^2 \mathcal{L}}{\partial z^2} \right|_{z=\widehat{y}_{T-\mathbf{1}}(x_n)} \right)$$

Можем выписать формулу для сдвигов s:

$$s = -H^{-1} g$$

Далее обучаем базовую модель на сдвиги,

находим коэффициент при ней, добавляем в композицию.

Методы оптимизации второго порядка

Почему так обычно не делают?

- При больших размерах выборки матрица Гессе будет очень большой.
- Работает гораздо дольше.
- В формулах для весов часто происходит деление на 0.
- Не для всех функций потерь (например, для ранжирования) матрица Гессе получается диагональной. Обращение недиагональной матрицы - неустойчивая операция.

Сравнение градиентного бустинга и леса

Случайный лес.

- Требуют большего числа деревьев
- Деревья могут строиться паралельно
- Особо не переобучаются
- Каждое дерево строится дольше
- Проще подбирать гиперпараметры
- Быстрее обучаются

Градиентный бустинг.

- Требуют небольшого числа деревьев
- Деревья строятся последовательно.
- Могут переобучаться
- Каждое дерево строиться быстрее
- Сложнее подбирать гиперпараметры
- Дольше обучаются


ВСЁ!