# Машинное обучение, ФКН ВШЭ
# Семинар №7

## 1 k ближайших соседей

Рассмотрим задачу классификации: $\mathbb{Y} = \{1, \dots, K\}$. Пусть дана обучающая выборка $X = (x_i, y_i)_{i=1}^{\ell}$ и функция расстояния $\rho : \mathbb{X} \times \mathbb{X} \to [0, \infty)$. Мы не будем требовать, чтобы функция расстояния являлась метрикой — достаточно, чтобы она была симметричной и неотрицательной. Не будем строить модель на этапе обучения, а вместо этого просто запомним обучающую выборку. Пусть теперь требуется классифицировать новый объект $u$. Расположим объекты обучающей выборки $X$ в порядке неубывания расстояний до $u$:

$$\rho(u, x_u^{(1)}) \leqslant \rho(u, x_u^{(2)}) \leqslant \ldots \leqslant \rho(u, x_u^{(\ell)}),$$

где через $x_u^{(i)}$ обозначается $i$-й сосед объекта $u$. Алгоритм $k$ *ближайших соседей* ($k$ nearest neighbours, kNN) относит объект $u$ к тому классу, представителей которого окажется больше всего среди $k$ его ближайших соседей:

$$a(u) = \arg\max_{y \in \mathbb{Y}} \sum_{i=1}^{k} [y_u^{(i)} = y]. \tag{1.1}$$

### §1.1 Метод парзеновского окна

Проблема формулы (1.1) состоит в том, что она никак не учитывает расстояния до соседей. Действительно, если рассматривается 7 ближайших соседей, и для объекта $u$ ближайшие два объекта находятся на расстоянии $\rho(u, x) \approx 2$, а остальные — на расстоянии $\rho(u, x) \geqslant 100$, то было бы логично обращать внимание только на первые два объекта. Чтобы добиться этого, можно ввести веса в модель:

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

Веса можно делать затухающими по мере роста номера соседа $i$ (например, $w(i, u, x) = \frac{k+1-i}{k}$), но лучше использовать расстояния при их вычислении. Это делается в *методе парзеновского окна*:

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

где $K$ — это ядро, $h$ — ширина окна.

## §1.2 Ядровая регрессия

В методе $k$ ближайших соседей мы, по сути, максимизировали взвешенную долю правильных ответов среди соседей при предсказании константой $y$. Иными словами, мы старались в каждой точке пространства выбрать такое значение модели, которое было бы оптимально для некоторой окрестности этой точки.

Попробуем воспользоваться этим соображением, чтобы обобщить подход на задачу регрессии. Будем выбирать в каждой точке такой ответ, который лучшим образом приближает целевую переменную для $k$ ближайших соседей:

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

Можно в явном виде выписать решение оптимизационной задачи:

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

Данную модель иногда называют формулой Надарая-Ватсона.

## §1.3 Некоторые проблемы метода

Разберем особенности и проблемы метода $k$ ближайших соседей, возникающие при использовании евклидовой метрики в качестве функции расстояния:

$$\rho(x, z) = \left(\sum_{j=1}^{d} |x_j - z_j|^2\right)^{1/2}.$$

### 1.3.1 Шумовые признаки

**Задача 1.1.** *Рассмотрим задачу с одним признаком и двумя объектами обучающей выборки: $x_1 = 0.1$, $x_2 = 0.5$. Первый объект относится к первому классу, второй — ко второму. Добавим к объектам шумовой признак, распределенный равномерно на отрезке $[0, 1]$. Пусть требуется классифицировать новый объект $u = (0, 0)$. Какова вероятность, что после добавления шума второй объект окажется к нему ближе, чем первый?*

**Решение.** Задача сводится к вычислению вероятности $\mathbb{P}(0.5^2 + \xi_2^2 \leqslant 0.1^2 + \xi_1^2)$, где $\xi_1$ и $\xi_2$ — независимые случайные величины, распределенные равномерно на $[0, 1]$. Вычислим ее:

$$\mathbb{P}(0.5^2 + \xi_2^2 \leqslant 0.1^2 + \xi_1^2) = \mathbb{P}(\xi_1^2 \geqslant 0.24 + \xi_2^2) =$$
$$= \int_0^{\sqrt{0.76}} \int_{\sqrt{x_2^2+0.24}}^{1} dx_1 dx_2 = \int_0^{\sqrt{0.76}} \left(1 - \sqrt{x_2^2 + 0.24}\right) dx_2 \approx 0.275.$$

■

Таким образом, шумовые признаки могут оказать сильное влияние на метрику. Обнаружить шумовые признаки можно, удаляя поочередно все признаки и смотря на ошибку на тестовой выборке или ошибку кросс-валидации.

### 1.3.2 «Проклятие размерности»

Пусть объекты выборки — это точки, равномерно распределенные в $d$-мерном кубе $[0,1]^d$. Рассмотрим выборку, состоящую из 5000 объектов, и применим алгоритм пяти ближайших соседей для классификации объекта $u$, находящегося в начале координат. Выясним, на сколько нужно отступить от этого объекта, чтобы с большой вероятностью встретить пять объектов выборки. Для этого построим подкуб $[0;\varepsilon]^d \subset [0;1]^d$, $\varepsilon \in (0,1)$, и положим его объём равным $\delta = \varepsilon^d$. Найдем такое значение $\delta$, при котором в этот подкуб попадет как минимум пять объектов выборки с вероятностью 0.95.

**Задача 1.2.** *Запишите выражение для $\delta$.*

**Решение.**

$$\min\left\{\delta \,|\, \sum_{k=5}^{5000} \binom{5000}{k} \delta^k (1-\delta)^{5000-k} \geqslant 0.95\right\} \approx 0.0018.$$

■

Таким образом, для того, чтобы найти пять соседей объекта $u$, нужно по каждой координате отступить на $0.0018^{1/d}$. Уже при $d = 10$ получаем, что нужно отступить на 0.53, при $d = 100$ — на 0.94. Таким образом, при больших размерностях объекты становятся сильно удалены друг от друга, из-за чего классификация на основе сходства объектов может потерять смысл. В то же время отметим, что в рассмотренном примере признаки объектов представляли собой равномерный шум, тогда как в реальных задачах объекты могут иметь осмысленные распределения, позволяющие построение модели классификации даже при больших размерностях.

## §1.4 Примеры функций расстояния

### 1.4.1 Метрика Минковского

Метрика Минковского определяется как:

$$\rho_p(x,z) = \left(\sum_{j=1}^{d} |x_j - z_j|^p\right)^{1/p}$$

для $p \geqslant 1$. При $p \in (0,1)$ данная функция метрикой не является, но все равно может использоваться как мера расстояния.

Частными случаями данной метрики являются:

- Евклидова метрика ($p = 2$). Задает расстояние как длину отрезка прямой, соединяющей заданные точки.
- Манхэттенское расстояние ($p = 1$). Минимальная длина пути из $x$ в $z$ при условии, что можно двигаться только параллельно осям координат.
- Метрика Чебышева ($p = \infty$), выбирающая наибольшее из расстояний между векторами по каждой координате:

$$\rho_\infty(x,z) = \max_{j=1,\ldots,d} |x_j - z_j|.$$

- «Считающее» расстояние ($p = 0$), равное числу координат, по которым векторы $x$ и $z$ различаются:

$$\rho_0(x, z) = \sum_{j=1}^{d} [x_j \neq z_j].$$

Отметим, что считающее расстояние не является метрикой.

Отметим, что по мере увеличения параметра $p$ метрика слабее штрафует небольшие различия между векторами и сильнее штрафует значительные различия.

В случае, если признаки неравнозначны, используют взвешенное расстояние:

$$\rho_p(x, z; w) = \left( \sum_{j=1}^{d} w_j |x_j - z_j|^p \right)^{1/p}, \quad w_j \geqslant 0.$$

**Задача 1.3.** *Рассмотрим функцию $f(x) = \rho_2(x, 0; w)$. Что представляют из себя линии уровня такой функции?*

**Решение.** Распишем квадрат функции $f(x)$ (форма линий уровня от этого не изменится):

$$f^2(x) = \sum_{j=1}^{d} w_j x_j^2.$$

Сделаем замену $x_j = \frac{x'_j}{\sqrt{w_j}}$:

$$f^2(x') = \sum_{j=1}^{d} {x'_j}^2.$$

В новых координатах линии уровня функции расстояния представляют собой окружности с центром в нуле. Сама же замена представляет собой растяжение вдоль каждой из координат, поэтому в исходных координатах линия уровня являются эллипсами, длины полуосей которых пропорциональны $\sqrt{w_j}$.

■

Вывод: благодаря весам линии уровня можно сделать эллипсами с осями, параллельными осям координат. Это может быть полезно, если признаки имеют разные масштабы — благодаря весам автоматически будет сделана нормировка.

Веса можно брать, например, равными корреляции между признаком и целевым вектором:

$$w_j = \left| \frac{\sum_{i=1}^{\ell} x_{ij} y_i}{\left( \sum_{i=1}^{\ell} x_{ij}^2 \right)^{1/2} \left( \sum_{i=1}^{\ell} y_i^2 \right)^{1/2}} \right|.$$

Однако, лучше всего настраивать веса под обучающую выборку с помощью покоординатного спуска или другого метода оптимизации.

### 1.4.2 Расстояние Махалонобиса

*(данный материал является опциональным)*

Расстояние Махалонобиса определяется следующим образом:

$$\rho(x,z) = \sqrt{(x-z)^T S^{-1}(x-z)},$$

где $S$ — симметричная положительно определенная матрица.

**Задача 1.4.** *Что представляют из себя линии уровня функции* $f(x) = \rho(x,0)$?

**Решение.** Поскольку матрица S — симметричная, то из её собственных векторов можно составить ортонормированный базис. Рассмотрим матрицу $Q$, столбцами которой являются элементы данного базиса. Заметим, что она является ортогональной (в силу ортонормированности базиса), т.е. $Q^TQ = I$, $Q^{-1} = Q^T$, а потому выполняются следующие соотношения:

$$SQ = Q\Lambda \Rightarrow \Lambda = Q^{-1}SQ,$$

где $\Lambda$ — диагональная матрица, в которой записаны соответствующие собственные значения матрицы $S$.

Распишем квадрат функции $f(x)$, сделав замену $x' = Q^Tx$:

$$f^2(x) = \rho^2(x,0) = x^TS^{-1}x = x'^TQ^TS^{-1}Qx' = x'^T\left(Q^{-1}SQ\right)^{-1}x' =$$
$$= x'^T\Lambda^{-1}x' = \sum_{j=1}^{d}\frac{x'^2}{\lambda_j}.$$

Получаем, что линии уровня в новых координатах представляют собой эллипсы с осями, параллельными осям координат, причем длины полуосей равны корням из собственных значений матрицы $S$. При этом замена $x' = Q^Tx$ соответствует такому повороту осей координат, что координатные оси совпадают со столбцами матрицы $Q$. Таким образом, расстояние Махалонобиса позволяет получить линии уровня в виде произвольно ориентированных эллипсов.

■

Матрицу $S$ можно настраивать либо по кросс-валидации, либо брать равной выборочной ковариационной матрице: $\hat{S} = \frac{1}{n-1}X^TX$.

### 1.4.3 Расстояния между текстами

Пусть заданы векторы $x$ и $z$. Известно, что их скалярное произведение и косинус угла $\theta$ между ними связаны следующим соотношением:

$$\langle x,z\rangle = \|x\|\|z\|\cos\theta.$$

Соответственно, косинусное расстояние определяется как

$$\rho_{\cos}(x,z) = \arccos\left(\frac{\langle x,z\rangle}{\|x\|\|z\|}\right) = \arccos\left(\frac{\sum_{j=1}^{d}x_jz_j}{\left(\sum_{j=1}^{d}x_j^2\right)^{1/2}\left(\sum_{j=1}^{d}z_j^2\right)^{1/2}}\right).$$

Косинусная мера часто используется для измерения схожести между текстами. Каждый документ описывается вектором, каждая компонента которого соответствует слову из словаря. Компонента равна единице, если соответствующее слово встречается в тексте, и нулю в противном случае. Тогда косинус между двумя векторами будет тем больше, чем больше слов встречаются в этих двух документах одновременно.

Один из плюсов косинусной меры состоит в том, что в ней производится нормировка на длины векторов. Благодаря этому она не зависит, например, от размеров сравниваемых текстов, измеряя лишь объем их схожести.

### 1.4.4 Расстояние Джаккарда

Выше мы рассматривали различные функции расстояния для случая, когда объекты обучающей выборки являются вещественными векторами. Если же объектами являются множества (например, каждый объект — это текст, представленный множеством слов), то их сходство можно измерять с помощью *расстояния Джаккарда*:

$$\rho_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}.$$

**Задача 1.5.** *Пусть все множества являются подмножествами некоторого конечного упорядоченного множества $U = \{u_1, \ldots, u_N\}$. Тогда любое множество $A$ можно представить в виде бинарного вектора длины $N$, в котором единица в $i$-й позиции стоит тогда и только тогда, когда $u_i \in A$. Запишите формулу для расстояния Джаккарда, исходя из таких обозначений, и сравните ее с формулой для косинусной меры.*

**Решение.** Пусть $X$ и $Y$ — два множества, $(x_j)_{j=1}^N$ и $(y_j)_{j=1}^N$ — их векторные представления. Тогда мощность их пересечения можно записать следующим образом:

$$|X \cap Y| = \sum_{j=1}^{N} x_j y_j = \langle X, Y \rangle,$$

а мощность их объединения как

$$\begin{aligned} |X \cup Y| &= \sum_{j=1}^{N} x_j + \sum_{j=1}^{N} y_j - \sum_{j=1}^{N} x_j y_j = \\ &= \sum_{j=1}^{N} x_j^2 + \sum_{j=1}^{N} y_j^2 - \sum_{j=1}^{N} x_j y_j = \\ &= \|X\|^2 + \|Y\|^2 - \langle X, Y \rangle. \end{aligned}$$

Тогда:

$$\rho_J(X, Y) = 1 - \frac{\langle X, Y \rangle}{\|X\|^2 + \|Y\|^2 - \langle X, Y \rangle}.$$

■

### 1.4.5 Редакторское расстояние

Для измерения сходства между двумя строками (например, последовательностями ДНК) можно использовать *редакторское расстояние*, которое равно минимальному числу вставок и удалений символов, с помощью которых можно преобразовать первую строку ко второй. В зависимости от специфики задачи можно также разрешать замены, перестановки соседних символов и прочие операции.

### 1.4.6 Функции расстояния на категориальных признаках

Категориальные признаки не имеют никакой явной структуры, и поэтому достаточно сложно ввести на них разумное расстояние. Как правило, ограничиваются сравнением их значений: если у двух объектов одинаковые значения категориального признака, то расстояние равно нулю, если разные — единице. Тем не менее, существуют определенные соображения по поводу того, как измерять сходство для таких признаков.

Будем считать, что метрика записывается как взвешенная сумма расстояний по отдельным признакам с некоторыми весами:

$$\rho(x, z) = \sum_{j=1}^{d} w_j \rho_j(x_j, z_j).$$

Способы измерения расстояния для вещественных признаков обсуждались выше. Обсудим некоторые варианты для категориальных признаков. Введем следующие обозначения:

1. $f_j(m) = \sum_{i=1}^{\ell} [x_{ij} = m]$ — количество раз, которое $j$-й признак принимает значение $m$ на обучающей выборке;
2. $p_j(m) = \frac{f_j(m)}{\ell}$ — частота категории $m$ на обучающей выборке;
3. $p_j^{(2)}(m) = \frac{f_j(m)(f_j(m)-1)}{\ell(\ell-1)}$ — оценка вероятности того, что у двух случайно выбранных различных объектов из обучающей выборки значения признака будут равны $m$.

Тогда расстояние на категориальных признаках можно задавать, например, одним из следующих способов:

1. Индикатор совпадения:

$$\rho_j(x_j, z_j) = [x_j \neq z_j]$$

2. Сглаженный индикатор совпадения. Чем выше частота у значения признака, тем больше расстояние (если оба человека живут в Москве, то эта информация не очень важна, поскольку вероятность такого совпадения высока; если оба человека живут в Снежинске, то это важная информация, так событие является достаточно редким):

$$\rho_j(x_j, z_j) = [x_j \neq z_j] + [x_j = z_j] \sum_{q: p_j(q) \leqslant p_j(x_j)} p_j^{(2)}(q)$$

3. Чем более частые значения оказались при несовпадении, тем больше расстояние (если оба человека из разных, но очень маленьких городов, то можно считать их похожими; если один человек из Москвы, а второй — из Питера, то они сильно отличаются):

$$\rho_j(x_j, z_j) = [x_j \neq z_j] \log f_j(x_j) \log f_j(z_j)$$

(обратите внимание, что для борьбы с численными проблемами имеет смысл добавлять единицу под логарифмом: $\log(f_j(x_j) + 1)$, $\log(f_j(z_j) + 1)$)

# 2 Методы поиска ближайших соседей

*(данный материал является опциональным)*

## §2.1 Точные методы

Разберем методы поиска ближайших соседей для евклидовой метрики. Будем рассматривать задачу поиска одного ближайшего соседа, все методы несложно обобщаются на случай с $k > 1$.

Если просто перебирать все объекты обучающей выборки, выбирая наиболее близкий к новому объекту, то получаем сложность $O(\ell d)$.

Можно выбрать подмножество признаков, и сначала вычислить расстояние только по этим координатам. Оно является нижней оценкой на полноценное расстояние, и если оно уже больше, чем текущий наилучший результат, то данный объект можно больше не рассматривать в качестве кандидата в ближайшего соседа. Такой подход является чисто эвристическим и не гарантирует сублинейной сложности по размеру обучения.

**kd-деревья.** Одной из структур данных, позволяющих эффективно искать ближайших соседей к заданной точке, является *kd-дерево*. Оно разбивает пространство на области (каждая вершина производит разбиение по определенной координате), и каждый лист соответствует одному объекту из обучающей выборки. Обходя это дерево определенным образом, можно найти точку из обучения, ближайшую к заданной. Если размерность пространства небольшая (10-20), то данный подход позволяет находить ближайшего соседа за время порядка $O(\log \ell)$.

Экспериментально было установлено, что в пространствах большой размерности сложность поиска ближайшего соседа в kd-дереве сильно ухудшается и приобретает линейный порядок сложности [**?**].

## §2.2 Приближенные методы

Есть два способа борьбы с высокой сложностью поиска ближайших соседей при большом числе признаков:

1. Запоминать не всю обучающую выборку, а лишь ее представительное подмножество. Существует большое число эвристических алгоритмов для отбора эталонных объектов (например, STOLP).

2. Искать $k$ ближайших соседей приближенно, то есть есть разрешать результату поиска быть чуть дальше от нового объекта, чем $k$ его истинных соседей. Ниже мы подробно разберем этот подход.

Опишем метод приближенного поиска ближайших соседей LSH (locality-sensitive hashing). Его идея заключается в построении такой хэш-функции для объектов выборки, которая с большой вероятностью присваивает одинаковые значения близким объектам и разные значения отдаленным объектам. Дадим формальное определение.

**Опр. 2.1.** *Семейство функций $\mathcal{F}$ называется $(d_1, d_2, p_1, p_2)$-чувствительным, если для всех $x, y \in \mathbb{X}$ выполнено:*

- *Если $\rho(x, y) \leqslant d_1$, то $\mathbb{P}_{f \in \mathcal{F}}[f(x) = f(y)] \geqslant p_1$.*
- *Если $\rho(x, y) \geqslant d_2$, то $\mathbb{P}_{f \in \mathcal{F}}[f(x) = f(y)] \leqslant p_2$.*

*Здесь под вероятностью $\mathbb{P}_{f \in \mathcal{F}}$ понимается равномерное распределение на всех функциях семейства $\mathcal{F}$.*

Отметим, что определение имеет смысл лишь если $d_1 \leqslant d_2$ и $p_1 \geqslant p_2$.

**Пример.** Рассмотрим пример семейства хэш-функций для меры Джаккарда, которое носит название MinHash. Пусть объекты представляют собой множества, являющиеся подмножествами универсального упорядоченного множества $U = \{u_1, \ldots, u_n\}$. Выберем перестановку $\pi$ на элементах этого множества, и определим хэш-функцию $f_\pi(A)$ так, чтобы она возвращала номер первого элемента в данной перестановке, входящего в $A$:

$$f_\pi(A) = \min\{\pi(i) \,|\, u_i \in A\}.$$

Это преобразование можно интерпретировать следующим образом. Будем считать, что элементы наших множеств — это слова. Перестановка $\pi$ задает *степени важности* слов (чем меньше $\pi(i)$, тем важнее $i$-е слово). Например, если мы решаем задачу классификации текстов на научные и ненаучные, то предлоги и союзы должны иметь малый уровень важности, а слова «аннотация», «бустинг», «переобучение» — высокий уровень важности, поскольку их наличие свидетельствует о научности текста. Описанная хэш-функция возвращает для документа уровень важности самого важного слова в нем — в нашем примере это означает, что мы находим самое «научное» слово в тексте, и характеризуем документ именно этим словом.

Покажем, что множество всех MinHash-функций $\mathcal{F} = \{f_\pi \,|\, \pi \in \mathrm{Sym}(U)\}$ является $(d_1, d_2, p_1, p_2)$-чувствительным.

Сначала докажем следующее утверждение: вероятность того, что случайно выбранная функция $f_\pi \in \mathcal{F}$ будет принимать одинаковые значения на двух заданных множествах $A$ и $B$, равна коэффициенту Джаккарда $\frac{|A \cap B|}{|A \cup B|} = 1 - \rho_J(A, B)$ этих двух множеств. Разобьем элементы $u$ универсального множества $U$ на три типа:

1. $u \in A$, $u \in B$.
2. $u \in A$, $u \notin B$ или $u \notin A$, $u \in B$.

3. $u \notin A$, $u \notin B$.

Обозначим число объектов первого типа через $p$, а число объектов второго типа — через $q$. Заметим, что через $p$ и $q$ можно выразить коэффициент Джаккарда для множеств $A$ и $B$: $1 - \rho_J(A, B) = \frac{p}{p+q}$.

Вероятность того, что значения случайно выбранной хэш-функции будут одинаковыми на множествах $A$ и $B$, равна вероятности того, что в случайно выбранной перестановке множества $U$ элемент первого типа встретится раньше элемента второго типа; элементы третьего типа на значение хэш-функции никак не влияют. Последняя же вероятность равна $\frac{p}{p+q}$. Утверждение доказано.

Пусть расстояние Джаккарда между двумя множествами $\rho_J(A, B)$ не превосходит $d_1$. Тогда для коэффициента Джаккарда выполнено $1 - \rho_J(A, B) \geqslant 1 - d_1$, а значит, для вероятности $p_1$ того, что случайно выбранная функция из $\mathcal{F}$ даст одинаковые хэши для этих множеств, выполнено $p_1 \geqslant 1 - d_1$. Отсюда получаем, что $\mathcal{F}$ является $(d_1, d_2, 1 - d_1, 1 - d_2)$-чувствительным семейством.