Введение в АД

Лекция 5.1

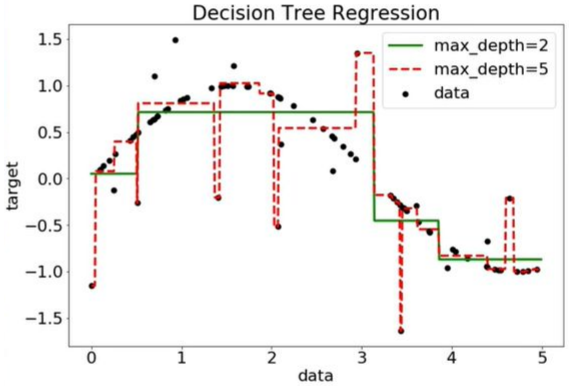
Решающие деревья, ядерная регрессия, k-средние

ФЭФМ МФТИ

Весенний семестр 2023

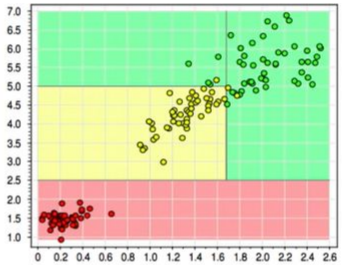
Решающие деревья (decision trees)

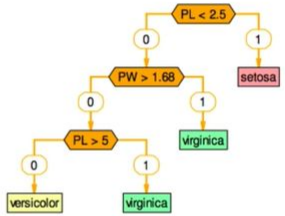
Принцип – делим признаковое пространство на подобласти, в каждой даем какое-то предсказание



Green - decision tree of depth 2
Red - decision tree of depth 5

Every leaf corresponds to some constant.





setosa	$r_1(x) = [PL \leqslant 2.5]$
virginica	$r_2(x) = [PL > 2.5] \wedge [PW > 1.68]$
virginica	$r_3(x) = [PL > 5] \wedge [PW \leqslant 1.68]$
versicolor	$r_4(x) = [PL > 2.5] \wedge [PL \leqslant 5] \wedge [PW < 1.68]$

Построение

- Теоретико-информационный критерий

$$H = -\sum_{i=1}^{n} \frac{N_i}{N} \log\left(\frac{N_i}{N}\right)$$

где *n* — число классов в исходном подмножестве, *N_i* — число примеров i-го класса, *N* — общее число примеров в подмножестве. Выбранное разбиение должно минимизировать энтропию

- Статистический критерий

$$\mathrm{Gini}(Q) = 1 - \sum_{i=1}^{n} p_i^2$$

где *Q* — результирующее множество, *n* — число классов в нем, *p_i* — вероятность i-го класса
Q = 0 – хорошо, Q = 1 – плохо

Редукция (pruning)

- Pre-pruning
  - Ранняя остановка (при достижении целевого параметра)
  - Ограничение глубины дерева
  - Задание минимально допустимого числа примеров в узле

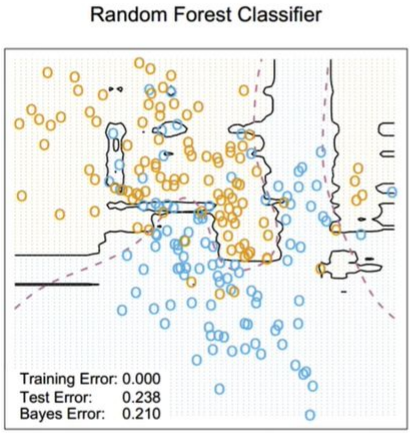
- Post-pruning
  - Упрощение полученного дерева

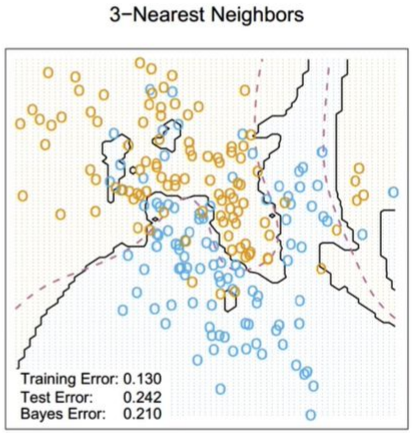
Основные алгоритмы построения деревьев – ID-3 (на энтропии) и CART (на Джини)

Ансамбли

В реальности редко используют просто деревья – используют ансамбли из нескольких обученных деревьев (Bagging)

Bagging + RSM (Random Subspace Method) = Random Forest



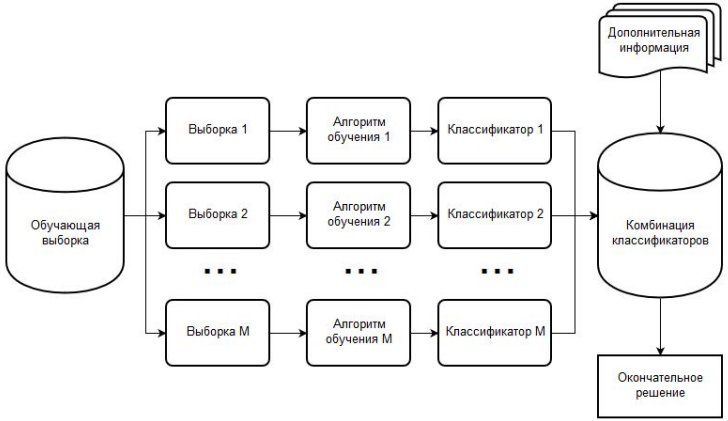


Bootstrap & Bagging

Бутстрэп:

Из всего множества объектов равновероятно выберем **N** объектов с возвращением. Это значит, что после выбора каждого из объектов мы будем возвращать его в множество для выбора. Из-за возвращения некоторые объекты могут повторяться в выбранном множестве.

Обозначим новую выборку через **X1**. Повторяя процедуру **M** раз, сгенерируем **M** подвыборок **X1...XM**. Теперь мы имеем достаточно большое число выборок и можем оценивать различные статистики исходного распределения.



Бэггинг:

- Генерируется с помощью бутстрэпа M выборок размера N для каждого классификатора.
- Производится независимое обучения каждого элементарного классификатора (каждого алгоритма, определенного на своем подпространстве).
- Производится классификация основной выборки на каждом из подпространств (также независимо).
- Принимается окончательное решение о принадлежности объекта одному из классов (консенсусом, большинством или взвешенным решением)

Ядерная регрессия

Значение a(**x**) вычисляется для каждого объекта **x** по нескольким ближайшим к нему объектам обучающей выборки. Для подсчета весов соседних объектов используем т.н. ядерную функцию, такую, что

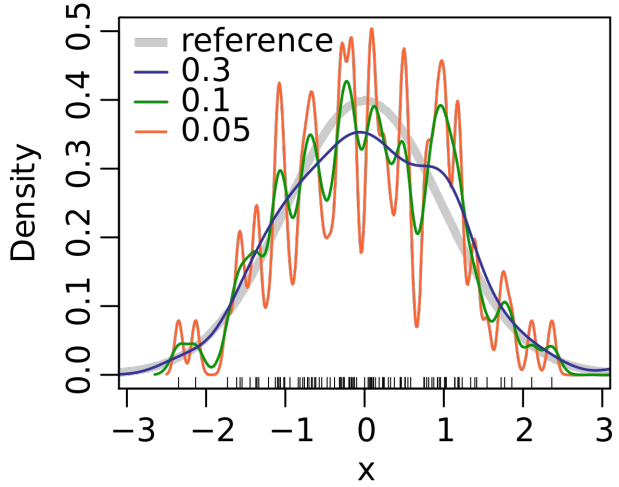
$$\int K(x)\,dx = 1, \quad K(x) = K(-x)$$

the Gaussian kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2),$$

and the Epanechnikov kernel:

$$K(x) = \begin{cases} 3/4(1-x^2) & \text{if } |x| \le 1 \\ 0 & \text{else} \end{cases}$$



Формула Надарая-Уотсона

- Given a choice of kernel K, and a bandwidth h, kernel regression is defined by taking

$$w(x, x_i) = \frac{K\left(\frac{x_i - x}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{x_j - x}{h}\right)}$$

in the linear smoother form (1). In other words, the kernel regression estimator is

$$\hat{r}(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right) \cdot y_i}{\sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right)}$$

Получаем, приравнивая к нулю производную в методе наименьших квадратов в такой форме:

$$Q(\alpha; X^\ell) = \sum_{i=1}^{\ell} w_i(x)(\alpha - y_i)^2 \to \min_{\alpha \in \mathbb{R}}.$$

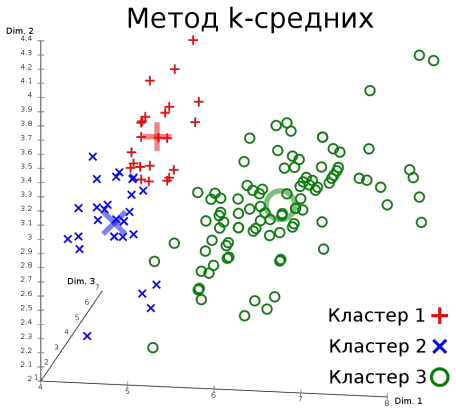
Метод k-средних

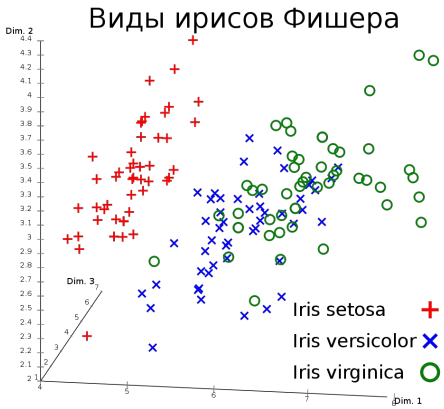
Алгоритм разбивает множество **X** на **k** кластеров **S1,...,Sk** таким образом, чтобы минимизировать сумму квадратов расстояний от каждой точки кластера до его центра масс. В математической форме можно записать так:

$$\arg\min_S \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \rho(\mathbf{x}, \mu_i)^2$$

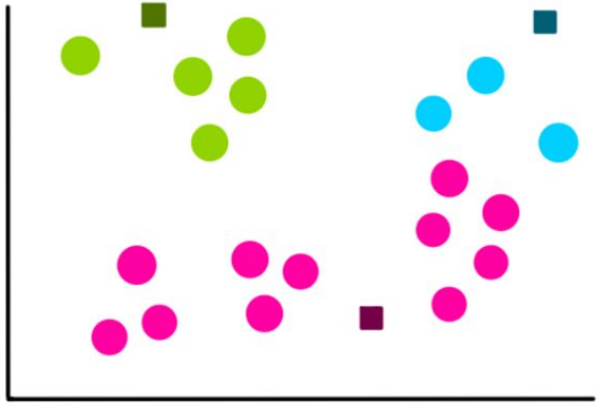
Шаги алгоритма:

- Выбираем количество кластеров **k**
- Выбираем начальные центры кластеров
- Определяем каждый объект из множества **X** в какой-либо из кластеров, находя ближайший центр
- Считаем центры масс получившихся кластеров
- Берем эти центры масс как новые центры и пересчитываем кластеры, повторяем пока не достигли сходимости

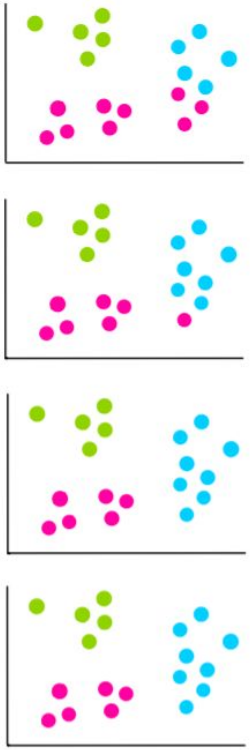




Взяли случайные начальные центры, определили кластеры



Пересчитали центры кластеров



Пересчитали состав кластеров

Сошлись (ошибка не уменьшается)

k-means be like: