Решающие деревья. Random Forest

Шаповал Егор
Леонович Роман
Попов Владимир

Санкт-Петербург
2022 г.

Решающие деревья

Решающее дерево — бинарное дерево, в котором

- каждой внутренней вершине v приписана функция $\beta_v : X \to \{0, 1\}$,
- каждому листу v приписан прогноз $c_v \in Y$ (возможно вектор вероятностей).

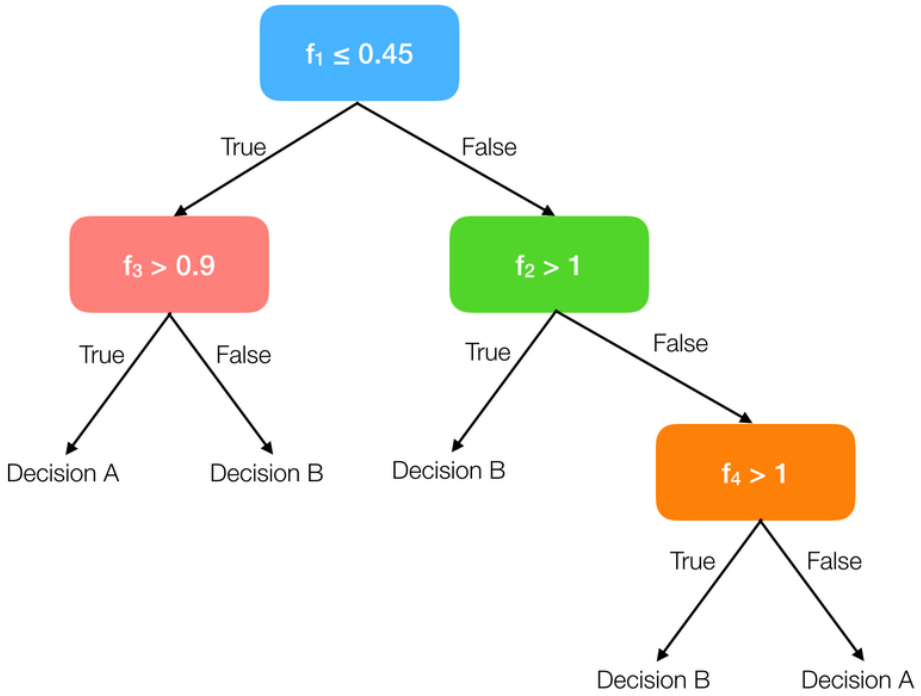

Рис.: Бинарное решающее дерево, f_i — некоторые характеристики

Постановка задачи

Решающие деревья можно применять как для задач регрессии, так и для задач классификации.

Пусть X — множество объектов, Y — множество ответов
$y : X \to Y$ — неизвестная зависимость.

Дано: обучающая выборка — $X^{(n)} = \{(x_i, y_i)\}_{i=1}^{n}$,
$y_i = y(x_i), i = 1, \ldots, n$ — известные ответы.

- $y_i \in \{1, \ldots, K\} \Rightarrow$ задача классификации.
- $y_i \in \mathbb{R} \Rightarrow$ задача регрессии.

Решающие деревья в задаче регрессии

$X \in \mathbb{R}^{n \times p}$ — матрица данных (p признаков, n наблюдений)
$Y \in \mathbb{R}^n$ — отклик.

Идея: разбить совокупность всех возможных значений X_i на J непересекающихся областей $R_1, \ldots, R_J$.

Предсказание для объекта x:

$$f(x) = \sum_{j=1}^{J} c_j \mathbb{1}(x \in R_j).$$

Многомерные прямоугольники $R_1, \ldots, R_J$ выбираем так, чтобы минимизировать сумму квадратов остатков

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - f(x_i))^2 \to \min_{R_1, \ldots, R_J}.$$

Тогда

$$\hat{c}_j = \frac{1}{|R_j|} \sum_{x_i \in R_j} y_i.$$

Решающие деревья в задаче регрессии

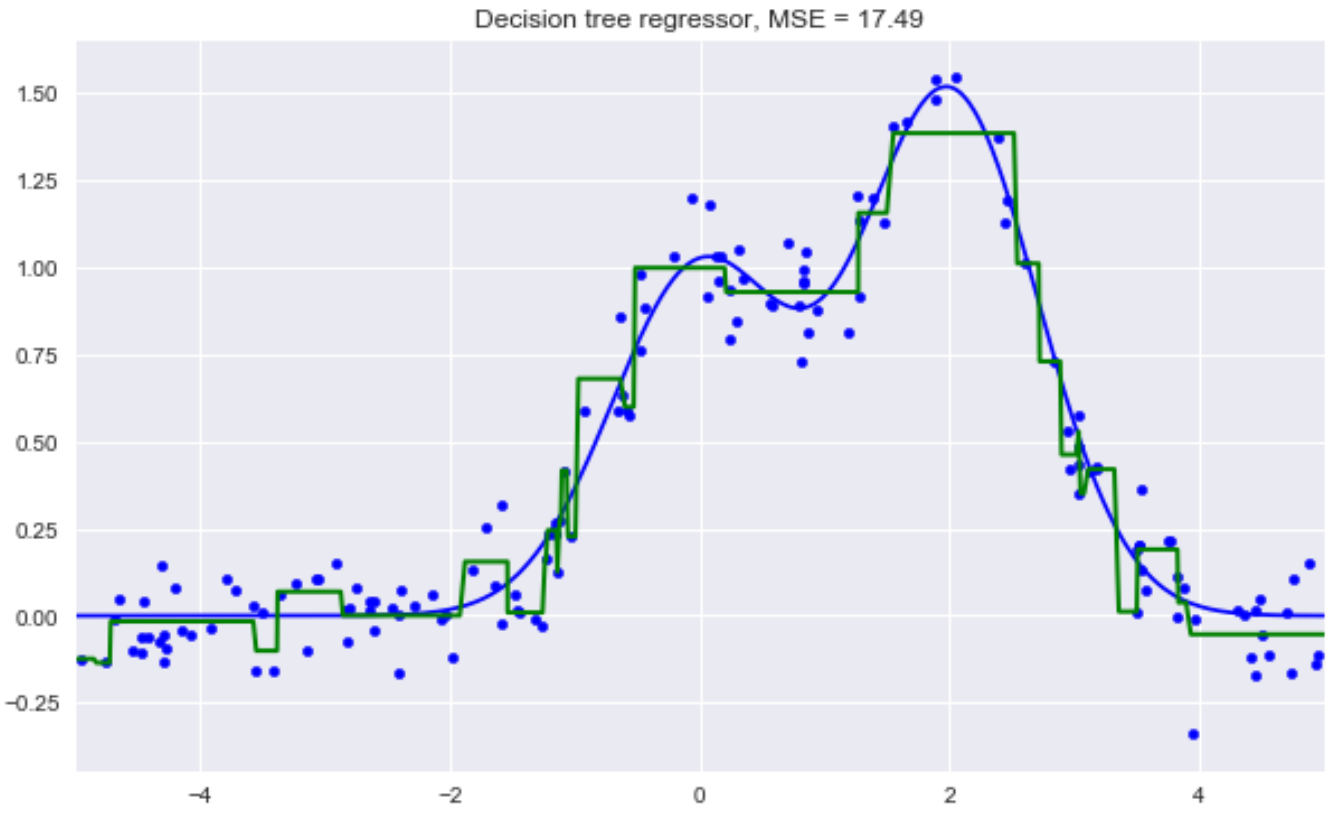

Рис.: Использование решающих деревьев в задачах регрессии

Решающие деревья в задаче классификации

В задаче классификации $R_1, \ldots, R_J$ минимизируется число ошибок классификации

$$M(j) = 1 - \max_k(\hat{p}_{jk})$$

где $\hat{p}_{jk}$ — доля объектов обучающей выборки класса k попавших в R_j.

На практике чаще используют две других метрики:

- $G(j) = \sum_{k=1}^{K} \hat{p}_{jk}(1 - \hat{p}_{jk})$ — индекс Джини,
- $CI(j) = -\sum_{k=1}^{K} \hat{p}_{jk} \log \hat{p}_{jk}$ — коэффициент перекрёстной энтропии,

Предсказание для объекта x:

$$f(x) = \operatorname*{argmax}_{k \in Y} \hat{p}_{jk}.$$

Решающие деревья в задаче классификации

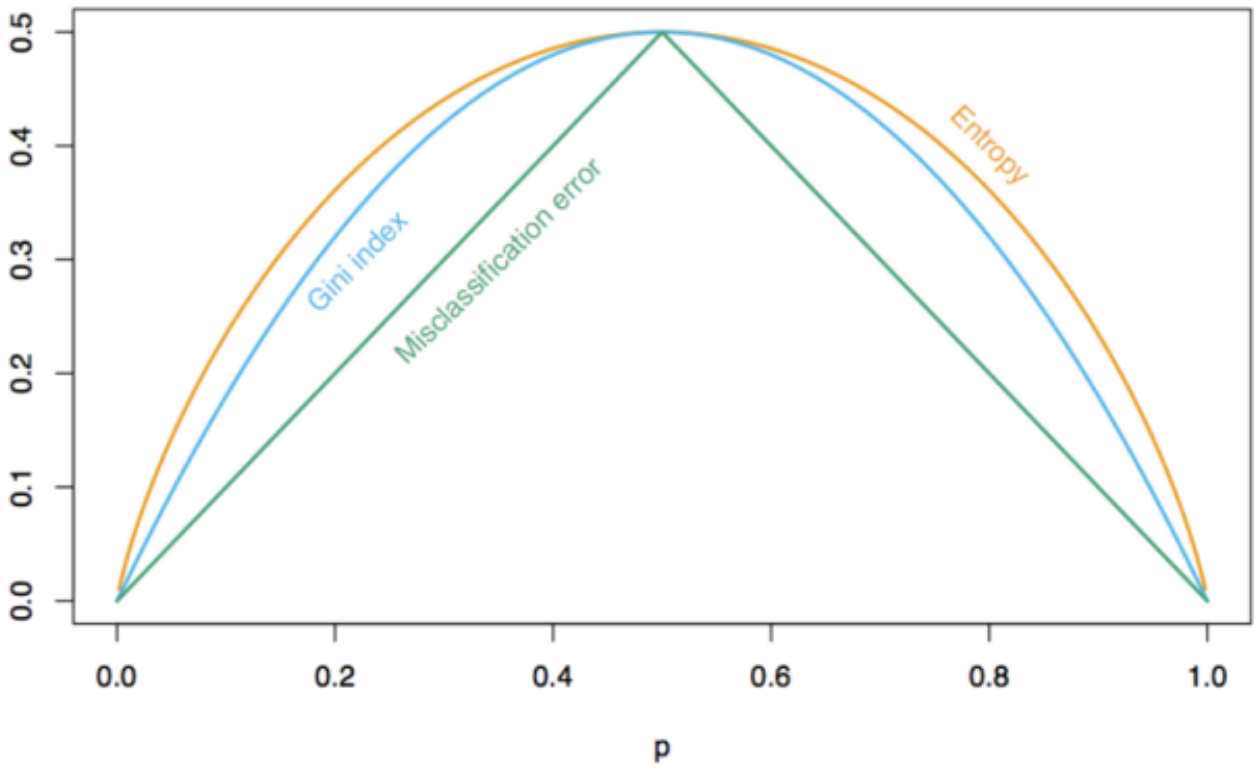

Рис.: Информационные индексы для двухклассовой классовой классификации, как функция от пропорции p для класса 2.

Решающие деревья в задаче классификации

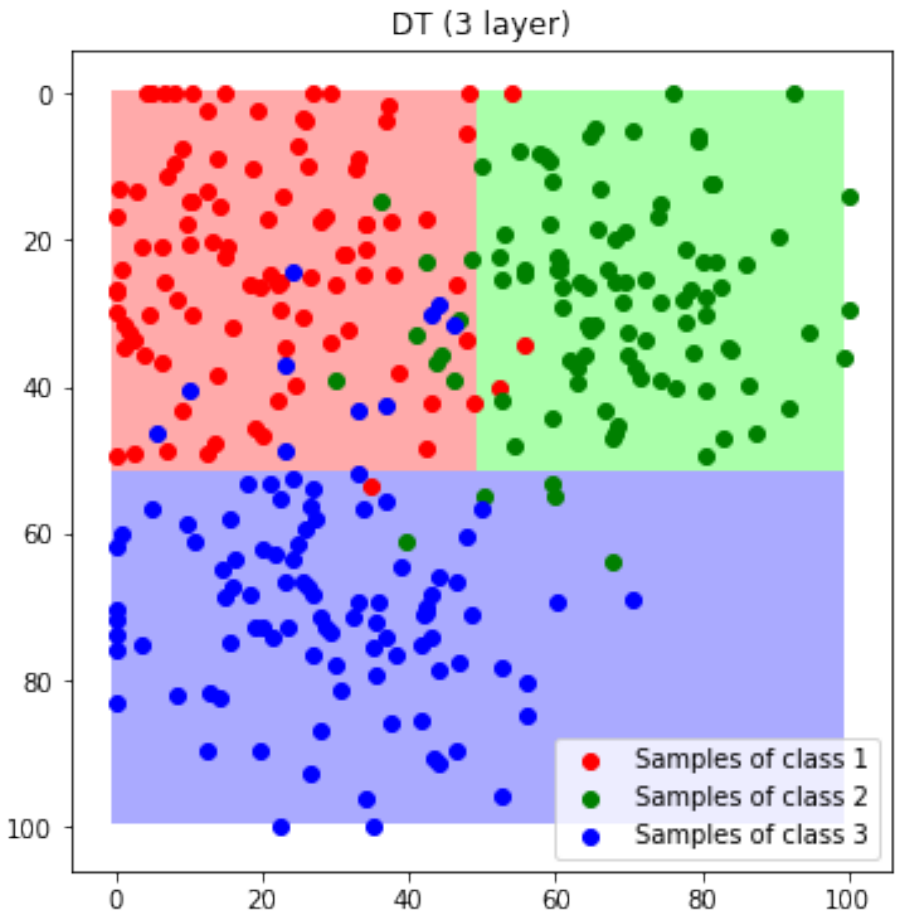

Рис.: Использование решающих деревьев в задачах классификации

Жадный алгоритм построения решающего дерева

Жадный нисходящий алгоритм построения дерева (для задачи регрессии):

1. Выбираем признак j и порог s так, чтобы разбиение $X^{(n)}$ на $R_1(j,s) = \{x \in X^{(n)} | X_j < s\}$ и $R_2(j,s) = \{x \in X^{(n)} | X_j \geq s\}$ решало задачу:

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \to \min_{j,s},$$

   где $\hat{y}_{R_l} = \frac{1}{|R_l|} \sum_{i:x_i \in R_l(j,s)} y_i, \quad l = 1, 2.$

2. Разбиваем выборку на области R_1 и R_2, образуя две дочерние вершины.
3. Повторяем процедуру в пределах каждой получаемой области, пока не выполнится критерий остановки.

На выходе получаем дерево, в каждом из листов которого содержится по крайней мере 1 объект исходной выборки X^n.

Критерии остановки

- Ограничение максимальной глубины дерева.
- Ограничение минимального числа объектов в листе n_{min}.
- Ограничение максимального количества листьев в дереве.
- Остановка в случае, если изменение метрики меньше порога.

Проблема: для очень глубоких деревьев имеем переобучение.

Стрижка деревьев (pruning tree)

- **Проблема:** переобучение — небольшое смещение, но большая дисперсия.
- Объединяя некоторые R_j можем уменьшить дисперсию за счет небольшого увеличения смещения.
- Например, останавливаем рост дерева, когда уменьшение ошибки на следующем разбиении не превзошло некоторого порога.
- Однако мы можем упустить «хорошее» разбиение, такой подход слишком недальновидный.

Cost complexity pruning

- Получим большое дерево T_0 и обрежем его в узле t, получив поддерево $T^t \subset T_0$.
- Рассмотрим последовательность деревьев проиндексированных положительным параметром α. Каждому α соответствует поддерево $T \subset T_0$, минимизирующее критерий

$$Q_\alpha(T) = Q(T) + \alpha|l(T)|,$$

где $Q(T)$ — training error, $\alpha \geq 0$, $|l(T)|$ — число листьев в поддереве T.

- Выберем α с помощью кросс-валидации и возьмём соответствующее поддерево.

Сравнение деревьев с линейными моделями

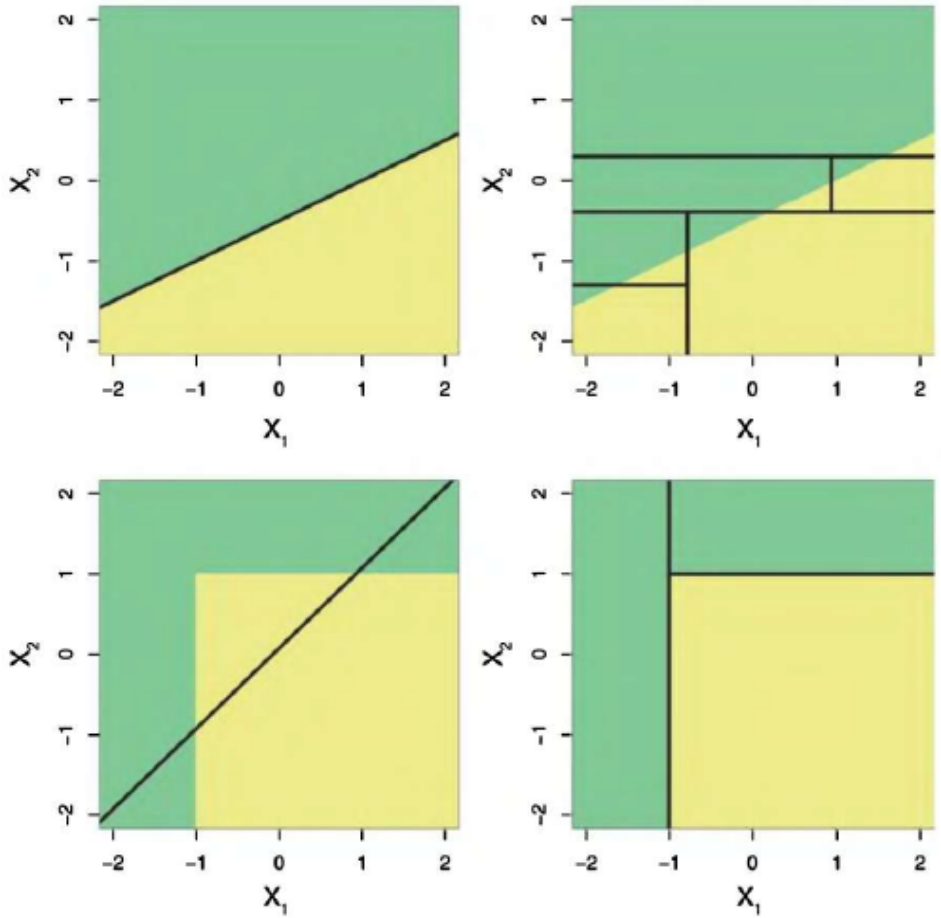

Рис.: Примеры решений задач классификации с линейной (верхний ряд) и нелинейной (нижний ряд) зависимостью. В левой части решение с помощью линейной модели, в правой — с помощью решающего дерева.

Преимущества и недостатки решающих деревьев

Преимущества:

- Простота интерпретации
- Пригодность и для задач регрессии, и для задач классификации
- Возможность работы с пропусками в данных
- Возможность работы с категориальными значениями

Недостатки:

- Основан на «жадном» алгоритме (решение является лишь локально оптимальным)
- Метод явялется неустойчивым и склонным к переобучению

Bootstrap

- 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 $y(x) = \sum \beta_i x_i + \varepsilon_i$ и $p(x)$ — распределение $\mathbf{X}$.
- Ошибка регрессии: $\varepsilon_j(x) = b_j(x) - y(x), \quad j = 1, ..., N.$
- $\mathbb{E}_x \varepsilon_j^2(x) = \mathbb{E}_x \left(b_j(x) - y(x)\right)^2$

Среднеквадратичная ошибка

Средняя ошибка построенных функций регрессии:

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

Пусть

- $\mathbb{E}_x\epsilon_j(x) = 0$ и $\mathbb{E}_x\epsilon_i(x)\epsilon_j(x) = 0,\ i \neq j$
- $a(x) = \frac{1}{N}\sum_{j=1}^{N} b_j(x)$

Тогда

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

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

Bias-Variance decomposition

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.

Ошибка метода обучения

Метод обучения

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

$$L(\mu) = \mathbb{E}_X \left[\mathbb{E}_{x,y} \left[(y - \mu(X)(x)))^2 \right] \right] \tag{1}$$

Среднеквадратичный риск на фиксированной выборке X

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

Подставим это в формулу (1).

Ошибка метода обучения

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

Преобразовываем второе слагаемое:

$$\mathbb{E}_{x,y}\left[\mathbb{E}_X\left[(\mathbb{E}[y|x] - \mu(X))^2\right]\right] =$$
$$= \mathbb{E}_{x,y}\left[\mathbb{E}_X\left[(\mathbb{E}[y|x] - \mathbb{E}_X[\mu(X)] + \mathbb{E}_X[\mu(X)] - \mu(X))^2\right]\right] =$$
$$= \mathbb{E}_{x,y}\left[\mathbb{E}_X\left[\underbrace{(\mathbb{E}[y|x] - \mathbb{E}_X\mu(X))^2}_{\text{не зависит от X}}\right]\right] + \mathbb{E}_{x,y}\left[\mathbb{E}_X\left[(\mathbb{E}_X\mu(X) - \mu(X))^2\right]\right.$$
$$+ 2\mathbb{E}_{x,y}\left[\mathbb{E}_X\left[(\mathbb{E}[y|x] - \mathbb{E}_X[\mu(X)])(\mathbb{E}_X[\mu(X)] - \mu(X))\right]\right] \tag{3}$$

Bias-Variance decomposition

Подставим (3) в (2).

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

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

Bagging

Цель: уменьшение дисперсии модели с сохранением низкого смещения.

Идея: пусть $\xi_1, \ldots, \xi_n$ — н.о.р.с.в., $\mathrm{D}\,\xi_i = \sigma^2$, тогда $\mathrm{D}\,\bar{\xi} = \frac{\sigma^2}{n}$.

Реализация:

$X^n = (x_i, y_i)_{i=1}^n$ — обучающая выборка.

- B бутстреп-выборок (с возвращением) $X_b^{*n}, \quad b = 1, \ldots, B$,
- B решающих деревьев $\{T_b\}_{b=1}^B$,
- находим оценку:
  - в задаче регрессии $\hat{f}_{bag}(x) = \frac{1}{B} \sum\limits_{b=1}^{B} T_b(x)$,
  - в задаче классификации с K классами: записываем класс предсказанный каждым деревом, итоговое предсказание — самый часто встречающийся класс среди предсказаний.

Оценка ошибки out-of-bag

- Дерево, обученное по бутстреп-выборке, использует в среднем $2/3$ наблюдений.
- Оставшуюся $1/3$ выборки называют *out-of-bag* наблюдениями.
- Те деревья, у которых i-ое наблюдение out-of-bag, могут использоваться для предсказания на i. Таким образом можно получить примерно $B/3$ предсказаний.
- В результате получаем способ тестирования bagged модели прямо на обучающей выборке.

Random forest

Идея: уменьшение дисперсии композиции за счёт уменьшения корреляции базовых алгоритмов.

Алгоритм построения случайного леса

1. B bootstrap-выборок $X_b^{*n}, \quad b = 1, \ldots, B$,
2. на основе X_b^{*n} рекурсивно строим решающее дерево T_b, пока не достигнем критерия остановки $(n_{min} = c)$ по следующим правилам для каждого листа:
   - случайно выбираем m признаков (из p),
   - выбираем признак X_j дающий лучшее разбиение из имеющихся m и порог s.
3. построенные деревья $\{T_b\}_{b=1}^{B}$ объединяются в композицию, предсказываем либо по среднему, либо голосованием.

Обычно для классификации $m \approx \sqrt{p}$,

Почему работают bagging и random forest?

Пусть $L(y) = (f(x) - y)^2$ — квадратичная функция потерь, $X^n = (x_i, y_i)_{i=1}^n \sim p(x, y)$, μ — метод обучения.
Среднеквадратический риск:

$$\mathrm{E}_{x,y}(f(x) - y)^2 = \int_X \int_Y L(y)p(x, y)dxdy.$$

Минимум среднеквадратического риска:

$$f^* = \mathrm{E}(y|x) = \int_Y yp(y|x)dx.$$

Мера качества обучения μ:

$$Q(\mu) = \mathrm{E}_{X^n}\,\mathrm{E}_{x,y}(\mu(X^n)(x) - y)^2,$$

где $\mu(X^n)(x)$ — результат применения алгоритма, построенного по выборке X^n, к объекту x.

Bias–variance decomposition

Теорема

В случае квадратичной функции потерь для любого μ

$$Q(\mu) = \underbrace{\mathrm{E}_{x,y}(f^*(x) - y))^2}_{\text{шум (noise)}} + \underbrace{\mathrm{E}_{x,y}(\bar{f}(x) - f^*(x))^2}_{\text{смещение (bias)}} + \\ + \underbrace{\mathrm{E}_{x,y}\,\mathrm{E}_{X^n}(\mu(X^n)(x) - \bar{f}(x))^2}_{\text{разброс (variance)}},$$

где $\bar{f}(x) = \mathrm{E}_{X^n}(\mu(X^n)(x))$.

Смещение и разброс композиции алгоритмов

Пусть $b_t,\ t = 1, \ldots, T$ — базовые алгоритмы, обучающиеся по случайным подвыборкам, $f_T(x) = \frac{1}{T}\sum\limits_{t=1}^{T} b_t(x)$ — композиция алгоритмов.

Смещение композиции совпадает со смещением базового алгоритма:

$$bias = \mathrm{E}_{x,y}(\mathrm{E}_{X^n}\, b_t(x) - f^*(x))^2.$$

Разброс состоит из дисперсии и ковариации:

$$variance = \frac{1}{T}\,\mathrm{E}_{x,y}\,\mathrm{E}_{X^n}(b_t(x) - \mathrm{E}_{X^n}\, b_t(x))^2 +$$

$$+\frac{T-1}{T}\,\mathrm{E}_{x,y}\,\mathrm{E}_{X^n}(b_t(x) - \mathrm{E}_{X^n}\, b_t(x))(b_s(x) - \mathrm{E}_{X^n}\, b_s(x)).$$

Таким образом, чем меньше коррелируют базовые алгоритмы, тем более эффективна их композиция.