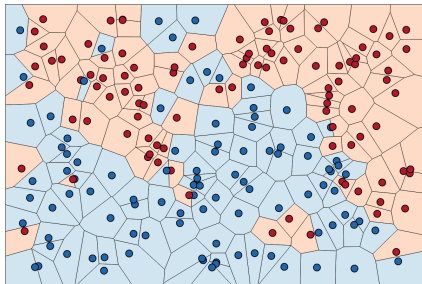


Метрические методы

Виктор Китов

victorkitov.github.io

Курс поддержан
фондом
'Интеллект'



Победитель
конкурса VK среди
курсов по IT



Содержание

- 1 Метод ближайших центроидов
- 2 К ближайших соседей
- 3 Свойства K-NN
- 4 Взвешенный учет объектов
- 5 Регрессия Надарая-Ватсона
- 6 Популярные функции расстояния

Метод ближайших центроидов

- Рассмотрим обучающую выборку $(x_1, y_1), \dots (x_N, y_N)$ с
 - N_1 представителями 1го класса
 - N_2 представителями 2го класса
 - и т.д.

- **Обучение:**

Рассчитать центроиды для каждого класса $c = 1, 2, \dots, C$:

$$\mu_c = \frac{1}{N_c} \sum_{n=1}^N x_n \mathbb{I}[y_n = c]$$

- **Классификация:**

- 1 Для каждого объекта x найти ближайший центроид:

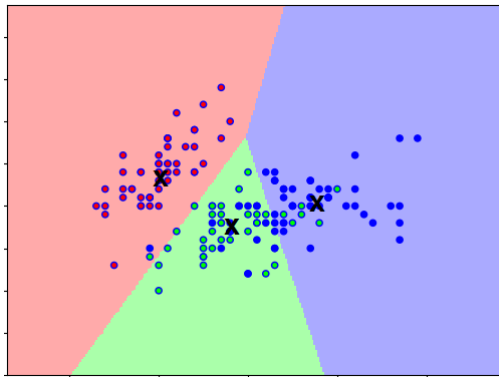
$$c = \arg \min_i \rho(x, \mu_i)$$

- 2 Назначить x самый распространенный класс в центроиде:

$$\hat{y}(x) = c$$

Демонстрация работы

Решающее правило для 3-х классового метода ближайших центроидов.



Пример использования

```
...  
from sklearn.neighbors import NearestCentroid  
from sklearn.metrics import accuracy_score  
  
X_train, X_test, Y_train, Y_test =  
    get_demo_classification_data()  
model = NearestCentroid()           # инициализация модели  
model.fit(X_train, Y_train);        # обучение модели  
Y_hat = model.predict(X_test)       # построение прогнозов  
print(f'Точность: \n{100*accuracy_score(Y_test, Y_hat):.1 f}%')
```

Больше информации. Полный код.

Вопросы

- Чему равны дискриминантные ф-ции $g_c(x)$ метода?
- Какова сложность:
 - обучения?
 - предсказания?
- Что собой представляют границы между классами?
- Применимы ли схожие принципы к регрессии?
(рассмотрите кластеризацию)
- Подвержен ли метод "проклятию размерности"?

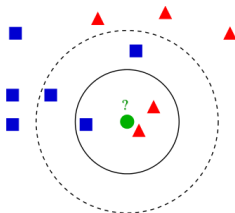
Содержание

- 1 Метод ближайших центроидов
- 2 К ближайших соседей
- 3 Свойства K-NN
- 4 Взвешенный учет объектов
- 5 Регрессия Надарая-Ватсона
- 6 Популярные функции расстояния

Метод ближайших соседей

Классификация:

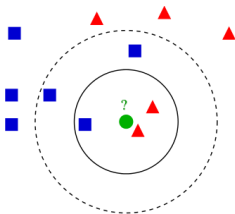
- 1 Найти K ближайших объектов в обучающей выборке к заданному x .
- 2 Сопоставить x самый частотный класс среди K ближайших объектов.



Метод ближайших соседей

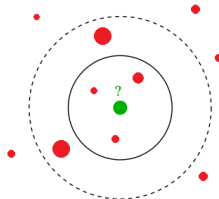
Классификация:

- 1 Найти K ближайших объектов в обучающей выборке к заданному x .
- 2 Сопоставить x самый частотный класс среди K ближайших объектов.



Регрессия:

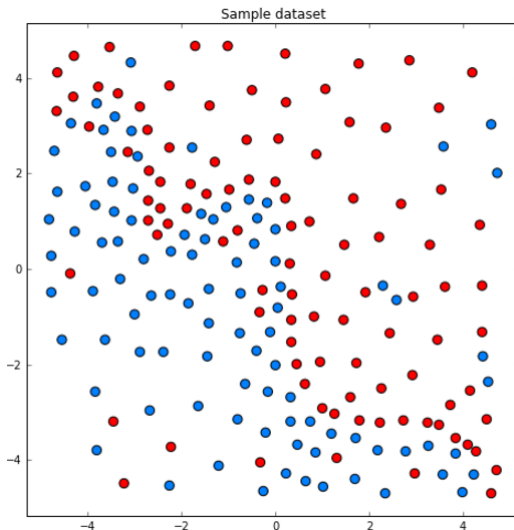
- 1 Найти K ближайших объектов в обучающей выборке к заданному x .
- 2 Сопоставить x среднему отклику среди K ближайших объектов.

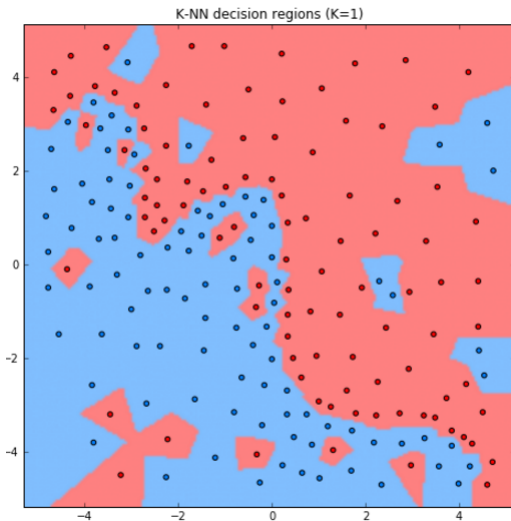


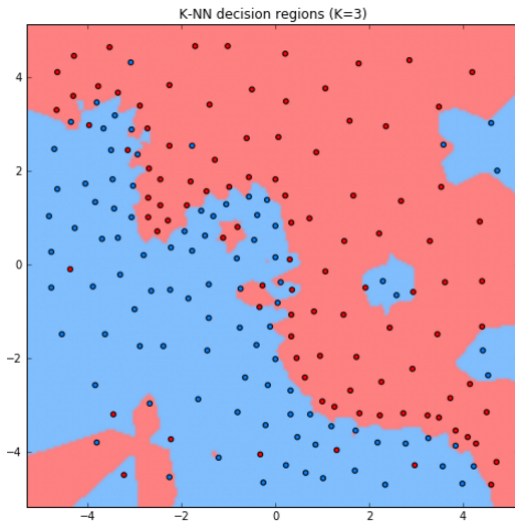
Комментарии

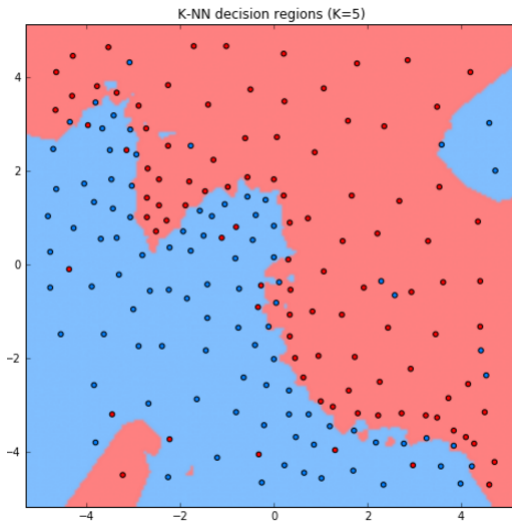
- Англ. *K-nearest neighbors*, сокращенно K-NN.
- $K = 1$: алгоритм ближайшего соседа.
- Базовое предположение: близким объектам соответствуют похожие отклики.
- Как будет работать алгоритм при $K = N$?
- Что вычислительно проще - обучить метод или применять его?

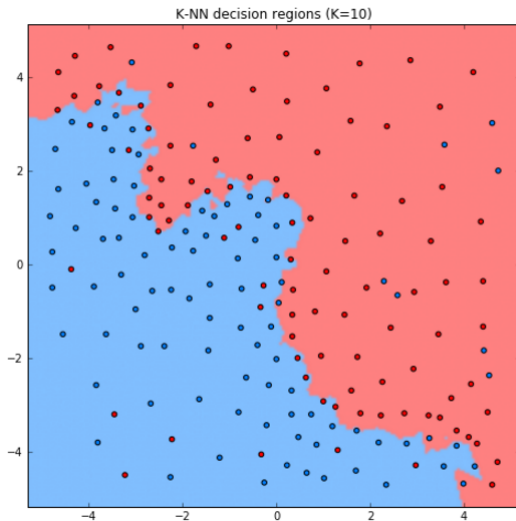
Демонстрационная выборка

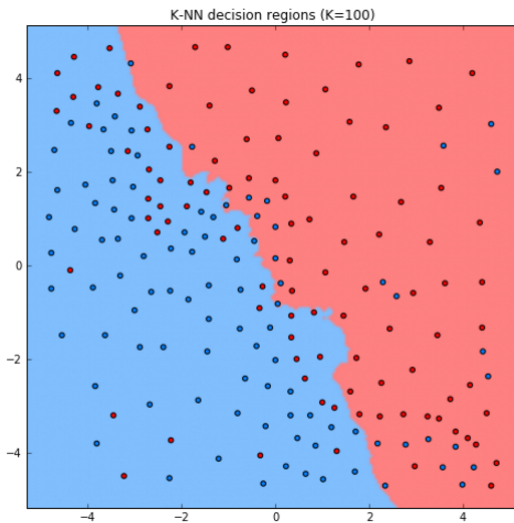


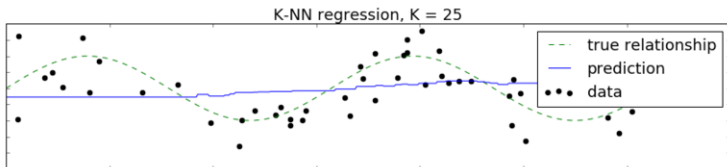
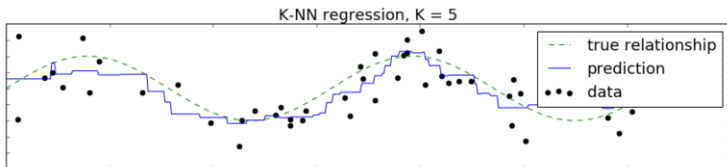
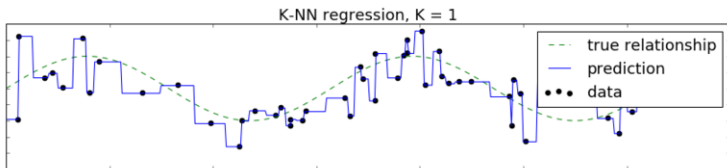
Пример: K-NN, $K=1$ 

Пример: К-NN, $K=3$ 

Пример: К-NN, $K=5$ 

Пример: К-NN, $K=10$ 

Пример: K-NN, $K=100$ 

Пример: K-NN, регрессия $y = \sin x + \varepsilon$ 

Если два класса равноценно побеждают

Если два класса набирают одинаковый рейтинг, можно сопоставить класс:

Если два класса равноценно побеждают

Если два класса набирают одинаковый рейтинг, можно сопоставить класс:

- случайно
- более распространенный в обучающей выборке
- имеющий ближайшего представителя:
 - из ближайших среди K соседей
 - ближайшего среднего представителя среди K соседей
 - из самых удаленных среди K соседей
(поощряем компактность распределения)

Параметры метода

- Параметры метода:
 - число ближайших соседей K
 - метрика расстояния $\rho(x, x')$
- Модификации:
 - возможный отказ, если прогноз неопределённый ¹
 - адаптивный выбор $K(x)$ ²

¹Предложите критерий неопределенности прогноза для классификации и для регрессии.

²Предложите, как именно $K(x)$ мог бы меняться в зависимости от локальной плотности объектов.

Свойства K-NN

- **Достоинства:**

- для прогноза нужна только степень близости между объектами, а не конкретные признаковые представления.
 - может применяться к объектам произвольно сложной структуры (тексты, графы, ...).
- легко реализовать
- интерпретируемый (прогноз по похожим известным случаям)
 - важно, например, в медицине.
- не требует обучения (нужно только сохранить объекты)
 - может применяться в онлайн-сценариях
 - кросс-валидация может заменяться скользящим контролем (leave-one-out).

- **Недостатки:**

- сложность прогноза $O(ND)$
- точность снижается с $\uparrow D$ ("проклятие размерности")

Пример использования для регрессии

```
...  
from sklearn.neighbors import KNeighborsRegressor  
from sklearn.metrics import mean_absolute_error  
  
X_train, X_test, Y_train, Y_test =  
    get_demo_regression_data()  
model = KNeighborsRegressor(n_neighbors=3) #  
    инициализация  
model.fit(X_train, Y_train) # обучение модели  
Y_hat = model.predict(X_test) # построение прогнозов  
print(f'Средний модуль ошибки (MAE): \n'  
{mean_absolute_error(Y_test, Y_hat):.2f}')
```

Больше информации. Полный код.

Пример использования

```
...
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import brier_score_loss

X_train, X_test, Y_train, Y_test =
    get_demo_classification_data()
model = KNeighborsClassifier(n_neighbors=3) # инициализация
model.fit(X_train, Y_train) # обучение модели
Y_hat = model.predict(X_test) # построение прогнозов
print(f'Точность прогнозов: {100*accuracy_score(Y_test,
    Y_hat):.1 f}%')

P_hat = model.predict_proba(X_test) # вероятности классов

# качество Бриера на вероятности положительного класса
loss = brier_score_loss(Y_test, P_hat[:,1])
print(f'Средняя ошибка прогноза вероятностей \
(по мере Бриера): {loss:.2 f}%')
```

Больше информации. Полный код.

Содержание

- 1 Метод ближайших центроидов
- 2 K ближайших соседей
- 3 Свойства K-NN**
- 4 Взвешенный учет объектов
- 5 Регрессия Надарая-Ватсона
- 6 Популярные функции расстояния

Зависимость от масштаба признаков

- Влияет ли масштабирование признаков на прогнозы K-NN?

Зависимость от масштаба признаков

- Влияет ли масштабирование признаков на прогнозы K-NN?
 - да, поэтому нужно нормализовывать их
- Единый масштаб \Rightarrow одинаковое влияние признаков
- Разный масштаб \Rightarrow различное влияние признаков.
 - Повышение масштаба увеличивает или уменьшает вклад признака в прогноз?

Распространенные нормализации признаков

Название	Преобразование	Выходные свойства
Стандартизация	$\frac{x_j - \mu_j}{\sigma_j}$	нулевое среднее и единичная дисперсия
Нормализация средним	$\frac{x_j - \mu_j}{\max(x_j) - \min(x_j)}$	нулевое среднее, с единичным диапазоном
Диапазонное шкалирование	$\frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)}$	принадлежит интервалу $[0, 1]$

- Какой тип шкалирования более устойчив к выбросам?
- Какой тип шкалирования сохраняет свойство разреженности? (много нулевых значений)

Пример использования

```
scaler_std = StandardScaler() # стандартизация
X_train_std = scaler_std.fit_transform(X_train)
X_test_std = scaler_std.transform(X_test)

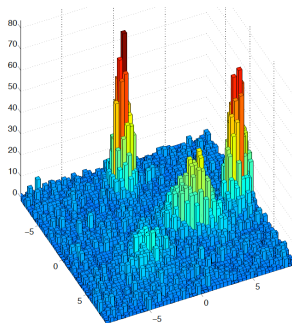
scaler_mm = MinMaxScaler() # диапазонное шкалирование
X_train_mm = scaler_mm.fit_transform(X_train)
X_test_mm = scaler_mm.transform(X_test)

scaler_rb = RobustScaler() # устойчивое к выбросам
# Вычитаем медиану и делим на разницу квантилей,
# например, Q(0.75)-Q(0.25)
X_train_rb = scaler_rb.fit_transform(X_train)
X_test_rb = scaler_rb.transform(X_test)
```

[Ссылка на полный код..](#)

Проклятие размерности: идея

- Проклятие размерности: $\uparrow D \Rightarrow$ близких точек становится мало.
 - за счет повышения объема пространства
- Пример: оценка гистограмм³



³С какой скоростью должно расти N , чтобы с точки зрения точности компенсировать увеличение D ?

Проклятие размерности: обоснование

- 1 Предположим, точки распределены равномерно в \mathbb{R}^D .
- 2 Шар радиуса R имеет объем $V(R) = CR^D$, где $C = \frac{\pi^{D/2}}{\Gamma(D/2+1)}$ ($2R$, πR^2 , $\frac{4}{3}\pi R^3$, ...).
- 3 Отношение объемов шаров радиуса $R - \varepsilon$ и R :

$$\frac{V(R - \varepsilon)}{V(R)} = \left(\frac{R - \varepsilon}{R}\right)^D \xrightarrow{D \rightarrow \infty} 0$$

- объем уменьшается вокруг центра (даже с малым ε) и концентрируется на поверхности.
- в K-NN: ближайшие соседи перестают быть близкими.
- в гистограмме: сложнее набрать сгусток точек в заданной окрестности.
- Хорошие новости: в практических задачах объекты распределены неравномерно, на многообразиях меньшей размерности.

Содержание

- 1 Метод ближайших центроидов
- 2 К ближайших соседей
- 3 Свойства K-NN
- 4 Взвешенный учет объектов**
- 5 Регрессия Надарая-Ватсона
- 6 Популярные функции расстояния

Равномерный учет объектов

- Обозначим K ближайших соседей к точке x :

$$(x_1, y_1), (x_2, y_2), \dots (x_K, y_K)$$

$$\rho(x, x_1) \leq \rho(x, x_2) \leq \dots \leq \rho(x, x_K)$$

- Регрессия:

$$\hat{y}(x) = \frac{1}{K} \sum_{k=1}^K y_k$$

- Классификация:

$$g_c(x) = \sum_{k=1}^K \mathbb{I}[y_k = c], \quad c = 1, 2, \dots, C.$$

$$\hat{y}(x) = \arg \max_c g_c(x)$$

Взвешенное голосование

- Взвешенная регрессия:

$$\hat{y}(x) = \frac{\sum_{k=1}^K w(k, \rho(x, x_k)) y_k}{\sum_{k=1}^K w(k, \rho(x, x_k))}$$

Взвешенное голосование

- Взвешенная регрессия:

$$\hat{y}(x) = \frac{\sum_{k=1}^K w(k, \rho(x, x_k)) y_k}{\sum_{k=1}^K w(k, \rho(x, x_k))}$$

- Взвешенная классификация:

$$g_c(x) = \sum_{k=1}^K w(k, \rho(x, x_k)) \mathbb{I}[y_k = c], \quad c = 1, 2, \dots, C.$$

$$\hat{y}(x) = \arg \max_c g_c(x)$$

Популярные варианты весов

Веса, зависящие от ранга близости:

$$w_k = \alpha^k, \quad \alpha \in (0, 1)$$

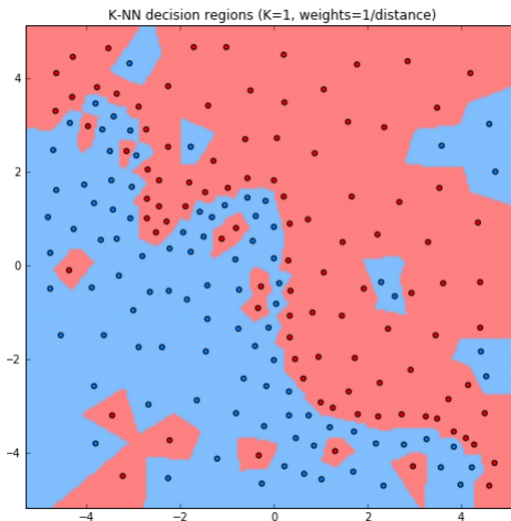
$$w_k = \frac{K + 1 - k}{K}$$

Веса, зависящие от расстояний до объектов:

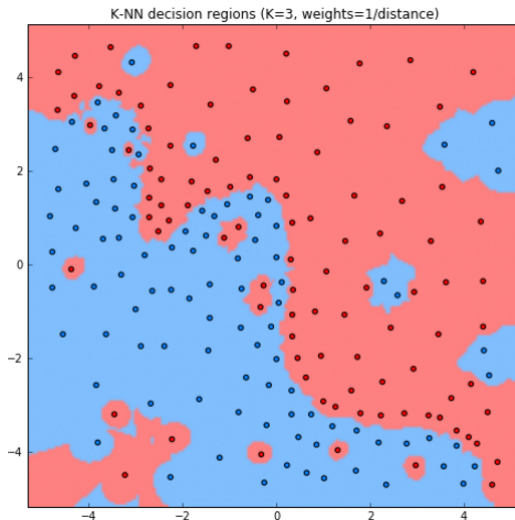
$$w_k = \begin{cases} \frac{\rho(z_K, x) - \rho(z_k, x)}{\rho(z_K, x) - \rho(z_1, x)}, & \rho(x_K, x) \neq \rho(x_1, x) \\ 1 & \rho(x_K, x) = \rho(x_1, x) \end{cases}$$

$$w_k = \frac{1}{\rho(x_k, x)}$$

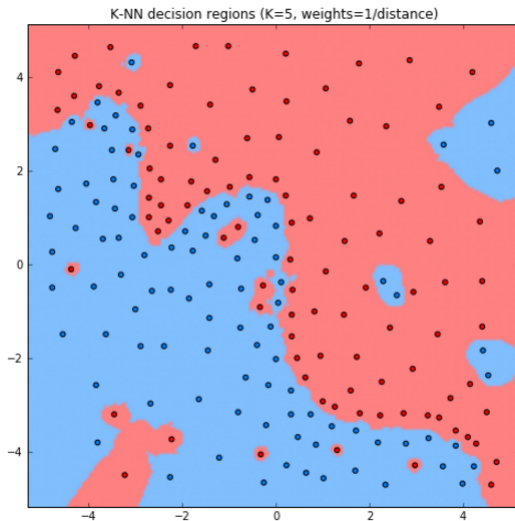
Пример: взвешенная классификация $K = 1$



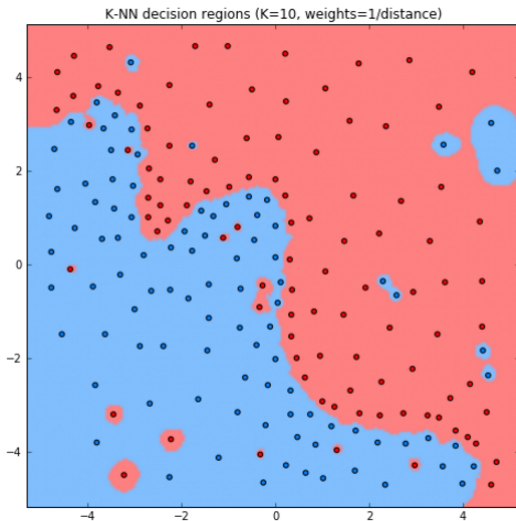
Пример: взвешенная классификация $K = 3$



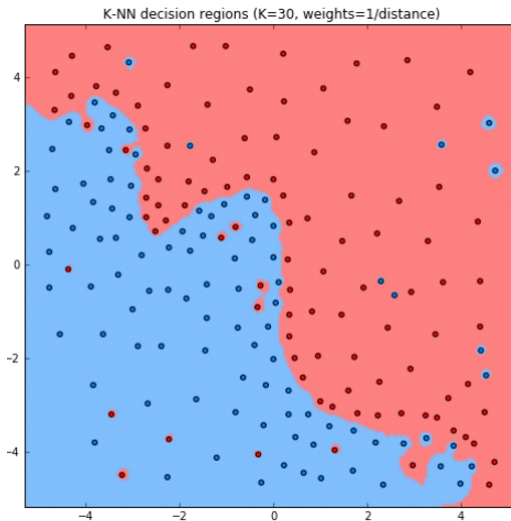
Пример: взвешенная классификация $K = 5$



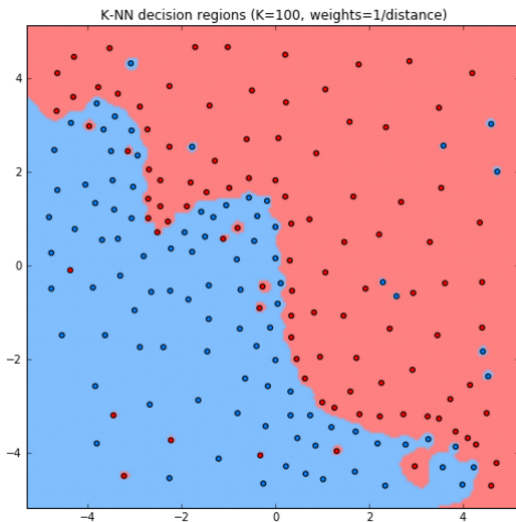
Пример: взвешенная классификация $K = 10$



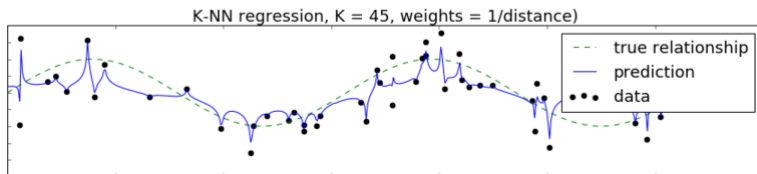
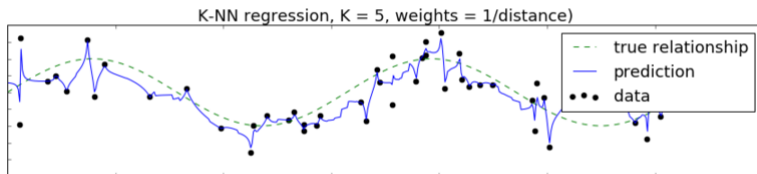
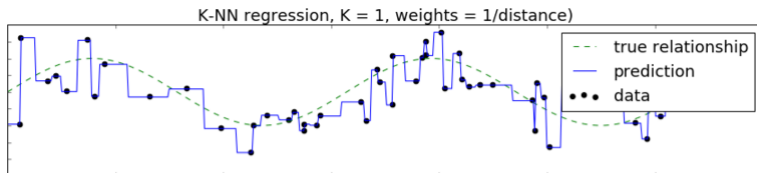
Пример: взвешенная классификация $K = 30$



Пример: взвешенная классификация $K = 100$



Пример: взвешенная регрессия K-NN



Содержание

- 1 Метод ближайших центроидов
- 2 К ближайших соседей
- 3 Свойства K-NN
- 4 Взвешенный учет объектов
- 5 Регрессия Надарая-Ватсона**
- 6 Популярные функции расстояния

Оптимальный константный прогноз

Найдем для обучающей выборки $(x_1, y_1), \dots, (x_N, y_N)$ оптимальный константный прогноз $\hat{y} \in \mathbb{R}$:

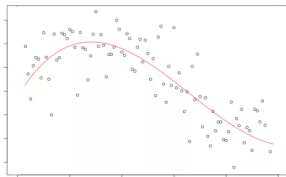
$$\hat{y} = \arg \min_{\hat{y} \in \mathbb{R}} \sum_{i=1}^N (\hat{y} - y_i)^2 = \frac{1}{N} \sum_{n=1}^N y_n$$

Оптимальный константный прогноз

Найдем для обучающей выборки $(x_1, y_1), \dots, (x_N, y_N)$ оптимальный константный прогноз $\hat{y} \in \mathbb{R}$:

$$\hat{y} = \arg \min_{\hat{y} \in \mathbb{R}} \sum_{i=1}^N (\hat{y} - y_i)^2 = \frac{1}{N} \sum_{n=1}^N y_n$$

Но нам нужно моделировать
нелинейные закономерности:

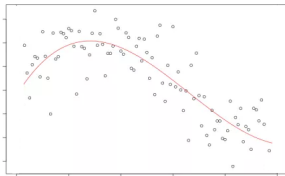


Оптимальный константный прогноз

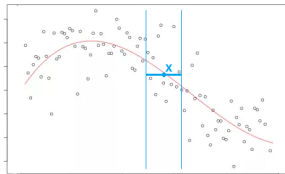
Найдем для обучающей выборки $(x_1, y_1), \dots, (x_N, y_N)$ оптимальный константный прогноз $\hat{y} \in \mathbb{R}$:

$$\hat{y} = \arg \min_{\hat{y} \in \mathbb{R}} \sum_{i=1}^N (\hat{y} - y_i)^2 = \frac{1}{N} \sum_{n=1}^N y_n$$

Но нам нужно моделировать нелинейные закономерности:



Регрессия Надарая-Ватсона - локальный константный прогноз.



Регрессия Надарая-Ватсона

- Найдем локальный оптимальный константный прогноз:

$$\hat{y}(x) = \arg \min_{\hat{y} \in \mathbb{R}} \sum_{i=1}^N w_i(x) (\hat{y} - y_i)^2 = \frac{\sum_{i=1}^N y_i w_i(x)}{\sum_{i=1}^N w_i(x)}$$

- Веса \downarrow при $\uparrow \rho(x, x_i)$ за счет убывающей $K(u)$ ("ядра"):

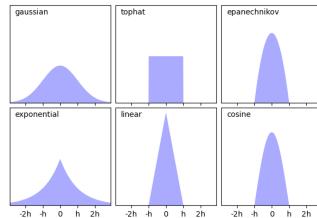
$$w_i(x) = K\left(\frac{\rho(x, x_i)}{h}\right)$$

- h - параметр "ширины окна"⁴
 - при $K(u) = \mathbb{I}[u \leq 1]$ решение зависит только от окрестности радиуса h вокруг x .
- Англ. local constant regression, kernel regression.

⁴Как он влияет на сложность модели?

Функция ядра

Ядро $K(u)$	Формула
top-hat	$\mathbb{I}[u < 1]$
линейное	$\max\{0, 1 - u \}$
Епанечникова	$\max\{0, 1 - u^2\}$
экспоненциальное	$e^{- u }$
Гауссово	$e^{-\frac{1}{2}u^2}$
квартичное	$(1 - u^2)^2 \mathbb{I}[u < 1]$



Комментарии

- Веса обеспечивают нелинейность прогноза, но требуют пересчета для каждого x .
- При достаточно общих условиях $\hat{y}(x) \xrightarrow{P} E[y|x]$
- Конкретный вид $K(u)$ не так важен для точности, как выбор h .
- Выбор $K(u)$ влияет на вычислительную сложность.
- Возможен динамический выбор $h(x)$.
 - $h(x)$ ниже, если локальная плотность точек - выше, например $h(x)$ - расстояние до K ближайшего соседа x .⁵

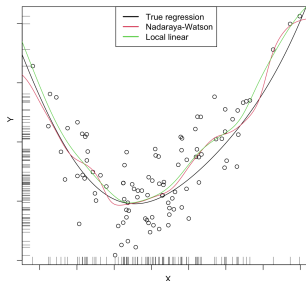
⁵При каком выборе $h(x)$ и $K(u)$ метод превращается в K-NN?

Локальная линейная регрессия

Вместо локальной константы можно оптимизировать локально линейную регрессию:

$$\sum_{i=1}^N w_i(x) (\mathbf{x}_i^T \boldsymbol{\beta} - y_i)^2 \rightarrow \min; \quad \hat{y}(x) = \mathbf{x}^T \boldsymbol{\beta}$$

Она устойчивее, лучше аппроксимирует области низкой плотности объектов, но вычислительно сложнее.



Содержание

- 1 Метод ближайших центроидов
- 2 К ближайших соседей
- 3 Свойства K-NN
- 4 Взвешенный учет объектов
- 5 Регрессия Надарая-Ватсона
- 6 Популярные функции расстояния**

Популярные функции расстояния⁶

Название	$\rho(x, z)$
Евклидова	$\sqrt{\sum_{i=1}^D (x^i - z^i)^2}$
L_p	$\sqrt[p]{\sum_{i=1}^D x^i - z^i ^p}$
L_∞	$\max_{i=1,2,\dots,D} x^i - z^i $
L_1	$\sum_{i=1}^D x^i - z^i $
Канберра	$\frac{1}{D} \sum_{i=1}^D \frac{ x^i - z^i }{ x^i + z^i }$
Ланса-Уильямса	$\frac{\sum_{i=1}^D x^i - z^i }{\sum_{i=1}^D x^i + z^i }$

Часто определяют меру близости $S(x, z)$, тогда $\rho(x, z) = K(S(x, z))$ для $\downarrow K$, например

$$\rho(x, z) = 1 - S(x, z) \quad \rho(x, z) = \frac{1}{S(x, z)}$$

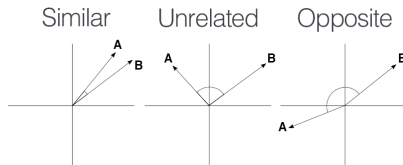
⁶Постройте единичные сферы по L_1, L_2, L_∞ метрикам.

Косинусная мера близости

- Косинусная мера близости: объекты близки, если угол между их векторами мал.

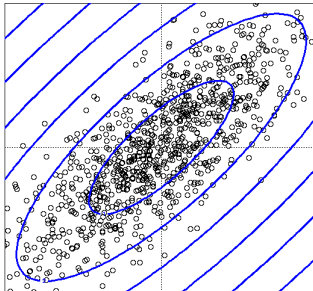
$$\text{sim}(x, z) = \frac{x^T z}{\|x\| \|z\|} = \frac{\sum_{i=1}^D x^i z^i}{\sqrt{\sum_{i=1}^D (x^i)^2} \sqrt{\sum_{i=1}^D (z^i)^2}}$$

- $\langle x, z \rangle = x^T z = \|x\| \|z\| \cos(\alpha)$, где α - угол между x и z .



- метрика $\in [-1, 1]$ и инвариантна к длинам $\|x\|$, $\|z\|$.
 - удобно для текстовых представлений в виде счетчиков слов.

Зависимые признаки



- Объекты вдоль оси $y = x$ более похожи, чем вдоль $y = -x$. Как это учесть?
- Посчитаем Евклидово расстояние, но для декоррелированных признаков.

Декоррелирующее преобразование

- $x \sim F(\mu, \Sigma)$, $\mu = \mathbb{E}[\mu]$, $\Sigma = \text{cov}(x, x)$, $\mu \in \mathbb{R}^D$, $\Sigma \in \mathbb{R}^{D \times D}$
- Декоррелирующее преобразование:

$$z = \Sigma^{-1/2}(x - \mu)$$

- Технически делается через спектральное разложение:
 - собственные вектора образуют ОНБ, т.к. $\Sigma = \Sigma^T$
 - собственные значение показывают растянутость данных вдоль осей собственных векторов.

$$\Sigma = Q\Lambda Q^T,$$
$$z = Q\Lambda^{-1/2}Q^T(x - \mu)$$

- Свойства⁷:

$$Ez = \mathbf{0}, \text{cov}[z, z] = I.$$

⁷Докажите.

Евклидово расстояние в декоррелированном пространстве

- Расстояние между x и x' = Евклидовому расстоянию в декоррелированном пространстве между $z = \Sigma^{-1/2}(x - \mu)$ и $z' = \Sigma^{-1/2}(x' - \mu)$:

$$\rho_M(x, x') = \rho_E(z, z') = \sqrt{(z - z')^T(z - z')} =$$

⁸Как расстояние упроститься для нескоррелированных признаков разной дисперсии? Проинтерпретируйте.

Евклидово расстояние в декоррелированном пространстве

- Расстояние между x и x' = Евклидовому расстоянию в декоррелированном пространстве между $z = \Sigma^{-1/2}(x - \mu)$ и $z' = \Sigma^{-1/2}(x' - \mu)$:

$$\begin{aligned}\rho_M(x, x') &= \rho_E(z, z') = \sqrt{(z - z')^T (z - z')} = \\ &= \sqrt{(\Sigma^{-1/2}(x - x'))^T \Sigma^{-1/2}(x - x')} \\ &= \sqrt{(x - x')^T \Sigma^{-1/2} \Sigma^{-1/2} (x - x')} \\ &= \sqrt{(x - x')^T \Sigma^{-1} (x - x')}\end{aligned}$$

- Это расстояние Махаланобиса⁸

⁸Как расстояние упроститься для нескоррелированных признаков разной дисперсии? Проинтерпретируйте.

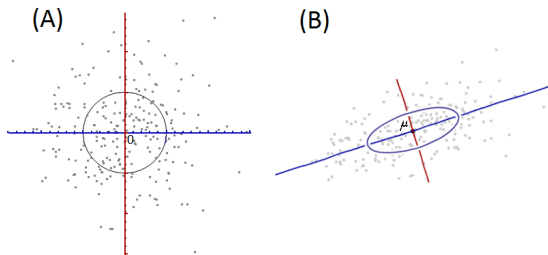
Идея расстояния Махаланобиса

Множество равноудаленных объектов от $(0,0)$

(A): в декоррелированном пространстве $\{z : \rho_E(z, 0)^2 = 1\}$.

(B): в исходном пространстве

$\{x : \rho_M(x, \mu)^2 = (x - \mu)^T \Sigma^{-1} (x - \mu) = 1\}^9$.



⁹ Докажите, что это действительно будет эллипс (используйте спектральное разложение и свойства Σ).

Обобщение расстояния Махаланобиса

- Заменяем Σ^{-1} на матрицу M :

$$\rho_M(x, x') = \sqrt{(x - x')^T M (x - x')}$$

- Нужна неотрицательность под корнем, поэтому ищем M в виде

$$M = LL^T$$

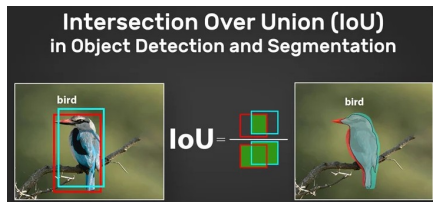
- Тогда $M = M^T$ и $M \succcurlyeq 0$.
- Выберем такую L , чтобы К-NN давал максимальную точность на кросс-валидации.
- Возможны и другие параметризации $\rho(x, x')$.
- Пример важности выбора расстояния:
 - классификация человека по фото
 - определение позы по фото

Функция близости между множествами

- Часто нужно сравнивать множества (корзины товаров, множества пикселей в задаче сегментации и детекции)
- Функция близости между множествами S_1, S_2 - близость Жаккарда (Jaccard coefficient):

$$\text{sim}(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}$$

- $\in [0, 1]$. 0-нет общих элементов, 1-одинаковые множества.



Более сложные типы данных

- Сравнение строк:
 - редакторское расстояние (edit distance, Levenstein distance): минимальное число правок для перевода одной строки в другую.

- каждому типу правки можно давать свой вес

i n t e n t i o n	← delete i
n t e n t i o n	← substitute n by e
e t e n t i o n	← substitute t by x
e x e n t i o n	← insert u
e x e n u t i o n	← substitute n by c
e x e c u t i o n	

- длина наибольшей общей подпоследовательности (longest common subsequence, используется в git):

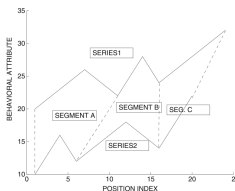
$$\text{MaxSubsequence}(ABCDE, XACYEZH V) = ACE$$

- Сравнение графов: редакторское расстояние и наибольший общий подграф.
- Реализуются эффективно методами динамического программирования.

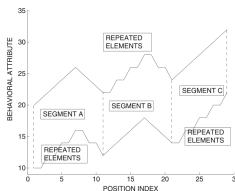
Сравнение временных рядов

Сравнение временных рядов

- Можно сравнивать спектры (коэффициенты разложения в ряде Фурье или др. базисе)
 - Пример: распознавание речи.
- Алгоритм динамической трансформации временной шкалы (dynamic time warping):
 - перед сравнением ищем оптимальное локальное растяжение рядов.



(a) Original series



(b) Warped series

Достоинства метрических методов

Метрические методы стоит использовать, если

- N мало
- важна быстрая подстройка к новым данным
- важна интерпретируемость
- есть разумная $\rho(x, x')$, отвечающая y
- сложно придумать признаки, но есть $\rho(x, x')$
 - тексты, графы (editor distance), временные ряды (dynamic time warping)
- есть много классов, но мало примеров для каждого
 - сложно выучивать свою $g_c(x)$ для каждого класса

Недостатки метрических методов

Метрические методы не стоит использовать, если

- D велико
 - $\rho(x, x')$ долго считается
 - проклятие размерности
 - проблема решается \downarrow размерности пространства (feature selection, PCA)
- N велико
 - сложность прогнозов $O(N)$
 - проблема решается
 - фильтрацией неинформативных объектов (prototype selection)
 - упорядочиванием объектов в пространстве (KD-tree, Ball-tree, MinHash)

Заключение

- Масштаб признаков влияет на прогноз.
- Проклятие размерности ухудшает качество локальных метрических методов.
- Метод ближайших центроидов - простая базовая модель.
- Метод К ближайших соседей:
 - K контролирует сложность модели
 - $\rho(x, x')$: выбирается из смысла задачи
 - быстрое обучение, медленный прогноз (возможны ускорения)
 - взвешенный учет соседей
- Регрессия Надарая-Ватсона
 - h контролирует сложность модели
 - обобщение: локальная линейная регрессия