Метрические методы

Предположение

Свойства объекта можно узнать, имея представление о его соседях.

Близкие точки похожи.

- Что такое точки?
- Что значит близкие?
- Что значит похожи?

Признаки

- Нормализация
- Фильтрация признаков
- Категориальные, текстовые и прочие невекторные признаки

Метод ближайших соседей. Классификация

- Вычисляем значения факторов интересующей нас точки.
- Находим k ближайших соседеи по выбраннои мере.
- Агрегируем значения искомой характеристики для найденных точек.

Метод ближайших соседей. Классификация

Обучающая выборка $X = (x_i, y_i)_{i=1}^N$, $x_i \in \mathbb{R}^n$, $y_i \in Y = \{1, 2, \dots, C\}$.
Некоторая симметричная функция расстояния

$$\rho : \mathbb{R}^d \times \mathbb{R}^d \to [0, \infty).$$

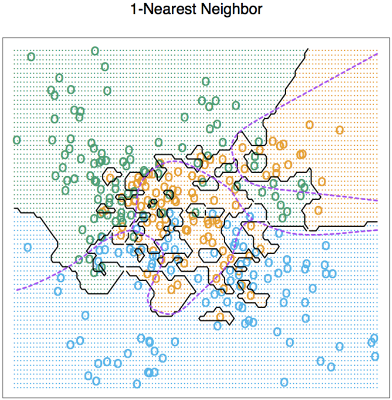
Для нового объекта u находим k наиболее близких в смысле расстояния ρ объектов обучающей выборки. Обозначим их метки $y_u^{(i)}$.

$$a(u) = \arg\max_{y \in Y} \sum_{i=1}^{k} \mathbb{I}[y_u^{(i)} = y]$$

Можем оценивать вероятности классов — частоты классов соседей

$$P(u \sim y) = \frac{1}{k} \sum_{i=1}^{k} \mathbb{I}[y_u^{(i)} = y]$$

Метод ближайших соседей. Классификация



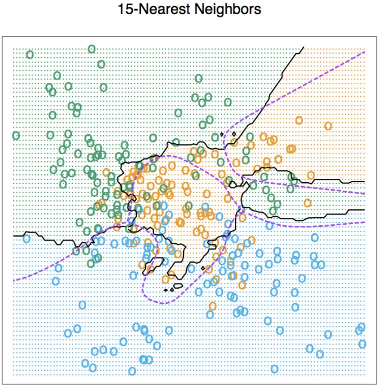


Рис.: KNN с разным k

Расстояния

1. Евклидово
2. Манхэттенское
3. Косинусное
4. Махаланобиса
5. ...

Взвешенный KNN

$$a(u) = \arg\max_{y \in Y} \sum_{i=1}^{k} w_i \mathbb{I}[y_u^{(i)} = y]$$

1. веса w_i зависят от порядка близости
2. веса w_i зависят от расстояния

Ядерная функция $K : \mathbb{R} \to \mathbb{R}$:

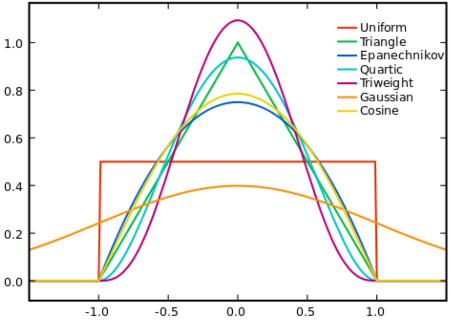
$$a(u) = \arg\max_{y \in Y} \sum_{i=1}^{k} K\left(\frac{\rho(u, x_u^{(i)})}{h}\right) \mathbb{I}[y_u^{(i)} = y],$$

где h — ширина окна.

Примеры ядер

1. $K(x) = \frac{1}{2}\mathbb{I}[|x| \le 1]$ — прямоугольное ядро
2. $K(x) = (1 - |x|)\mathbb{I}[|x| \le 1]$ — треугольное ядро
3. $K(x) = \frac{3}{4}(1 - x^2)\mathbb{I}[|x| \le 1]$ — ядро Епанечникова
4. $K(x) = \frac{15}{16}(1 - x^2)^2\mathbb{I}[|x| \le 1]$ — биквадратное ядро
5. $K(x) = \frac{1}{\sqrt{2\pi}}e^{-2x^2}$ — Гауссовское ядро

Примеры ядер



Метод ближайших соседей. Регрессия

Среднее:

$$a(u) = \sum_{i=1}^{k} y_u^{(i)}$$

Взвешенное среднее:

$$a(u) = \frac{\sum_{i=1}^{k} K\left(\frac{\rho(u, x_u^{(i)})}{h}\right) y_u^{(i)}}{\sum_{i=1}^{k} K\left(\frac{\rho(u, x_u^{(i)})}{h}\right)}$$

соответствует минимизации функции потерь

$$\arg\min_{y \in \mathbb{R}} \sum_{i=1}^{k} K\left(\frac{\rho(u, x_u^{(i)})}{h}\right) (y - y_u^{(i)})^2$$

Прототипирование

Будем выбирать характерные «прототипные» точки

- случайно
- центроиды кластеров
- точки подальше от границ классов

Проклятие размерности

- Точки все ближе «жмутся» к краю
- Углы между точками выравниваются
- Окрестности все чаще упираются в границы
- Для того, чтобы пространство было плотным надо слишком много точек

Свойства KNN

Плюсы:

- Непараметрический, то есть не делает явных предположений о распределении данных
- Простота реализации и наглядность
- Ничего не требует на стадии обучения

Минусы:

- Большое потребление памяти и низкая скорость работы из-за хранения и вычисления расстояний до обучающей выборки
- Чувствителен к масштабу данных, а также к неинформативным признакам
- Необходимо, чтобы метрическая близость объектов совпадала с их семантической близостью

Расстояние Махаланобиса

Мера расстояния между точкой x и распределением D:

$$d_M(x, D) = (x - \mu)^T \Sigma^{-1} (x - \mu)$$

Обощённое расстояние Махаланобиса с $M = L^T L \succ 0$:

$$d_M(x, y) = (x - y)^T M (x - y) = (Lx - Ly)^T (Lx - Ly)$$

равно евклидову расстояние после применения линейной проекции L.

Линейные преобразования глобальны.

KNN + metric learning

Neighbourhood Components Analysis: ищем L, максимизирующее качество KNN при leave-one-out валидации. Может использоваться для сокращения размерности.

Large margin nearest neighbor: ищем M, минимизирующее расстояния между точками одного класса и максимизирующее между точками разных классов:

$$max \sum_{i,j} (1 - y_{i,j}) \sqrt{d_M(x_i, x_j)}$$

$$\sum_{i,j} y_{i,j} d_M(x_i, x_j) \leq 1$$

$$M \succeq 0$$

$y_{i,j}$ равно 1 для одинаковых классов и 0 для разных

Поиск ближайших соседей

1. Точные методы
   - перебор $O(dn)$
   - kd-деревья)
2. Приближённые методы
   - Locality-sensitive hashing (LSH)
   - Hierarchical navigable small world (HNSW)