Машинное обучение и дополнительные главы анализа данных

Бустинг

Мы уже знаем

Решающее дерево

- \+ Интерпретируемое;
- \+ Восстанавливает сложные зависимости;
- \- Переобучается;

Случайный лес

- *Усредняет* показания деревьев;
- \+ Имеет низкое смещение и низкий разброс;
- \+ Восстанавливает сложные зависимости;
- \- Деревья глубокие, долгое обучение и предсказание;

Аналогии

Ученики класса совместно учатся писать диктант.

Бэггинг, случайный лес

Обучение всех учеников на разных частях текста.

Голосование всех учеников по всем потенциальным ошибкам,

принятие усредненного варианта.

Бустинг

Последовательное исправление диктанта каждым следующим учеником.

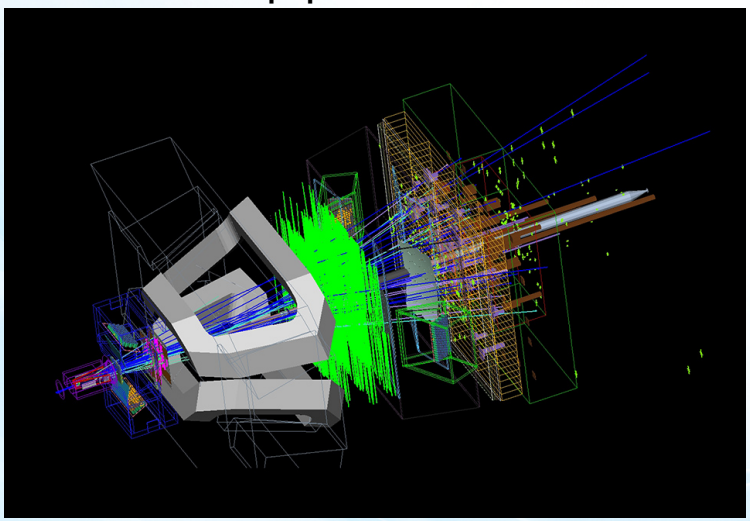
Учится исправлять еще не исправленные *ошибки*, может ошибиться сам.

Пример — большой адронный коллайдер

Данные, полученные на детекторе LHCb, проходят через несколько триггеров, которые оставляют только потенциально интересные события.

Классификацию событий можно делать методами машинного обучения.

Используя возможности машинного обучения (CatBoost), ученым удалось повысить эффективность системы триггеров на 40-50 процентов.



LHCb collaboration, CERN

https://nplus1.ru/material/2017/10/25/cern-yandex

Примеры

- Efficient gradient boosting for prognostic biomarker discovery

  Разработка фреймворка для применения в биологии, тестирование на данных метилирования меланомы.

  https://academic.oup.com/bioinformatics/article-abstract/38/6/1631/6493225

- Diagnostic classification of cancers using extreme gradient boosting algorithm and multi-omics data

  Классификация опухолей на основе мультиомиксных данных.

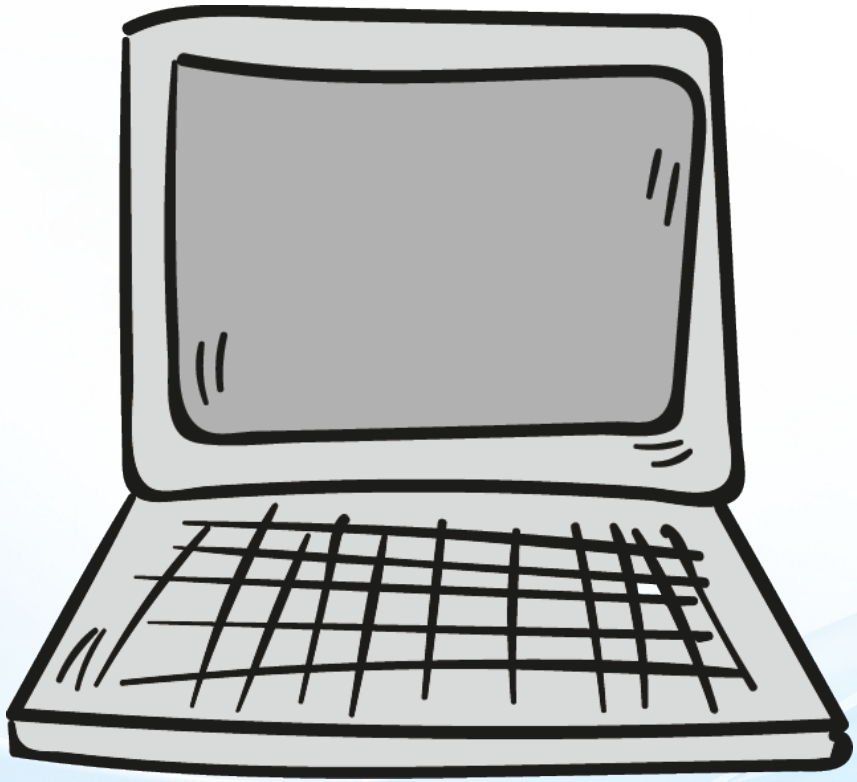
  https://www.sciencedirect.com/science/article/abs/pii/S0010482520301360

- Prediction of protein-protein interaction sites through eXtreme gradient boosting with kernel principal component analysis

  Предсказание сайтов связывания белков.

  https://www.sciencedirect.com/science/article/abs/pii/S0010482521003103

Простая практика



Бустинг

Бустинг в задаче регрессии

Общий случай градиентного бустинга

Вывод для разных функций потерь

Смещение и разброс

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.

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

Вывод для разных функций потерь

Смещение и разброс

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

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

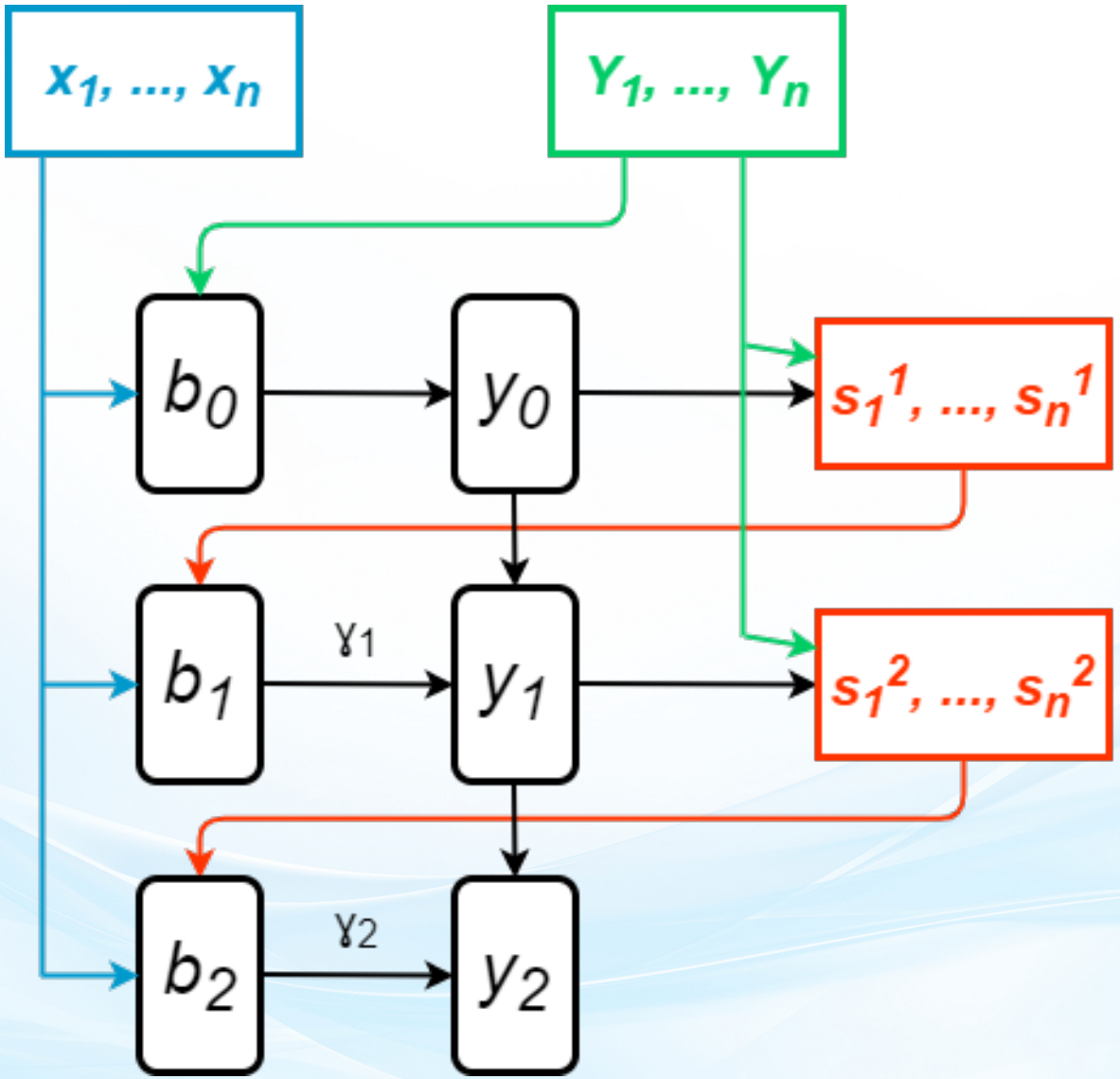
   2.3 Подобрать коэффициент при b_t

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

   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 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

Бустинг в задаче регрессии

Общий случай градиентного бустинга

Вывод для разных функций потерь

Смещение и разброс

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: $Y_i \in \{-1, +1\}$

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

Экспоненциальная функция потерь:

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

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

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

$$= Y_i \cdot \exp(-Y_i \cdot \widehat{y}_{T-1}(x_i))$$

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

Отступ на объекте

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

Решающее правило:

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

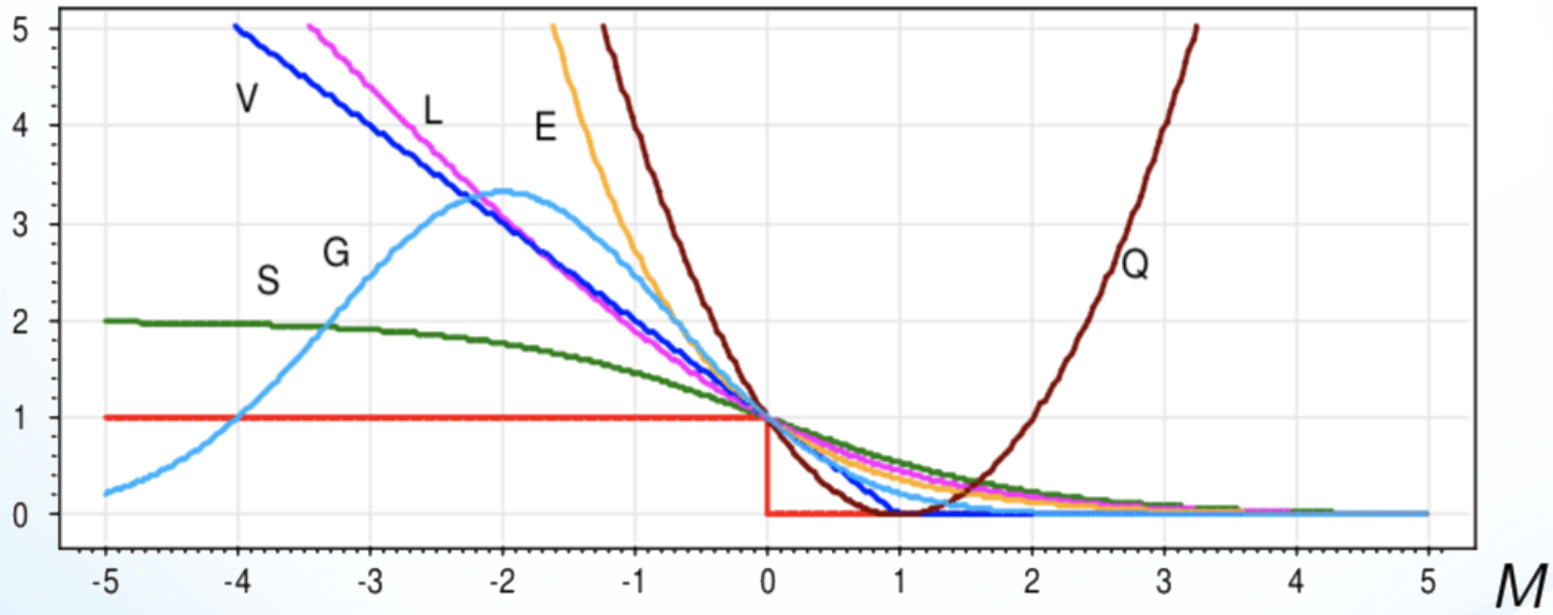
Введем понятие отступа на объекте:

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

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.

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

Смещение и разброс

Какие деревья используются в случайных лесах?

Почему?

Лес

- понижает разброс моделей
- не изменяет смещение

Какие деревья используются в бустинге?

Почему?

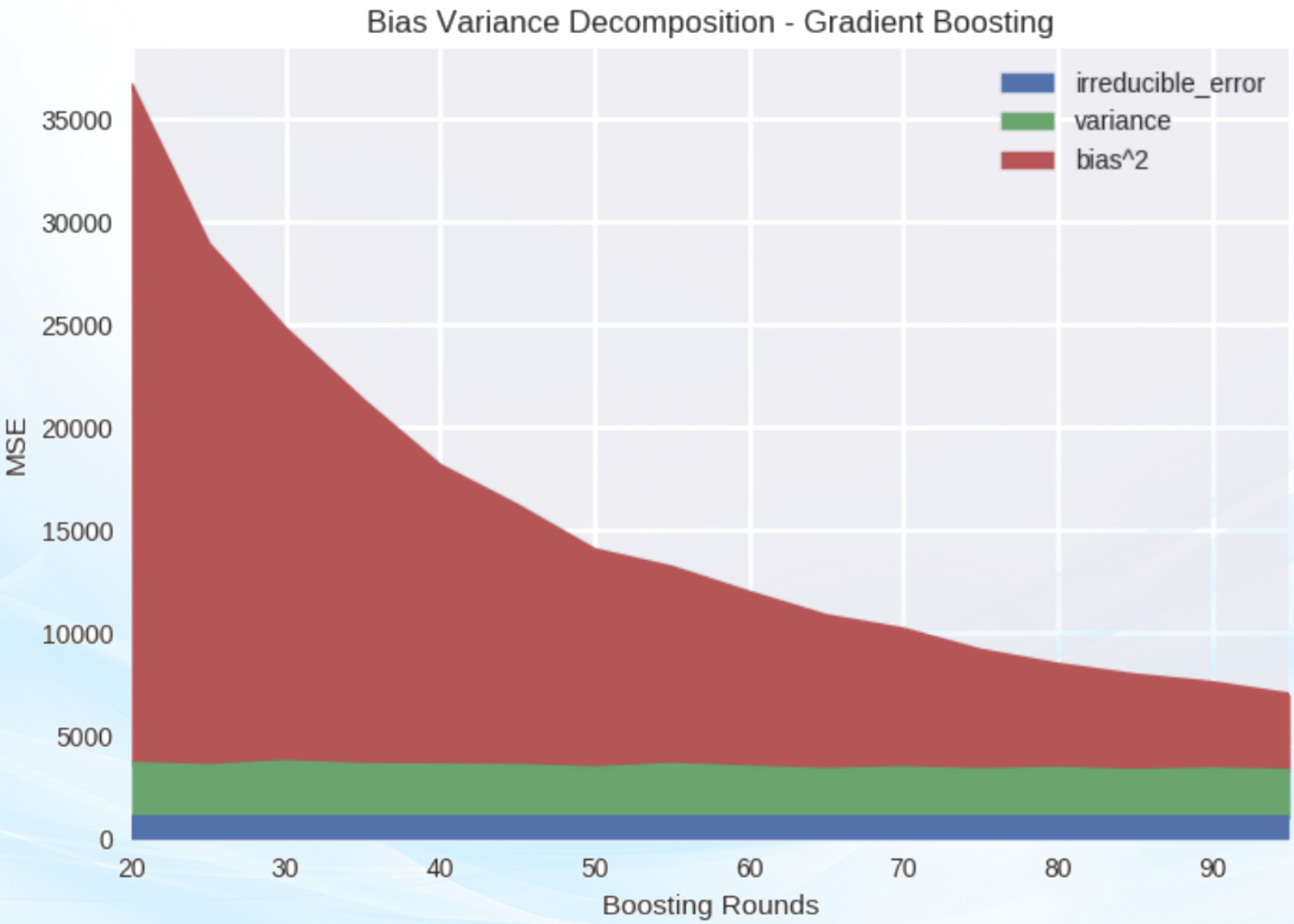
Бустинг

- понижает смещение моделей
- разброс либо останется таким же, либо увеличится

$\Rightarrow$ Нужны модели с большим смещением и низким разбросом.

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

Смещение и разброс



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

Популярные фреймворки

eXtreme Gradient Boosting (XGBoost)

1. Использование вторых производных
   Базовая модель приближает направление,
   посчитанное с учетом вторых производных функции потерь.
2. Регуляризация
   Добавляются штрафы за количество листьев
   и за норму ответов в листьях.
3. Другой критерий информативности (б/д)
   При построении дерева используется критерий информативности,
   зависящий от оптимального вектора сдвига.
4. Другой критерий останова (б/д)
   Критерий останова при обучении дерева также зависит
   от оптимального сдвига.
5. И много другого
   Обработка пропущенных значений, аппроксимация порога t, ...

LightGBM

1. Другая структура дерева.

2. Отбор объектов для построения дерева по градиентам.

3. Exclusive Feature Bundling
   Признаки, которые редко бывают одновременно ненулевыми на одном объекте, заменяются на один признак.

4. Histogram-based поиск оптимального порога (как и в XGBoost).
   Возможные значения признака разбиваются на бины.
   В качестве порогов рассматриваются только граница бинов.

CatBoost

1. Другая структура дерева.

2. Minimum Variance Sampling.
   Признаки для построения дерева выбираются на основе градиента.
   В отличие от LightGBM производится сэмлирование согласно величине градиента.

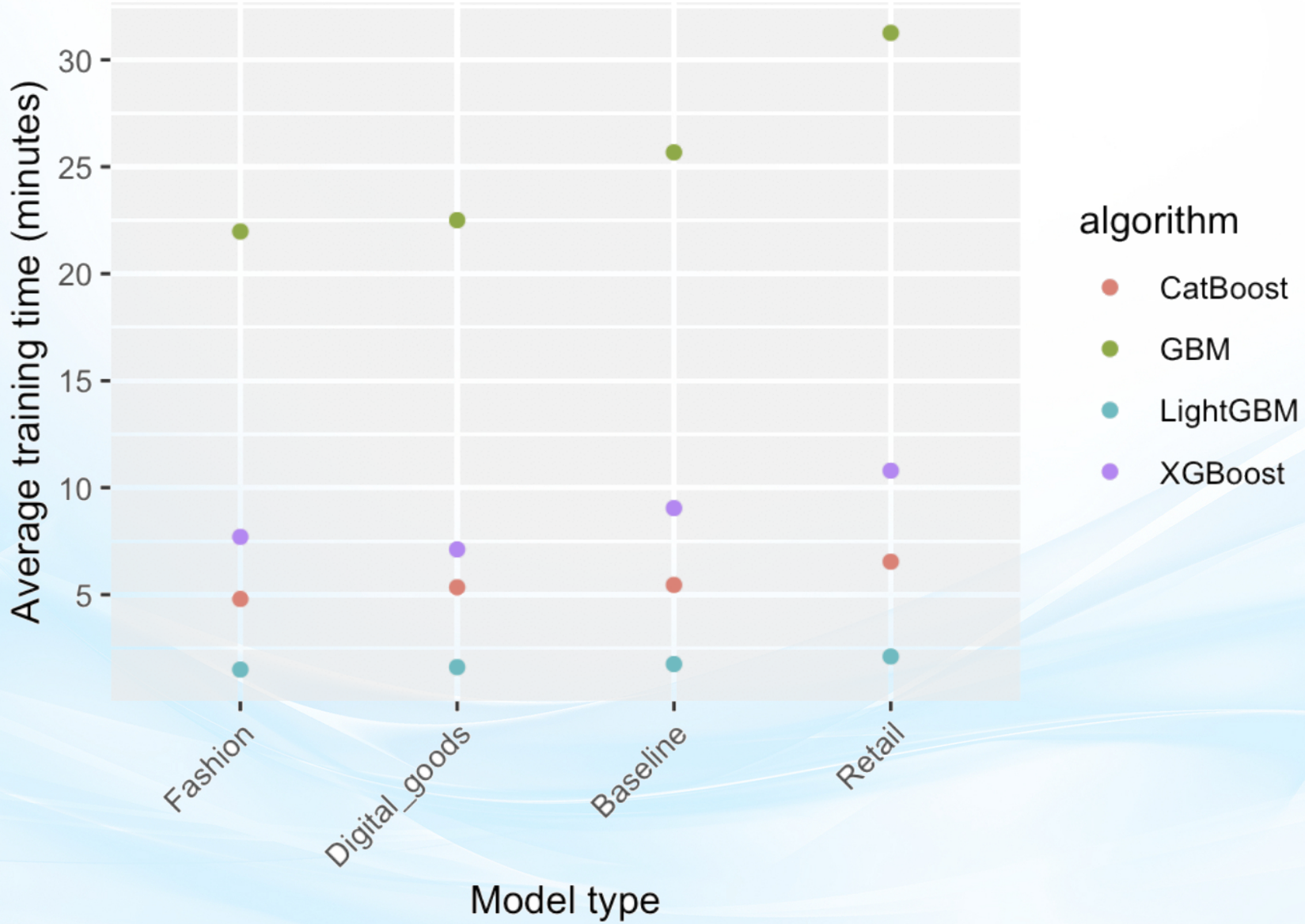
3. **Крутая обработка категориальных признаков.**

4. Histogram-based поиск оптимального порога (как и в XGBoost).
   Возможные значения признака разбиваются на бины.
   В качестве порогов рассматриваются только граница бинов.

5. И многое другое.

Сравнение скорости обучения разных методов

Сравнения времени на обучение для XGBoost, LightGBM, CatBoost.




ВСЁ!