Машинное обучение

DS-поток

Лекция 7

Бустинг

Ансамбли

Подходы к построению композиций:

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

Бустинг

Бустинг в задаче регрессии

Общий случай градиентного бустинга

Градиентный бустинг над деревьями

Взвешивание объектов
для задачи классификации

Бустинг в задаче регрессии

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

$\mathscr{F}$ — семейство *базовых моделей*

Рассматриваем модели вида

$$\widehat{y}_T(x) = \sum_{t=1}^{T} b_t(x), \quad \text{где } b_t \in \mathscr{F}.$$

Как было в беггинге.

Строим каждую модель независимо на случайной подвыборке и другими случайными факторами.

Как будет в бустинге.

1. Построим одну модель по всей выборке.
2. Посчитаем ошибки модели на обучающей выборке.
3. Построим вторую модель предсказывать ошибки.
4. И т.д.

Бустинг в задаче регрессии

Оптимизируемый функционал — MSE.

1. Построим первую базовую модель:
$$b_1 = \arg\min_{b \in \mathscr{F}} \frac{1}{2} \sum_{i=1}^{n} (b(x_i) - Y_i)^2.$$
2. Посчитаем остатки первой модели: $e_i^1 = Y_i - b_1(x_i)$.
3. Построим вторую базовую модель так, чтобы ее ответы как можно лучше приближали остатки e_i^1:
$$b_2 = \arg\min_{b \in \mathscr{F}} \frac{1}{2} \sum_{i=1}^{n} \left(b(x_i) - e_i^1\right)^2.$$
4. 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:

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

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

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

Таким образом:

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

Бустинг в задаче регрессии

Вспомним, что мы оптимизируем

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

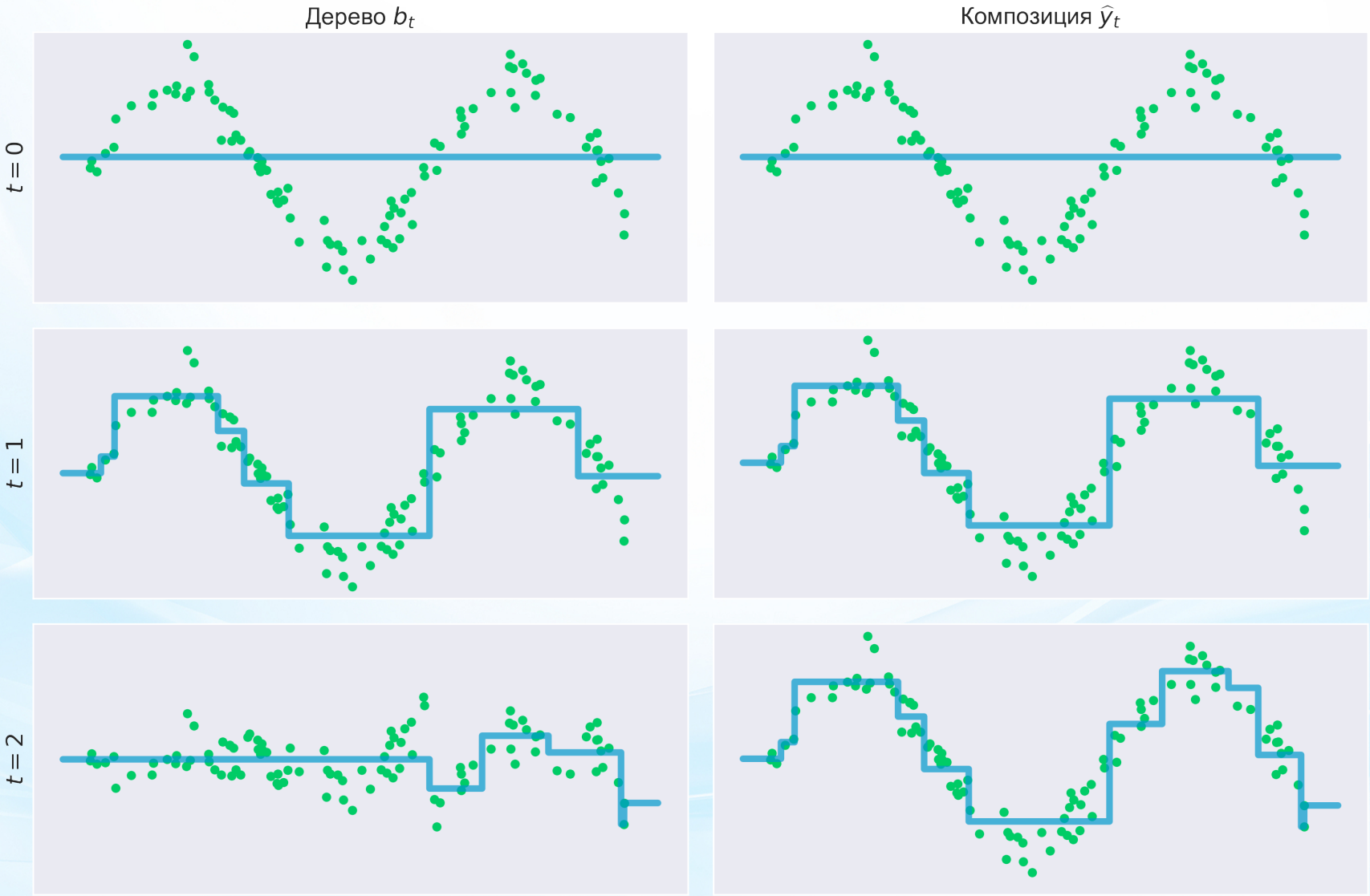
Заметим, что производная Q по ответу модели $\widehat{y}_{t-1}$ на объекте x_i равна $\widehat{y}_{t-1}(x_i) - Y_i = -e_i^{t-1}$. Получаем

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

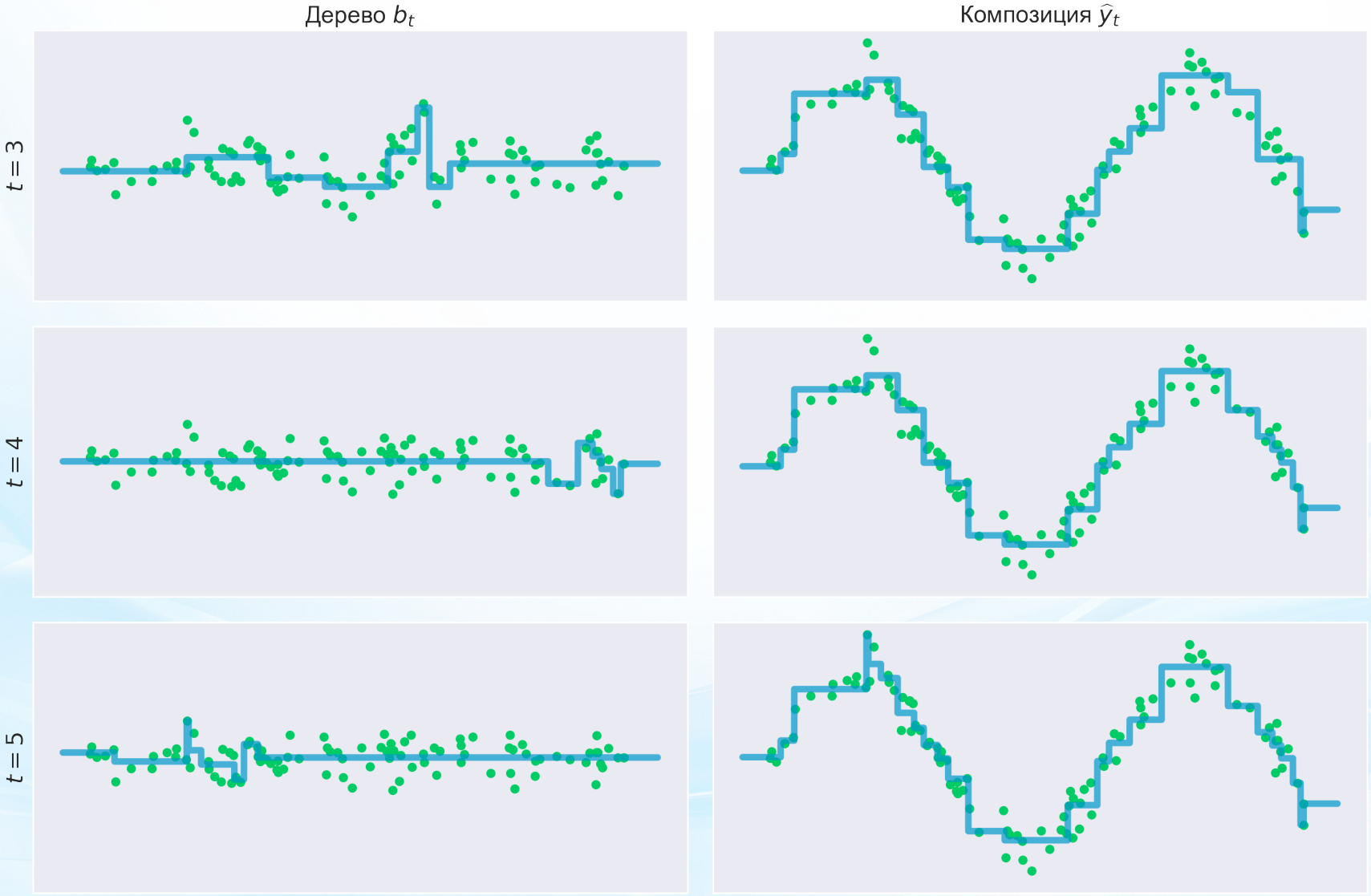
$\Rightarrow$ Модель шагает в сторону антиградиента,
т.е. направления наискорейшего спуска.

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

Пример



Пример



В чем смысл?

Кажется, подобная процедура слишком сложная и неоптимальная.
Оптимизируем $Q(Y, \widehat{y}) = \frac{1}{2}\sum_{i=1}^{n}(\widehat{y}(x_i) - Y_i)^2 \longrightarrow \min\limits_{\widehat{y}}$.

Решение задачи известно: $Q(Y, \widehat{y}) = 0$ при $\widehat{y}(x_i) = Y_i$.
Зачем же выполнять сложную процедуру и обучать на остатках?

Ответ

Мы не можем *в точности* обеспечить условие $\widehat{y}(x_i) = Y_i$,
т.к. ограничены только моделями из класса $\mathscr{F}$.

Соответственно, имея уже какие-то приближения, хочется понять,
в какую сторону стоит сдвинуться, чтобы улучшить предсказания.

Даже любыми моделями, которые умеем строить. Если и построить модель, которая обеспечивает выполнение $\widehat{y}(x_i) = Y_i$, то скорее всего она переобучилась.

Почему бы тогда не строить более глубокие деревья?

Они будут слишком шумными и переобученными,
ведь *в листья попадет слишком мало объектов*.

В композиции мы можем точнее предсказывать сдвиги,
используя *достаточно большую часть объектов в листьях*.

Бустинг

Бустинг в задаче регрессии

Общий случай градиентного бустинга

Градиентный бустинг над деревьями

Взвешивание объектов
для задачи классификации

Градиентный бустинг

Будем строить взвешенную сумму базовых моделей:

$$\widehat{y}_T(x) = \sum_{t=0}^{T} \gamma_t b_t(x).$$

Под индексом $t = 0$ обозначена **начальная базовая модель**.

- Обычно берут $\gamma_0 = 1$.
- Саму 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 очередной базовой модели

Функционал качества $Q(Y, \widehat{y}) = \sum_{i=1}^{n} \mathcal{L}(Y_i, \widehat{y}(x_i)) \longrightarrow \min_{\widehat{y}}$,

где $\mathcal{L}(y, z)$ — кусочно дифф. функция потерь.

Забудем о том, что нам нужно построить новую модель.

Рассмотрим пространство $\mathbb{R}^n$, в котором решим задачу оптимизации

$$Q(Y, s) = \sum_{i=1}^{n} \mathcal{L}(Y_i, s_i) \longrightarrow \min_{s \in \mathbb{R}^n}$$

градиентным спуском

$$s^t = s^{t-1} - \eta \nabla_s Q(Y, s^{t-1}) = s^{t-1} - \eta g^t$$

$$g^t = \left(\nabla_s \mathcal{L}(Y_i, s_i^{t-1}) \right)_{i=1}^{n}$$

Построение новой базовой модели

Теперь вспомним про модель. В идеале должно быть

$$\widehat{y}_t(x_i) = \widehat{y}_{t-1}(x_i) - \eta \widetilde{g}_i^t$$

$$\widetilde{g}^t = \left(\nabla_s \mathcal{L}(Y_i, s)|_{s=\widehat{y}_{t-\mathbf{1}}(x_i)} \right)_{i=1}^n$$

То есть модель должна выдавать $\widetilde{g}_i^t$ на объектах x_i.

Но такой модели может не быть в $\mathscr{F}$.

Тогда просто **обучим новую модель** по выборке

$(x_1, -\widetilde{g}_1^t), ..., (x_n, -\widetilde{g}_n^t)$, оптимизируя MSE

$$b_t(x) = \arg\min_{b \in \mathscr{F}} \sum_{i=1}^n \left(b(x_i) + \widetilde{g}_i^t \right)^2.$$

Замечания

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

2. Мы получили *приближение* градиентного спуска
   в пространстве $\mathbb{R}^n$ на объектах обучающей выборки,
   дополненное на все признаковое пространство $\mathscr{X}$.

3. В общем случае мы также не можем в точности обеспечить
   $\widehat{y}(x_i) = Y_i$ и пытаемся идти в сторону уменьшения ошибки.

4. Оптимизируем с/к функцию потерь независимо
   от функционала исходной задачи —
   вся информация о $\mathcal{L}$ находится в векторе $\widetilde{g}^t$.

5. Можно использовать и другие функционалы,
   но с/к ошибки обычно достаточно.

Выбор коэффициента при базовой модели

Коэффициент при b_t подберем без учета шага обучения η:

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

Зачем он нужен?

Мы выполнили только приближение градиентного спуска, теперь можно немного поправить значения.

Итог:

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

Смысл η

Понижаем доверие к направлению, предсказан. базовой моделью. Обычно, чем меньше η, тем лучше качество итоговой композиции, но требуется больше итераций для сходимости.

Итог

1. Выбрать базовую модель $b_0(x)$, положить $\widehat{y}_0(x) = b_0(x)$.
2. Повторять для $t = 1, ..., T$:

   2.1 Вычислить градиенты по обучающей выборке

$$\widetilde{g}^t = \left(\nabla_s \mathcal{L}(Y_i, s)|_{s=\widehat{y}_{t-1}(x_i)} \right)_{i=1}^{n}.$$

   2.2 Обучить новую модель по MSE по выборке

$$(x_1, -\widetilde{g}_1^t), ..., (x_n, -\widetilde{g}_n^t).$$

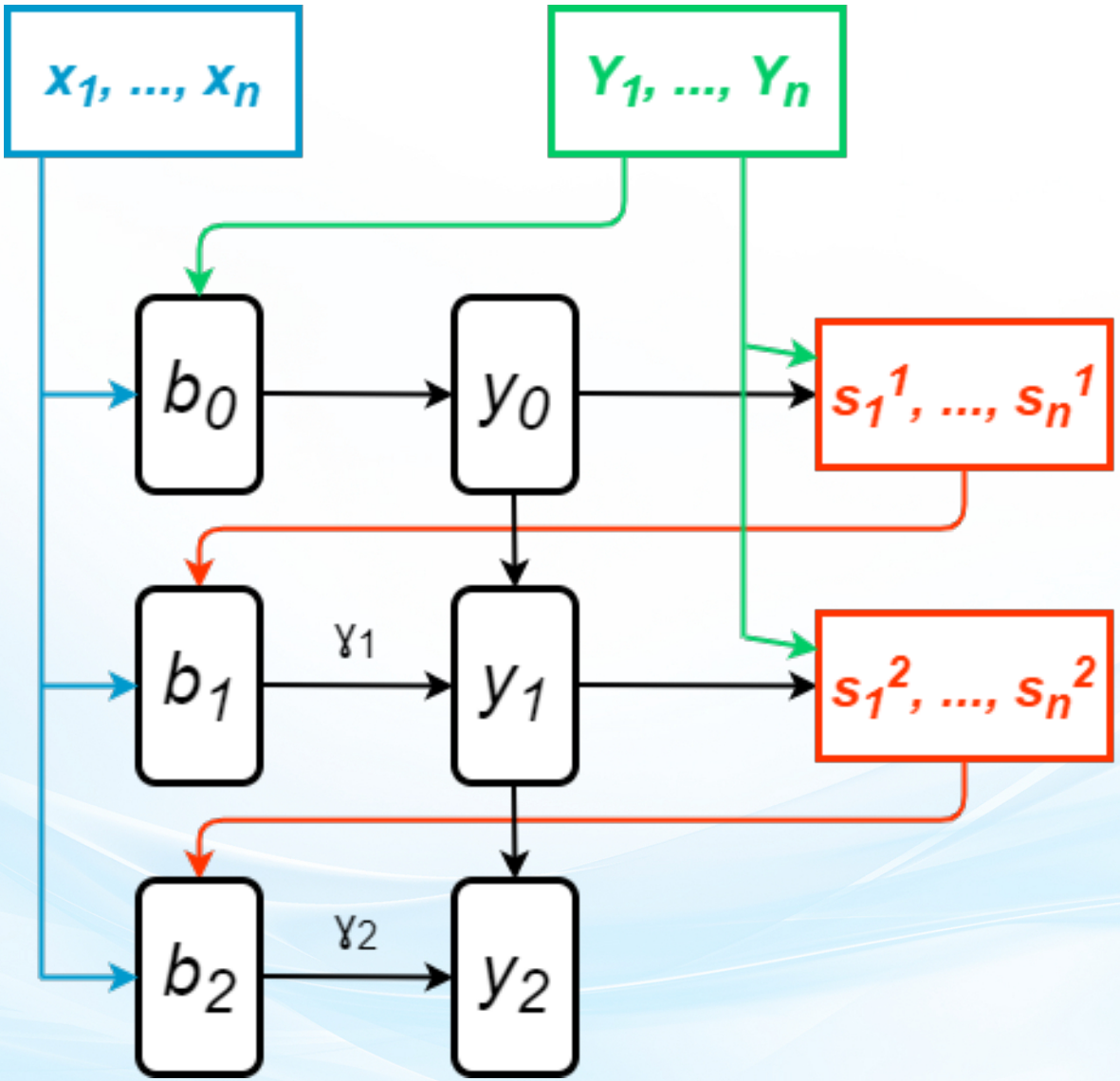
   2.3 Подобрать коэффициент при b_t

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

   2.4 Добавить модель к композиции

$$\widehat{y}_t(x) = \widehat{y}_{t-1}(x) + \eta \widetilde{\gamma}_t b_t(x).$$

Схема градиентного бустинга



Стохастический градиентный ~~спуск~~ 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 обобщающая способность

Рекомендация:

Брать подвыборки, размер которых вдвое меньше исходной выборки.

Частные случаи: Регрессия

- MSE

$$\mathcal{L}(y,z) = \frac{1}{2}(y-z)^2, \qquad \frac{\partial \mathcal{L}(y,z)}{\partial z} = z - y,$$

$$\widetilde{g}^t = \frac{\partial \mathcal{L}(Y_i, \widehat{y}_{t-1}(x_i))}{\partial z} = \widehat{y}_{t-1}(x_i) - Y_i$$

$\Rightarrow$ b_t обучается на выборке $\left(x_i, Y_i - \widehat{y}_{t-1}(x_i)\right)_{i=1}^n$.

- MAE

$$\mathcal{L}(y,z) = |y-z|, \qquad \frac{\partial \mathcal{L}(y,z)}{\partial z} = \operatorname{sign}(z-y),$$

$$\widetilde{g}^t = \frac{\partial \mathcal{L}(Y_i, \widehat{y}_{t-1}(x_i))}{\partial z} = \operatorname{sign}(\widehat{y}_{t-1}(x_i) - Y_i)$$

$\Rightarrow$ b_t обучается на выборке $\left(x_i, -\operatorname{sign}(\widehat{y}_{t-1}(x_i) - Y_i)\right)_{i=1}^n$.

Частные случаи: Классификация

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

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

Экспоненциальная функция потерь:

$$\mathcal{L}(y, z) = \exp(-yz),$$

$$\frac{\partial \mathcal{L}(y, z)}{\partial z} = -y \exp(-yz),$$

$$\widetilde{g}^t = \frac{\partial \mathcal{L}(Y_i, \widehat{y}_{t-1}(x_i))}{\partial z} = -Y_i \cdot \exp(-Y_i \cdot \widehat{y}_{T-1}(x_i)),$$

$\Rightarrow$ b_t обучается на выборке $\left(x_i, Y_i \cdot \exp(-Y_i \cdot \widehat{y}_{T-1}(x_i))\right)_{i=1}^n$.

Бустинг

Бустинг в задаче регрессии

Общий случай градиентного бустинга

Градиентный бустинг над деревьями

Взвешивание объектов
для задачи классификации

Градиентный бустинг над деревьями

Решающее дерево разбивает все пространство на *непересек. области*, в которых его ответ равен константе:

$$b_T(x) = \sum_{k=1}^{\ell_T} b_{Tk} \cdot I\{x \in R_k\}$$

где $k = 1, \ldots, \ell_T$ — индексы листьев,

R_k — соответствующие области разбиения: $\bigsqcup_{k=1}^{\ell_T} R_k = \mathscr{X}$,

b_{Tk} — значения в листьях.

На T-й итерации композиция обновляется как

$$\widehat{y}_T(x) = \widehat{y}_{T-1}(x) + \gamma_T \sum_{k=1}^{\ell_T} b_{Tk} I\{x \in R_k\} = \widehat{y}_{T-1}(x) + \sum_{k=1}^{\ell_T} \underbrace{\gamma_T b_{Tk}}_{\gamma_{Tk}} I\{x \in R_k\}$$

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

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

Перенастройка в листьях

Схема:

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

Тогда потребность в γ_T и b_{Tk} отпадает:

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

Т.к. области разбиения R_k не пересекаются,
задача распадается на ℓ_T **независимых подзадач**:

$$\gamma_{Tk} = \arg\min_{\gamma} \sum_{x_i \in R_k} \mathcal{L}(y_i, \widehat{y}_{T-1}(x_i) + \gamma), \qquad k = 1, \ldots, \ell_T$$

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

Перенастройка в листья

Рассмотрим экспоненциальную функцию потерь.

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

После построения дерева в каждом листе решаем 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 один шаг метода Ньютона из нач. приближения $\gamma_{Tj} = 0$:

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

Bias-variance

*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

Градиентный бустинг над деревьями

Взвешивание объектов
для задачи классификации

Отступ на объекте

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

Решающее правило: $f(x) = \text{sign}(\widehat{y}(x))$.

Введем понятие **отступа на объекте**: $M_i = Y_i \cdot \widehat{y}(x_i)$.

Свойства:

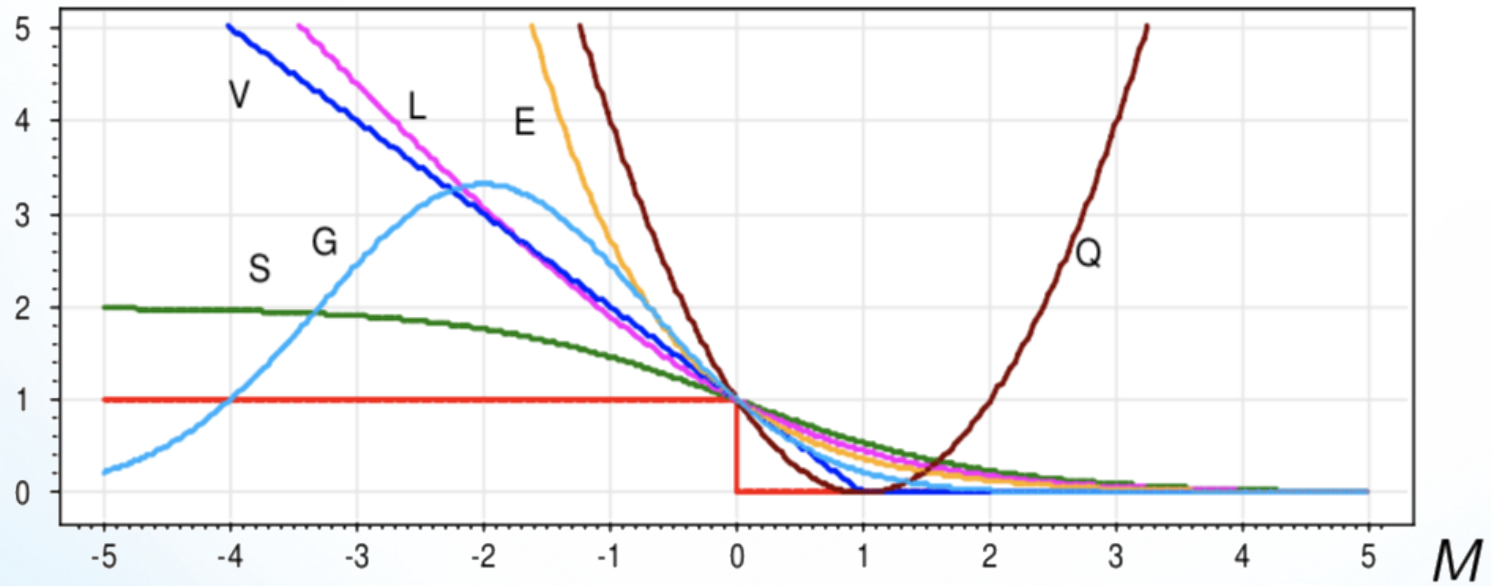
- $M_i > 0 \Leftrightarrow$ объект x_i классифицируется верно.
- $M_i < 0 \Leftrightarrow$ объект x_i классифицируется неверно.
- Чем больше $|M_i|$, тем больше уверенность в своем ответе.

Функционал качества — число ошибок на обучении:

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

В качестве аппроксимации пороговой функции потерь $I\{M < 0\}$ используются разные гладкие функции.

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

Взвешивание объектов для задачи классификации

Экспоненциальная функция потерь.

$$Q(Y, \widehat{y}_T) = \sum_{i=1}^{n} \exp\Big(- Y_i \cdot \widehat{y}_T(x_i)\Big) =$$

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

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

Если $M_i \gg 0$, то данный объект вносит малый вклад в ошибку.
Если $M_i \ll 0$, то данный объект вносит большой вклад в ошибку.

$\Rightarrow w_i$ — мера важности объекта x_i на T-ой итерации.
Причем, что b_t обучается на выборке $\big(x_i, Y_i \underbrace{\exp(-Y_i \widehat{y}_{T-1}(x_i))}_{w_i}\big)_{i=1}^{n}$.

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

AdaBoost

Обозначим

$\widetilde{W} = (\widetilde{w_1}, .., \widetilde{w_n})$, $\quad \widetilde{w_i} = w_i \Big/ \sum_{j=1}^{n} w_j$ — отнормированные веса,

$N(b, \widetilde{W}) = \sum_{i=1}^{n} \widetilde{w_i} \cdot I\{b(x_i) = -Y_i\}$ — взвешенное число ошибочных классификаций.

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 Q достигается при

$$b_T = \arg\min_{b \in \mathscr{F}} N(b, \widetilde{W}), \qquad \gamma_T = \frac{1}{2} \log \frac{1 - N(b_T, \widetilde{W})}{N(b_T, \widetilde{W})}.$$

AdaBoost

1. Инициализировать веса объектов: $\widetilde{w}_i = \frac{1}{n}$.
2. Для всех t от 1 до T:
   - 2.1 Обучить базовую модель: $b_t = \arg\min_{b \in \mathscr{F}} N(b, \widetilde{W})$.
   - 2.2 Вычислить коэффициент $\gamma_t = \frac{1}{2} \log \frac{1 - N(b_t, \widetilde{W})}{N(b_t, \widetilde{W})}$.
   - 2.3 Обновить веса объектов: $\widetilde{w}_i = \widetilde{w}_i \cdot \exp(-Y_i \gamma_t b_t(x_i))$.
   - 2.4 Нормировать веса: $\widetilde{w}_i = \widetilde{w}_i \Big/ \sum_{j=1}^{n} \widetilde{w}_j$.
   - 2.5 Отсев шума: отбросить объекты с наибольшими w_i (опционально).

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

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 строится быстрее
- Сложнее подбирать гиперпараметры
- Дольше обучаются


ВСЁ!