Введение в искусственный интеллект. Машинное обучение

Тема: Метод ближайших соседей в задачах классификации и регрессии

Бабин Д.Н., Иванов И.Е.

кафедра Математической Теории Интеллектуальных Систем

План лекции

1. Метод ближайших соседей в задаче классификации
2. Теорема о среднем риске метода ближайшего соседа
3. Непараметрическая регрессия
4. Методы поиска ближайшего соседа

Параметрические и непараметрические методы машинного обучения

Параметрические методы

- исходят из предположения, что искомая зависимость имеет некоторый специальный вид с точностью до некоторых параметров
- параметры находятся решением оптимизационной задачи

Параметрические и непараметрические методы машинного обучения

Параметрические методы

- исходят из предположения, что искомая зависимость имеет некоторый специальный вид с точностью до некоторых параметров
- параметры находятся решением оптимизационной задачи

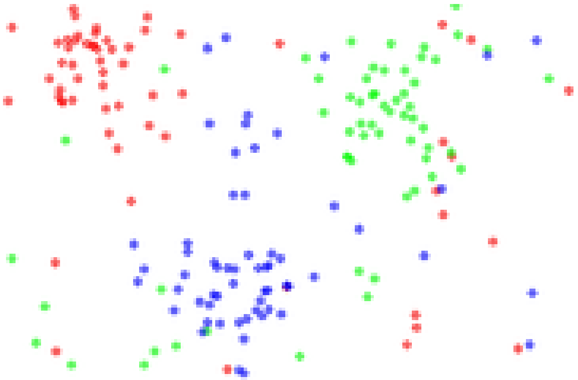
Непараметрические методы

Непараметрические методы – методы не являющиеся параметрическими

- Метрические алгоритмы, ядерные методы

Основное предположение

- "Близкие"объекты лежат в одном классе
- Близость задаётся метрикой
- Типичный пример [1]



[1] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Метод ближайшего соседа

- Параметр метода: метрика
- Алгоритм: по заданной метрике ищем ближайший объект в обучающей выборке и классифицируем объект так же

Метод ближайшего соседа

- Параметр метода: метрика
- Алгоритм: по заданной метрике ищем ближайший объект в обучающей выборке и классифицируем объект так же

Преимущества

- Простота реализации (нет как таковой процедуры обучения в наивной реализации)
- Хорошая интерпретируемость

Метод ближайшего соседа

- Параметр метода: метрика
- Алгоритм: по заданной метрике ищем ближайший объект в обучающей выборке и классифицируем объект так же

Преимущества

- Простота реализации (нет как таковой процедуры обучения в наивной реализации)
- Хорошая интерпретируемость

Недостатки

- Неустойчивость к выбросам
- Неоднозначность классификации при равных расстояниях до двух объектов
- Необходимость хранить всю обучающую выборку
- Алгоритм поиска вычислительно сложен (если обучающая выборка довольно большая)
- Не учитывается значение расстояния

Метод *k* ближайших соседей

- Параметр метода: метрика, **k**
- Алгоритм: по заданной метрике ищем k ближайших объектов в обучающей выборке и классифицируем объект как большинство из k объектов

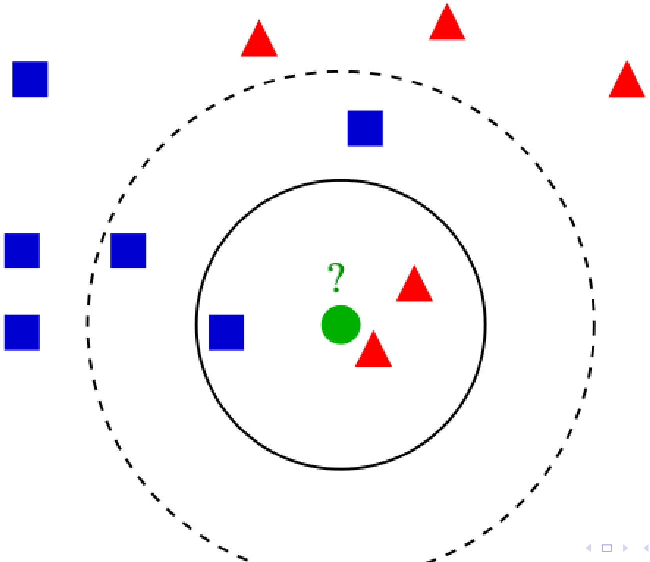
Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Параметр k можно оптимизировать по скользящему контролю

Недостатки

- ~~Неустойчивость к выбросам~~
- ~~Неоднозначность классификации при равных расстояниях до двух объектов~~
- Необходимость хранить всю обучающую выборку
- Алгоритм поиска вычислительно сложен (если обучающая выборка довольно большая)
- Не учитывается значение расстояния

Метод k ближайших соседей



Метод k ближайших взвешенных соседей

- Параметры метода: метрика, k, **веса**
- Алгоритм: по заданной метрике ищем k ближайших объектов в обучающей выборке и классифицируем объект взвешенным голосованием

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Параметр k можно оптимизировать по скользящему контролю

Недостатки

- ~~Неустойчивость к выбросам~~
- ~~Неоднозначность классификации при равных расстояниях до двух объектов~~
- Необходимость хранить всю обучающую выборку
- Алгоритм поиска вычислительно сложен (если обучающая выборка довольно большая)
- ~~Не учитывается значение расстояния~~

Метод k ближайших взвешенных соседей: выбор весов

- Веса в зависимости от порядкового номера

Метод k ближайших взвешенных соседей: выбор весов

- Веса в зависимости от порядкового номера
  - Линейно убывающие веса

Метод k ближайших взвешенных соседей: выбор весов

- Веса в зависимости от порядкового номера
  - Линейно убывающие веса
  - Экспоненциально убывающие веса

Метод k ближайших взвешенных соседей: выбор весов

- Веса в зависимости от порядкового номера
  - Линейно убывающие веса
  - Экспоненциально убывающие веса
  - Любая невозрастающая функция от порядкового номера

Метод k ближайших взвешенных соседей: выбор весов

- Веса в зависимости от порядкового номера
  - Линейно убывающие веса
  - Экспоненциально убывающие веса
  - Любая невозрастающая функция от порядкового номера
- Веса в зависимости от расстояния

Метод k ближайших взвешенных соседей: выбор весов

- Веса в зависимости от порядкового номера
  - Линейно убывающие веса
  - Экспоненциально убывающие веса
  - Любая невозрастающая функция от порядкового номера
- Веса в зависимости от расстояния
  - Любая невозрастающая функция от расстояния

Метод k ближайших взвешенных соседей: выбор весов

- Веса в зависимости от порядкового номера
  - Линейно убывающие веса
  - Экспоненциально убывающие веса
  - Любая невозрастающая функция от порядкового номера
- Веса в зависимости от расстояния
  - Любая невозрастающая функция от расстояния
- Фиксированные веса объектов

Метод k ближайших взвешенных соседей среди набора эталонов

- Параметры метода: метрика, k, веса, **метод выбора эталонов**
- Алгоритм: по заданной метрике ищем k ближайших объектов среди эталонов выбранных из обучающей выборки и классифицируем объект взвешенным голосованием

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Параметр k можно оптимизировать по скользящему контролю

Недостатки

- ~~Неустойчивость к выбросам~~
- ~~Неоднозначность классификации при равных расстояниях до двух объектов~~
- ~~Необходимость хранить всю обучающую выборку~~
- ~~Алгоритм поиска вычислительно сложен~~
- ~~Не учитывается значение расстояния~~

Выбор эталонов

Задача

Получить примерно такое же качество работы алгоритма при меньшем количестве хранимых данных.
Возможно получить улучшение качества, так как в процессе выбора эталонов будут удалены выбросы.

Идеи

- Кластеризация объектов
- Жадный алгоритм

Выбор эталонов кластеризацией k средних (k-means)

Задача

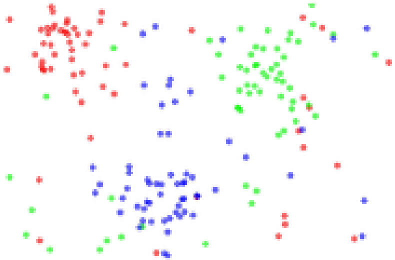
$$V = \sum_{i=1}^{k} \sum_{x \in S_i} (x - \mu_i)^2 \to \min_{S_i},$$

где k — число кластеров, S_i — полученные кластеры, μ_i — центр масс S_i кластера.

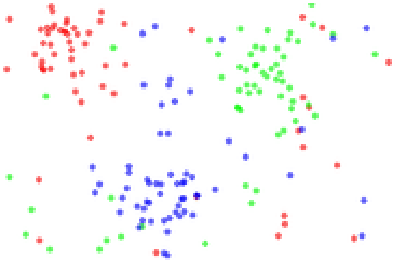
Алгоритм

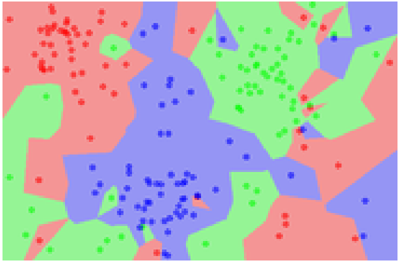
1. Случайно выбираются k элементов из выборки и объявляются центроидами
2. Для фиксированных центроидов каждый элемент выборки относится к одному из кластеров
3. Для фиксированных кластеров вычисляются центроиды
4. Пункты 2,3 повторяются до сходимости

Примеры 1-nn, 5-nn, 1-nn с выбором эталонов

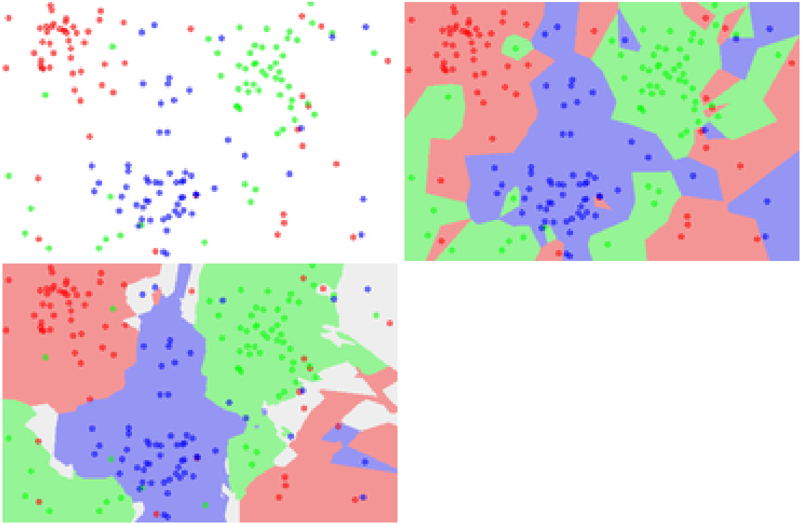


Примеры 1-nn, 5-nn, 1-nn с выбором эталонов

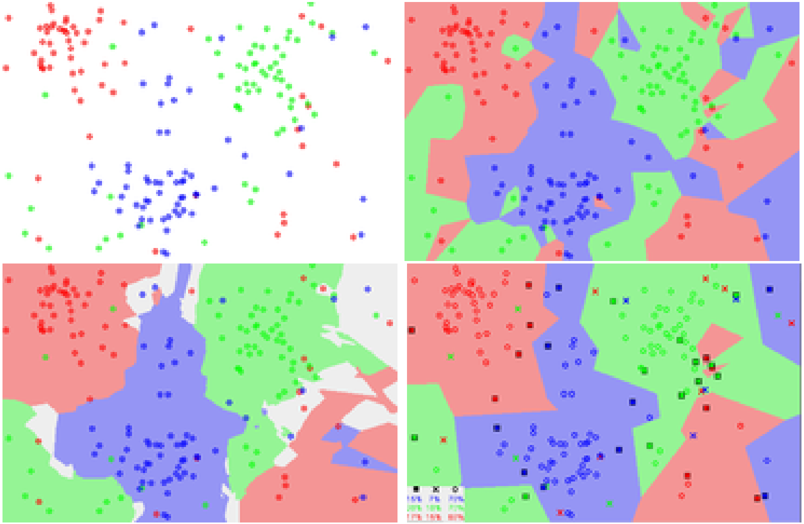




Примеры 1-nn, 5-nn, 1-nn с выбором эталонов



Примеры 1-nn, 5-nn, 1-nn с выбором эталонов



Дополнительные модификации: RadiusNN

Идея

Часть имеет смысл искать соседей на расстоянии не больше чем некоторый радиус r

Параметр r

Вместа входного параметра количества соседей используется радиус

Время для вопросов

Средний риск метода ближайшего соседа [2]

Теорема (Cover-Hart inequality)

1. Для задачи двуклассовой классификации с функцией потерь $L(a(x), y) = [a(x)! = y]$ и непрерывной функцией $\eta(x) = P(y = 1|x)$ выполнено неравенство:

$$R^* \leq R^{1-NN}(\infty) \leq 2R^*(1 - R^*),$$

где $R^{1-NN}(n) = E\ R^n(x)$ — математическое ожидание эмпирического риска метода одного ближайшего соседа для выборки размера n, а $R^{1-NN}(\infty) = \lim\limits_{n\to\infty} R^{1-NN}(n)$.

[2] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13:21–27, January 1967.

Средний риск метода ближайшего соседа [2]

Теорема (Cover-Hart inequality)

1. Для задачи двуклассовой классификации с функцией потерь $L(a(x), y) = [a(x)! = y]$ и непрерывной функцией $\eta(x) = P(y = 1|x)$ выполнено неравенство:

$$R^* \leq R^{1-NN}(\infty) \leq 2R^*(1 - R^*),$$

где $R^{1-NN}(n) = E\ R^n(x)$ — математическое ожидание эмпирического риска метода одного ближайшего соседа для выборки размера n, а $R^{1-NN}(\infty) = \lim\limits_{n\to\infty} R^{1-NN}(n)$.

2. В аналогичных условия для многоклассовой (M классов) классификации выполнено

$$R^* \leq R^{1-NN}(\infty) \leq R^*(2 - \frac{M}{M-1}R^*).$$

[2] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13:21–27, January 1967.

Доказательство теоремы

$$R^{1-NN} = \lim_{n\to\infty} R^{1-NN}(n) = \lim_{n\to\infty} E_{x_0} E_{y_0|x_0} E_{x_1,x_2,\ldots,x_n} L(a(x_0), y_0) =$$

Доказательство теоремы

$$R^{1-NN} = \lim_{n\to\infty} R^{1-NN}(n) = \lim_{n\to\infty} E_{x_0} E_{y_0|x_0} E_{x_1,x_2,\ldots,x_n} L(a(x_0), y_0) =$$

$$= E_{x_0} \lim_{n\to\infty} E_{y_0|x_0} E_{x_1,x_2,\ldots,x_n} L(a(x_0), y_0) = E_{x_0} \lim_{n\to\infty} R_n(x_0)$$

Доказательство теоремы

$$R^{1-NN} = \lim_{n\to\infty} R^{1-NN}(n) = \lim_{n\to\infty} E_{x_0} E_{y_0|x_0} E_{x_1,x_2,\ldots,x_n} L(a(x_0), y_0) =$$

$$= E_{x_0} \lim_{n\to\infty} E_{y_0|x_0} E_{x_1,x_2,\ldots,x_n} L(a(x_0), y_0) = E_{x_0} \lim_{n\to\infty} R_n(x_0)$$

$$R_n(x_0) = E_{y_0|x_0} E_{x_1,x_2,\ldots,x_n} L(a(x_0), y_0) = E_{y_0|x_0} E_{x_1,x_2,\ldots,x_n} L(y_{(1)}, y_0) =$$

Доказательство теоремы

$$R^{1-NN} = \lim_{n\to\infty} R^{1-NN}(n) = \lim_{n\to\infty} E_{x_0} E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(a(x_0), y_0) =$$

$$= E_{x_0} \lim_{n\to\infty} E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(a(x_0), y_0) = E_{x_0} \lim_{n\to\infty} R_n(x_0)$$

$$R_n(x_0) = E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(a(x_0), y_0) = E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(y_{(1)}, y_0) =$$

$$= P(y_{(1)} \neq y_0 | x_{(1)}, x_0) = P(y_{(1)} = 0, y_0 = 1 | x_{(1)}, x_0) + P(y_{(1)} = 1, y_0 = 0 | x_{(1)}, x_0) =$$

Доказательство теоремы

$$R^{1-NN} = \lim_{n\to\infty} R^{1-NN}(n) = \lim_{n\to\infty} E_{x_0} E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(a(x_0), y_0) =$$

$$= E_{x_0} \lim_{n\to\infty} E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(a(x_0), y_0) = E_{x_0} \lim_{n\to\infty} R_n(x_0)$$

$$R_n(x_0) = E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(a(x_0), y_0) = E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(y_{(1)}, y_0) =$$

$$= P(y_{(1)} \neq y_0 | x_{(1)}, x_0) = P(y_{(1)} = 0, y_0 = 1 | x_{(1)}, x_0) + P(y_{(1)} = 1, y_0 = 0 | x_{(1)}, x_0) =$$

$$= P(y_{(1)} = 0 | x_{(1)}) P(y_0 = 1 | x_0) + P(y_{(1)} = 1 | x_{(1)}) P(y_0 = 0 | x_0) =$$

Доказательство теоремы

$$R^{1-NN} = \lim_{n\to\infty} R^{1-NN}(n) = \lim_{n\to\infty} E_{x_0} E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(a(x_0), y_0) =$$

$$= E_{x_0} \lim_{n\to\infty} E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(a(x_0), y_0) = E_{x_0} \lim_{n\to\infty} R_n(x_0)$$

$$R_n(x_0) = E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(a(x_0), y_0) = E_{y_0|x_0} E_{x_1,x_2,\dots,x_n} L(y_{(1)}, y_0) =$$

$$= P(y_{(1)} \neq y_0 | x_{(1)}, x_0) = P(y_{(1)} = 0, y_0 = 1 | x_{(1)}, x_0) + P(y_{(1)} = 1, y_0 = 0 | x_{(1)}, x_0) =$$

$$= P(y_{(1)} = 0 | x_{(1)}) P(y_0 = 1 | x_0) + P(y_{(1)} = 1 | x_{(1)}) P(y_0 = 0 | x_0) =$$

$$= (1 - \eta(x_{(1)}))\eta(x_0) + \eta(x_{(1)})(1 - \eta(x_0))$$

Доказательство теоремы

$$R^{1-NN} = E_{x_0} \lim_{n\to\infty} \left((1-\eta(x_{(1)}))\eta(x_0) + \eta(x_{(1)})(1-\eta(x_0))\right) =$$

Доказательство теоремы

$$R^{1-NN} = E_{x_0} \lim_{n\to\infty} \left((1-\eta(x_{(1)}))\eta(x_0) + \eta(x_{(1)})(1-\eta(x_0))\right) =$$

$$= E_{x_0} \left[2\eta(x_0)(1-\eta(x_0))\right] = E_{x_0} \left[2R^*(x_0)(1-R^*(x_0))\right]$$

$$R^{1-NN} = E_{x_0} \left[2R^*(x_0)(1-R^*(x_0))\right] =$$

Доказательство теоремы

$$R^{1-NN} = E_{x_0} \lim_{n\to\infty} \left((1-\eta(x_{(1)}))\eta(x_0) + \eta(x_{(1)})(1-\eta(x_0)) \right) =$$

$$= E_{x_0} \left[2\eta(x_0)(1-\eta(x_0)) \right] = E_{x_0} \left[2R^*(x_0)(1-R^*(x_0)) \right]$$

$$R^{1-NN} = E_{x_0} \left[2R^*(x_0)(1-R^*(x_0)) \right] =$$

$$= 2E_{x_0} \left[R^*(x_0) \right] - 2E_{x_0} \left[(R^*(x_0))^2 \right] =$$

Доказательство теоремы

$$R^{1-NN} = E_{x_0} \lim_{n\to\infty} ((1-\eta(x_{(1)}))\eta(x_0) + \eta(x_{(1)})(1-\eta(x_0))) =$$

$$= E_{x_0}\left[2\eta(x_0)(1-\eta(x_0))\right] = E_{x_0}\left[2R^*(x_0)(1-R^*(x_0))\right]$$

$$R^{1-NN} = E_{x_0}\left[2R^*(x_0)(1-R^*(x_0))\right] =$$

$$= 2E_{x_0}\left[R^*(x_0)\right] - 2E_{x_0}\left[(R^*(x_0))^2\right] =$$

$$= 2R^* - 2(R^*)^2 - 2D_{x_0}R^*(x_0) = 2R^*(1-R^*) - 2D_{x_0}R^*(x_0) \le 2R^*(1-R^*)$$

Доказательство теоремы

$$R^{1-NN} = E_{x_0} \lim_{n\to\infty} \left((1-\eta(x_{(1)}))\eta(x_0) + \eta(x_{(1)})(1-\eta(x_0))\right) =$$

$$= E_{x_0} \left[2\eta(x_0)(1-\eta(x_0))\right] = E_{x_0} \left[2R^*(x_0)(1-R^*(x_0))\right]$$

$$R^{1-NN} = E_{x_0} \left[2R^*(x_0)(1-R^*(x_0))\right] =$$

$$= 2E_{x_0} \left[R^*(x_0)\right] - 2E_{x_0} \left[(R^*(x_0))^2\right] =$$

$$= 2R^* - 2(R^*)^2 - 2D_{x_0} R^*(x_0) = 2R^*(1-R^*) - 2D_{x_0} R^*(x_0) \leq 2R^*(1-R^*)$$

Нижняя оценка:

$$R^{1-NN} = 2E_{x_0} \left[R^*(x_0)\right] \geq R^*,$$

так как $R^* \leq 0.5$, что и завершает доказательство

Средний риск метода ближайшего соседа

Лемма1.

Пусть $x_0, x_1, ..., x_n$ — независимые одинаково распределенные случайные величины. Тогда $x_{(1)} \to x_0$ с вероятностью 1 при $n \to \infty$.

Лемма 2.

Пусть $\eta(x) = p(y = +1|x)$. Тогда

$$R^*(x) = \min(\eta(x), 1 - \eta(x))$$

Средний риск метода ближайшего соседа

Теорема (Cover-Hart inequality)

1. Для задачи двуклассовой классификации с функцией потерь $L(a(x), y) = [a(x)! = y]$ и непрерывной функцией $\eta(x) = P(y = 1|x)$ выполнено неравенство:

$$R^* \leq R^{1-NN}(\infty) \leq 2R^*(1 - R^*),$$

где $R^{1-NN}(n) = E\ R^n(x)$ — математическое ожидание эмпирического риска метода одного ближайшего соседа для выборки размера n, а $R^{1-NN}(\infty) = \lim\limits_{n\to\infty} R^{1-NN}(n)$.

2. В аналогичных условия для многоклассовой (M классов) классификации выполнено

$$R^* \leq R^{1-NN}(\infty) \leq R^*(2 - \frac{M}{M-1}R^*).$$

Средний риск метода ближайшего соседа

Теорема (Cover-Hart inequality)

1. Для задачи двуклассовой классификации с функцией потерь $L(a(x), y) = [a(x)! = y]$ и непрерывной функцией $\eta(x) = P(y = 1|x)$ выполнено неравенство:

$$R^* \leq R^{1-NN}(\infty) \leq 2R^*(1 - R^*),$$

где $R^{1-NN}(n) = E\ R^n(x)$ — математическое ожидание эмпирического риска метода одного ближайшего соседа для выборки размера n, а $R^{1-NN}(\infty) = \lim\limits_{n\to\infty} R^{1-NN}(n)$.

2. В аналогичных условия для многоклассовой (M классов) классификации выполнено

$$R^* \leq R^{1-NN}(\infty) \leq R^*(2 - \frac{M}{M-1}R^*).$$

Следствие

Если $R^* = 0$ или $R^* = \frac{1}{2}$, то $R^{1-NN}(\infty) = R^*$.

Заключение

- Метод ближайших соседей – простой и хорошо интерпретируемый метод классификации

Заключение

- Метод ближайших соседей – простой и хорошо интерпретируемый метод классификации
- Метод имеет большое число вариаций для настройки

Заключение

- Метод ближайших соседей – простой и хорошо интерпретируемый метод классификации
- Метод имеет большое число вариаций для настройки
    - Подбор метрики (metric learning)

Заключение

- Метод ближайших соседей – простой и хорошо интерпретируемый метод классификации
- Метод имеет большое число вариаций для настройки
  - Подбор метрики (metric learning)
  - Число ближайших соседей

Заключение

- Метод ближайших соседей – простой и хорошо интерпретируемый метод классификации
- Метод имеет большое число вариаций для настройки
  - Подбор метрики (metric learning)
  - Число ближайших соседей
  - Веса во взвешенном варианте метода

Заключение

- Метод ближайших соседей – простой и хорошо интерпретируемый метод классификации
- Метод имеет большое число вариаций для настройки
  - Подбор метрики (metric learning)
  - Число ближайших соседей
  - Веса во взвешенном варианте метода
  - Алгоритм подбора эталонов

Время для вопросов

Непараметрическая регрессия

- Главный минус параметрических моделей, что для описания зависимости необходимо иметь параметрическую модель

Непараметрическая регрессия

- Главный минус параметрических моделей, что для описания зависимости необходимо иметь параметрическую модель
- В случае невозможности подбора адекватной модели имеет смысл пользоваться непараметрическими регрессионными методами

Непараметрическая регрессия

- Главный минус параметрических моделей, что для описания зависимости необходимо иметь параметрическую модель
- В случае невозможности подбора адекватной модели имеет смысл пользоваться непараметрическими регрессионными методами

Предположение

Близким объектам соответствуют близкие ответы

Непараметрическая регрессия

Простейшая модель

Приближаем искомую зависимость константой в некоторой окрестности

Формула Надарая-Ватсона

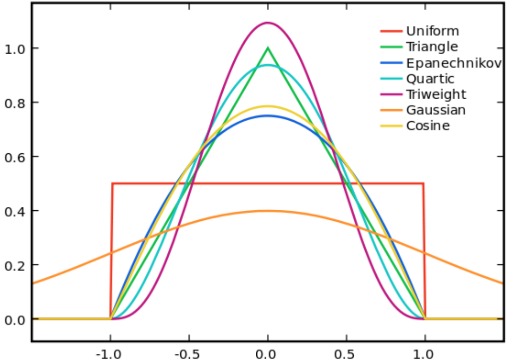
Если в окрестности точки несколько объектов из обучающей выборки, то разумно использовать взвешенное среднее в качестве предсказания алгоритма

$$a(x) = \frac{\sum\limits_i y_i \omega_i(x)}{\sum\limits_i \omega_i(x)},$$

где $\omega_i(x) = K_h(x, x_i)$, а функция K_h называется ядром с шириной окна сглаживания h.

Примеры ядер

- $K_h(x, x_i) = K(\frac{||x - x_i||}{h})$
- Типичные примеры [3]



[3] https://ru.wikipedia.org/wiki/Ядро_(статистика)

Напоминание: Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.

Напоминание: Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.

Полагаем, что ε и a — независимые $(Ea\varepsilon = EaE\varepsilon)$. $Ey = Ef$, $Dy = D\varepsilon = \sigma^2$.

Разложение квадрата ошибки

$$E(y-a)^2 = E(y^2 + a^2 - 2ya) = Ey^2 + Ea^2 - 2Eya =$$

$$= Ey^2 + Ea^2 - 2E(f+\varepsilon)a = Ey^2 + Ea^2 - 2Efa - 2E\varepsilon a =$$

$$= Ey^2 - (Ey)^2 + (Ey)^2 + Ea^2 - (Ea)^2 + (Ea)^2 - 2fEa =$$

$$= Dy + Da + (Ey)^2 + (Ea)^2 - 2fEa = Dy + Da + (Ef)^2 - 2fEa + (Ea)^2 =$$

$$= Dy + Da + (E(f-a))^2 = \sigma^2 + variance(a) + bias^2(f, a)$$

Разброс

$$Variance(a) = D\left(\frac{1}{k}\sum_{i=1}^{k} y(x_{(i)})\right) = \frac{1}{k^2} D\left(\sum_{i=1}^{k} y(x_{(i)})\right) =$$

$$= \frac{1}{k^2} D\left(\sum_{i=1}^{k} (f(x_{(i)}) + \varepsilon_i)\right) = \frac{1}{k^2} D\left(\sum_{i=1}^{k} f(x_{(i)})\right) + \frac{1}{k^2} D\left(\sum_{i=1}^{k} \varepsilon_i\right) =$$

$$= 0 + \frac{1}{k^2} k\sigma^2 = \frac{\sigma^2}{k}$$

Разброс и смещение для kNN

Разброс

$$Variance(a) = D\left(\frac{1}{k}\sum_{i=1}^{k} y(x_{(i)})\right) = \frac{1}{k^2} D\left(\sum_{i=1}^{k} y(x_{(i)})\right) =$$

$$= \frac{1}{k^2} D\left(\sum_{i=1}^{k}(f(x_{(i)}) + \varepsilon_i)\right) = \frac{1}{k^2} D\left(\sum_{i=1}^{k} f(x_{(i)})\right) + \frac{1}{k^2} D\left(\sum_{i=1}^{k} \varepsilon_i\right) =$$

$$= 0 + \frac{1}{k^2} k\sigma^2 = \frac{\sigma^2}{k}$$

Смещение

$$bias^2(f, a) = (E(f(x_0) - a(x_0)))^2 = \left(f(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_{(i)})\right)^2$$

Bias-variance разложение для kNN

$$Error(x_0) = E(a(x_0) - f(x_0))^2 = \left(f(x_0) - \frac{1}{k} \sum_{i=1}^{k} f(x_{(i)}) \right)^2 + \frac{\sigma^2}{k} + \sigma^2$$

- С ростом k разброс уменьшается
- С ростом n смещение уменьшается

Заключение

- Главное преимущество непараметрической регрессии — это отсутствие предположений о виде модели зависимости

Заключение

- Главное преимущество непараметрической регрессии — это отсутствие предположений о виде модели зависимости
- Метод имеет большое число вариаций для настройки

Заключение

- Главное преимущество непараметрической регрессии — это отсутствие предположений о виде модели зависимости
- Метод имеет большое число вариаций для настройки
  - Подбор метрики (metric learning)

Заключение

- Главное преимущество непараметрической регрессии — это отсутствие предположений о виде модели зависимости
- Метод имеет большое число вариаций для настройки
  - Подбор метрики (metric learning)
  - Число ближайших соседей

Заключение

- Главное преимущество непараметрической регрессии — это отсутствие предположений о виде модели зависимости
- Метод имеет большое число вариаций для настройки
  - Подбор метрики (metric learning)
  - Число ближайших соседей
  - Веса во взвешенном варианте метода

Заключение

- Главное преимущество непараметрической регрессии — это отсутствие предположений о виде модели зависимости
- Метод имеет большое число вариаций для настройки
  - Подбор метрики (metric learning)
  - Число ближайших соседей
  - Веса во взвешенном варианте метода
  - Ширину окна сглаживания

Время для вопросов

Где могут быть полезны методы поиска ближайших соседей?

Методы поиска ближайших соседей

Точные

- Полный перебор
- K-мерное дерево (KD-tree)
- Метрическое дерево (ball-tree)

[4] https://arxiv.org/abs/1603.09320

Методы поиска ближайших соседей

Точные

- Полный перебор
- K-мерное дерево (KD-tree)
- Метрическое дерево (ball-tree)

Приближенные

- Locality sensitive hashing (LSH)
- Navigable Small World (NSW)
- HNSW [4]

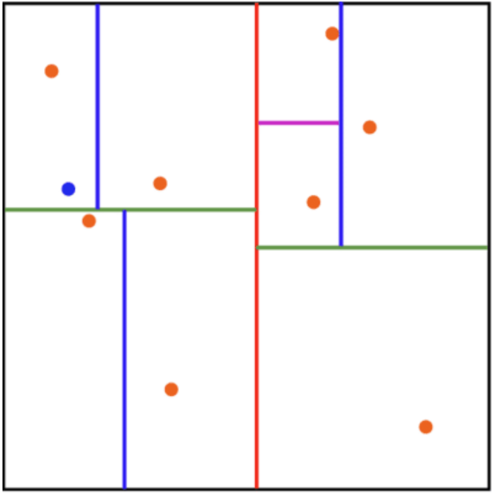
[4] https://arxiv.org/abs/1603.09320

K-мерное дерево [5]

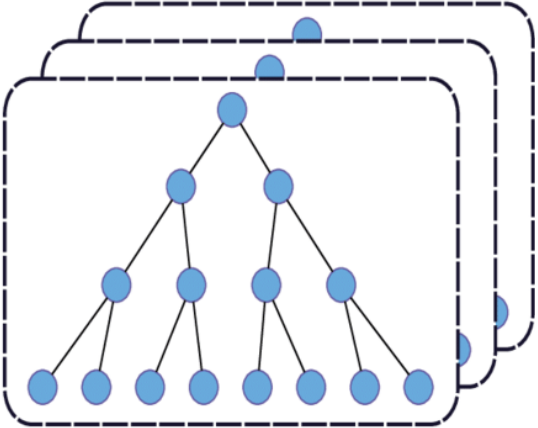
Алгоритм построения

1. Если количество элементов меньше некоторого порогового значения, то отбивается лист и в него помещаются все элементы, в противном случае переходим к следующему пункту
2. Случайно выбирается признак, по которому будет разделение. По этому признаку ищется медиана
3. Все объекты с выбранным признаком левее медианы идет в левое поддерево, остальные в правое
4. Для левого и правого поддерева применяется та же процедура построения

[5] Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. Communications of the ACM. 18 (9): 509–517. doi:10.1145/361002.361007



Kd-Tree in 2D



Multiple Randomized Kd-Trees

Алгоритм поиска I

Для нашего запроса идём по дереву и в соотвествующем листе ищем нужное количество ближайших соседей

Поиск ближайшего соседа в K-мерном дереве

Алгоритм поиска I

Для нашего запроса идём по дереву и в соотвествующем листе ищем нужное количество ближайших соседей

Идея

Если расстояние до дальнего ближайшего соседа меньше, чем расстояние до разделяющей гиперплоскости, то это означает, что во втором поддереве ближайших соседей нет.

Поиск ближайшего соседа в K-мерном дереве

Алгоритм поиска I

Для нашего запроса идём по дереву и в соотвествующем листе ищем нужное количество ближайших соседей

Идея

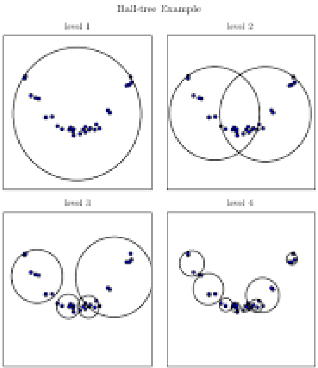
Если расстояние до дальнего ближайшего соседа меньше, чем расстояние до разделяющей гиперплоскости, то это означает, что во втором поддереве ближайших соседей нет.

Алгоритм поиска II

1. Выполняем шаги алгоритма I, считая в каждой вершине расстояние до разделяющей гиперплоскости
2. Делаем обратный ход алгоритма, если расстояние до разделяющей гиперплоскости меньше, чем расстояние до дальнего ближайшего соседа

Идея

Использовать вместо полугиперплоскостей шары



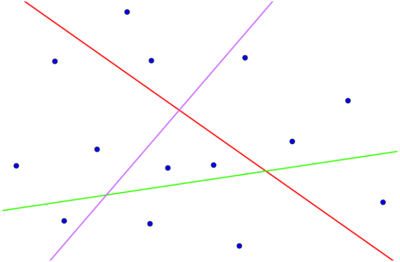
[6] Omohundro, Stephen M. (1989), Five Balltree Construction Algorithms

Идея

Разделить пространство, используя хэш-функции

Пример

В качестве семейства функций можно рассмотеть гиперплоскости



[7] https://codeforces.com/blog/entry/54080?locale=ru

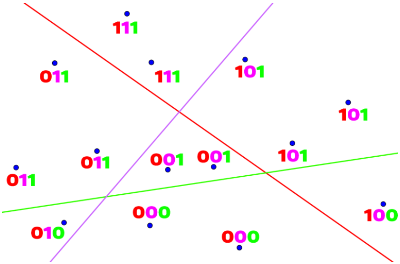
Locality Sensitive Hashing (LSH) [7]

Идея

Разделить пространство, используя хэш-функции

Пример

В качестве семейства функций можно рассмотеть гиперплоскости



[7] https://codeforces.com/blog/entry/54080?locale=ru

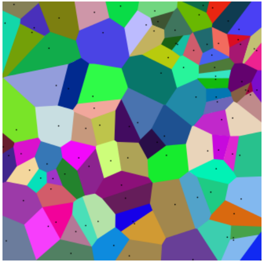
Жадные алгоритмы на графе

Жадный алгоритм

Итеративно ищем ближайшего соседа в графе

Теорема

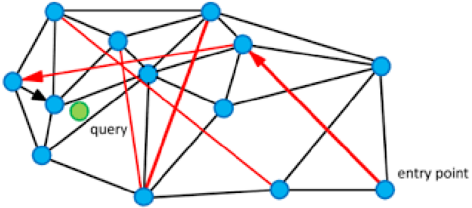
Для графа Делоне жадный алгоритм решает задачу поиска ближайшего соседа точно



Navigable Small World (NSW) [8]

Идея

Поиск по графу типа Small World



Гарантии

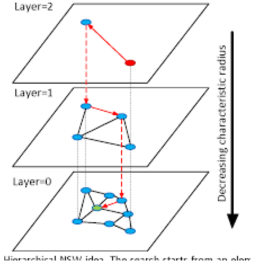
Теоретических гарантий нет, поэтому алгоритм поиска запускают несколько раз в зависимости от требуемой точности

[8] Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov, Approximate nearest neighbor algorithm based on navigable small world graphs, Information Systems, vol. 45, pp. 61-68, 2014.

Hierarchical Navigable Small World (HNSW) [9]

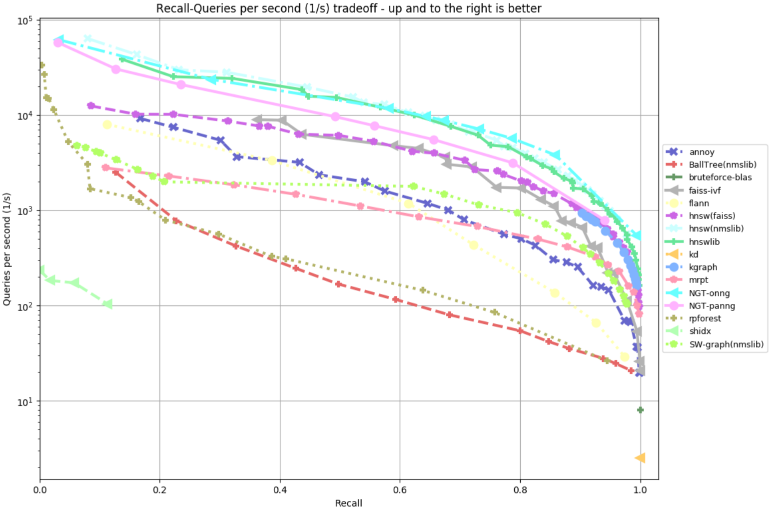
Идея

В графе типа Small World можно выделить подграфы меньшего размера и сделать поиск итеративным



[9] https://arxiv.org/abs/1603.09320

Сравнение методов поиска ближайших соседей [10]



[10] https://github.com/erikbern/ann-benchmarks

Заключение

- Метод поиска ближайших соседей — важная задача теории алгоритмов
- Нужно помнить, что есть методы в среднем быстрее, чем полный перебор
- Для современных индустриальных систем характерно использование не точных, но очень быстрых алгоритмов поиска

Время для вопросов

Спасибо за внимание!