Метрические алгоритмы и SVM

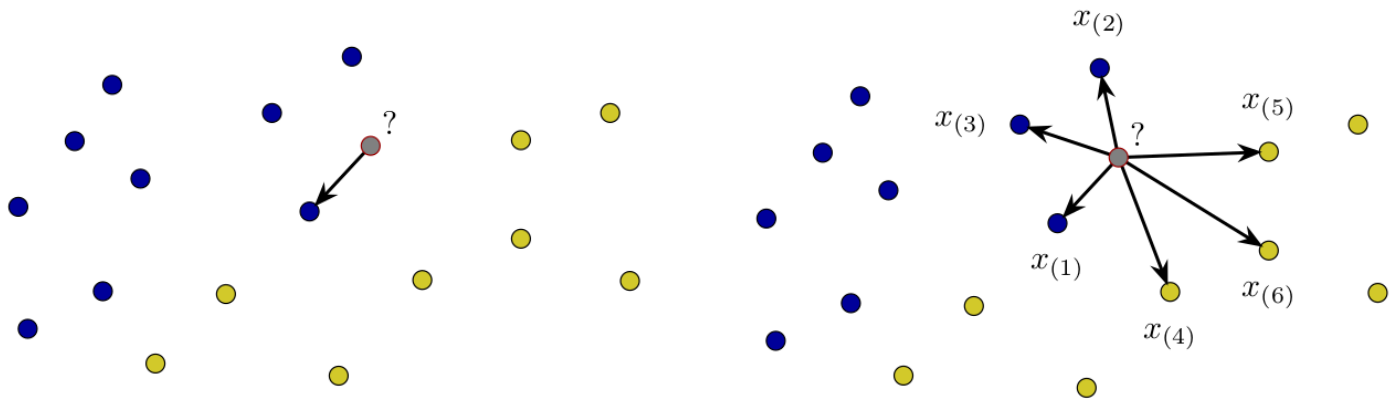
Методы анализа данных

Лекция 6

Москва, МФТИ, 2020

Метод ближайших соседей (kNN)

Метод ближайших соседей: объект относится к тому классу, к которому принадлежит большинство из его k ближайших соседей. Еще больше повысить надежность можно правильным образом определив веса в методе ближайших соседей. Веса могут зависеть как от номера соседа $w(x(i)) = w(i)$, так и от расстояния до него $w(x(i)) = w(d(x, x(i)))$.



Метод ближайших соседей (kNN)

Во взвешенном *kNN* объект x относится к тому классу, взвешенная сумма по объектам из множества k ближайших соседей для которого больше:

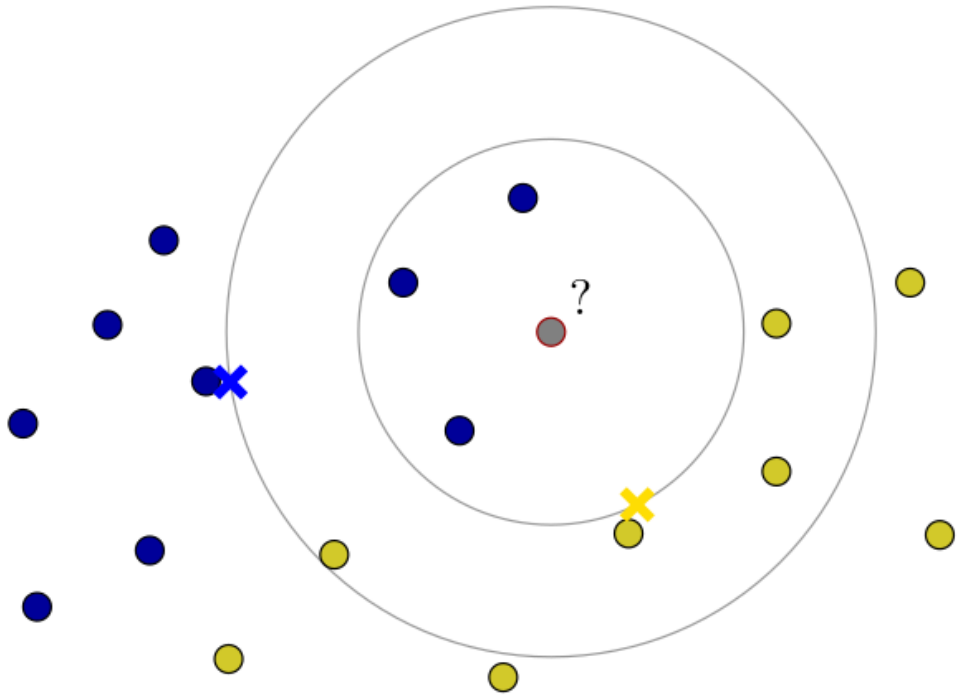
$$a(x) = \text{argmax}_{y \in Y} \sum_{i=1}^{k} [x_{(i)} = y] w(x_{(i)}).$$

Центроидный классификатор

По обучающей выборке $(x_i, y_i)_{i=1}^m$ определяются «центры» всех классов (ℓ_y — количество объектов класса y):

$$\mu_y = \frac{1}{\ell_y} \sum_{i:y_i=y} x_i.$$

Центроидный классификатор



После этого центроидный классификатор относит каждый новый объект x к тому классу, центр которого находится ближе всего в пространстве признаков к признаковому описанию нового объекта:

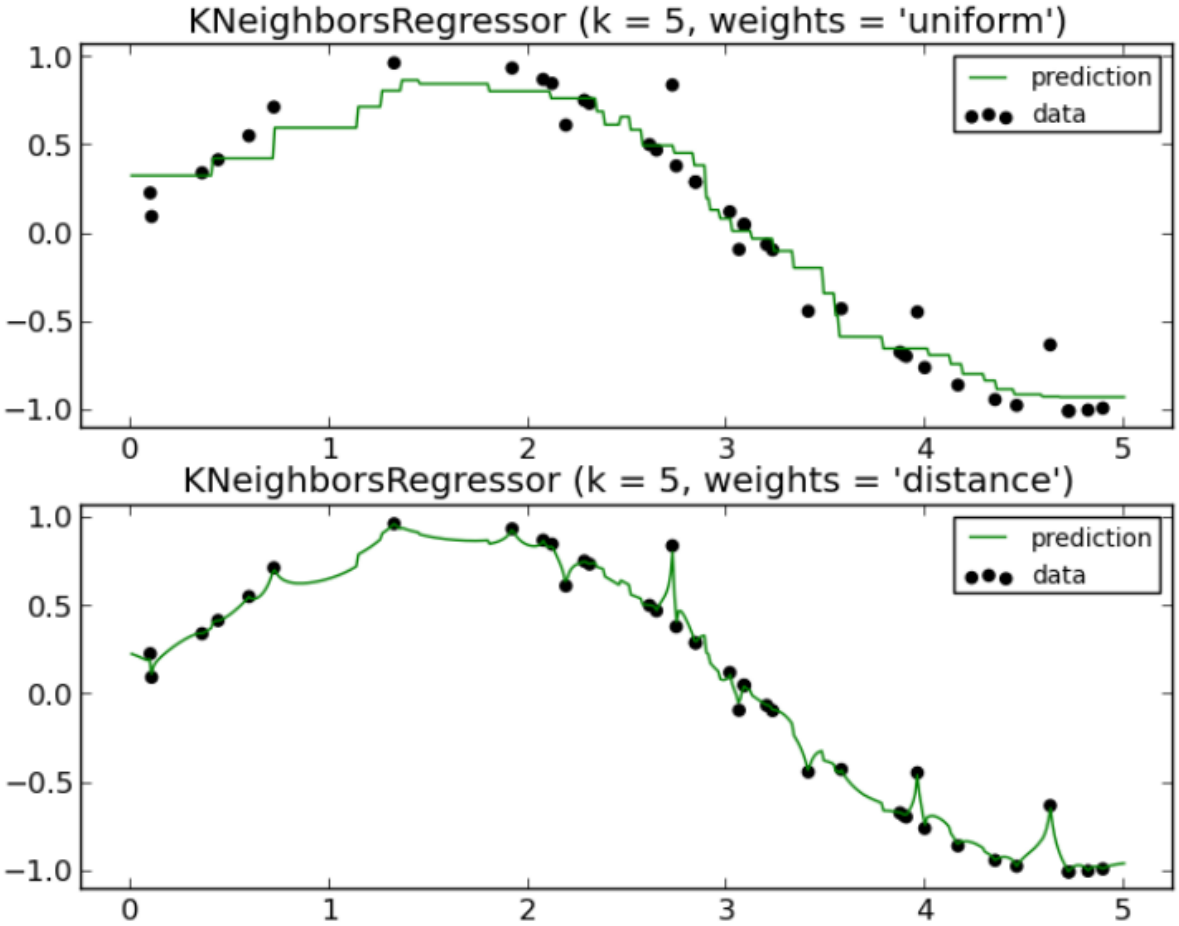
$$a(x) = \operatorname{argmin}_{y \in Y} d(\mu_y, x).$$

Взвешенный kNN для регрессии

Пусть x — новый объект, который требуется классифицировать, а $x_{(i)}$ — i-ый ближайший сосед из обучающей выборки. Взвешенный kNN для задачи регрессии в таком случае определяется выражением:

$$a(x) = \frac{\sum_{i=1}^{k} w(x_{(i)}) x_{(i)}}{\sum_{i=1}^{k} w(x_{(i)})}.$$

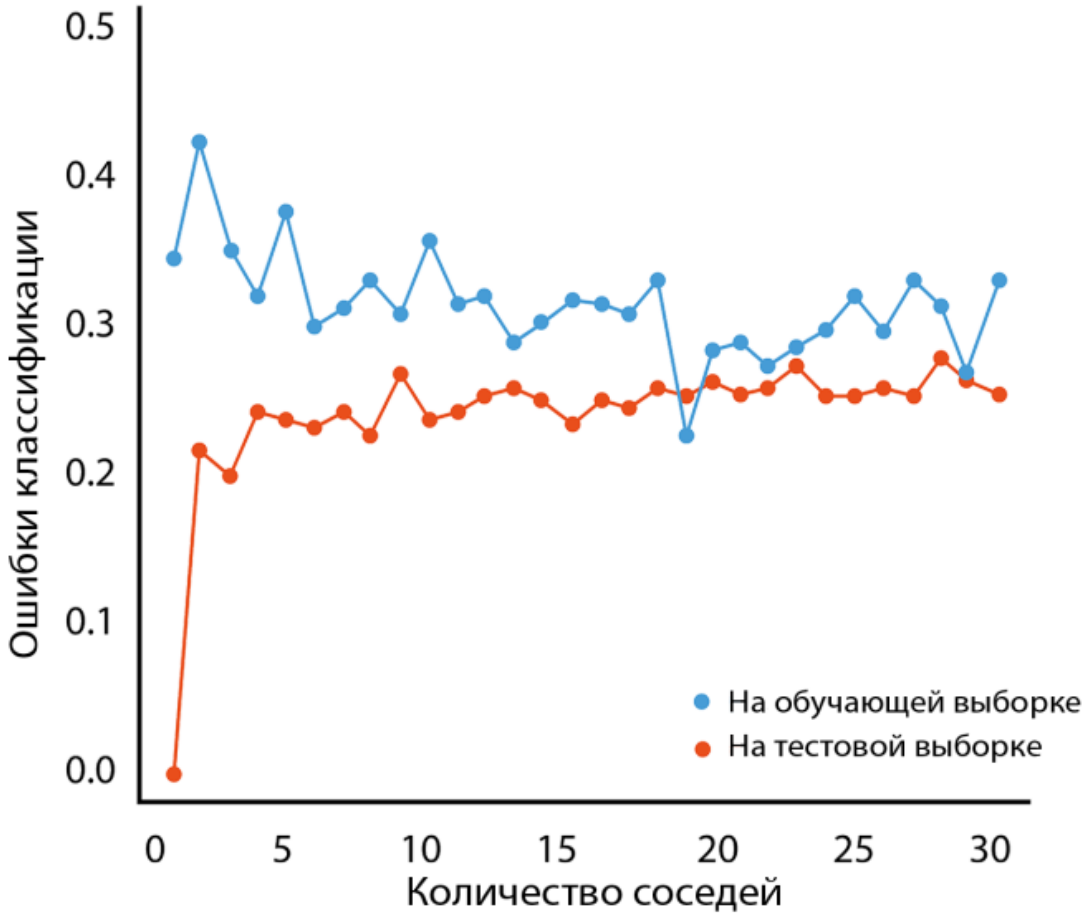
Взвешенный kNN для регрессии



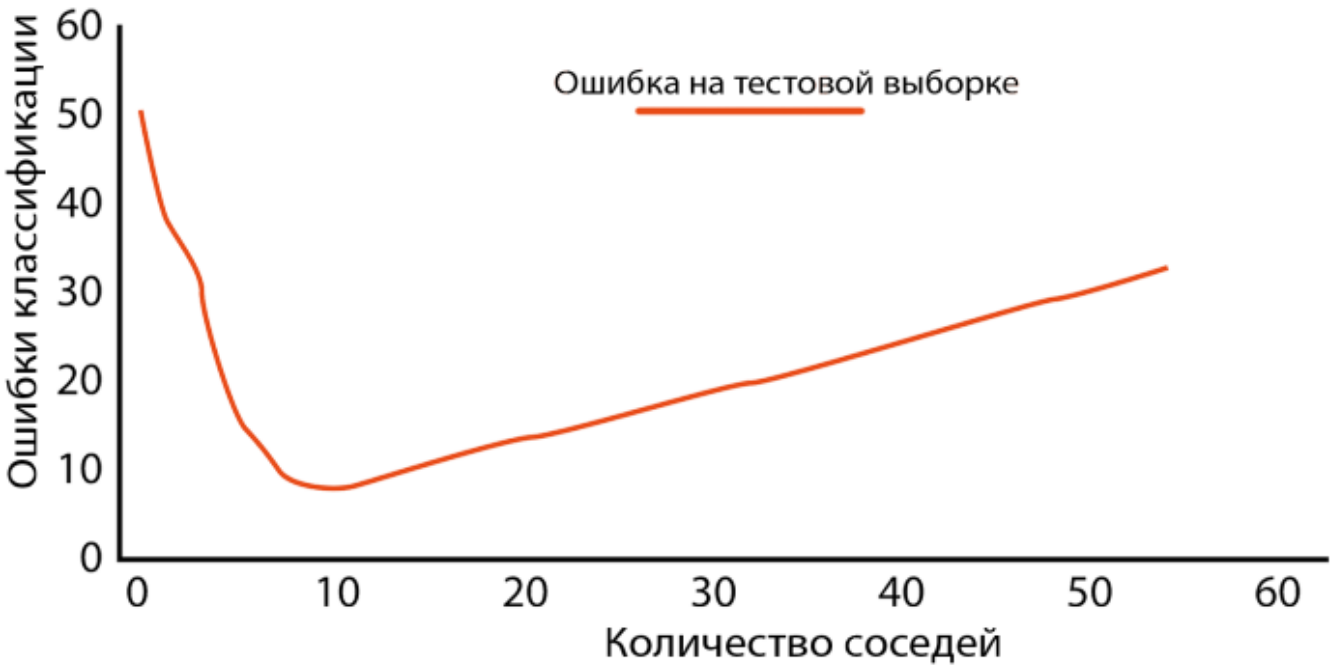
Настройка параметров в kNN

Проверять качество работы алгоритма с выбранными параметрами лучше не на обучающей выборке, а на отложенной. Также можно использовать кросс-валидацию.

Количество соседей



Настройка параметров в kNN



Выбирать следует такое значение k, при котором достигается наилучшая оценка качества работы алгоритма на контроле.

Веса соседей как функция от номера

Простейший вариант — это $w(x) = 1$.
Если выбор $w(x) = 1$ не дает желаемых результатов, можно попробовать определить веса как функцию от номера соседа:
$w(i) = q^i, \quad 0 < q < 1$ $w(i) = \frac{1}{i}, \quad w(i) = \frac{1}{i+a}, \quad w(i) = \frac{1}{(i+a)^b}$
$w(i) = 1 - \frac{i-1}{k}$ (не очень удачный вариант).

Веса объектов как функция от расстояния

$$w(d) = \frac{1}{(d+a)^b}$$

$$w(d) = q^d, \quad 0 < q < 1$$

Метрики в kNN

Метрика является функцией, задающей расстояние в метрическом пространстве, и должна удовлетворять следующим аксиомам:

- $\rho(x, y) \geq 0$, причем $\rho(x, y) = 0 \Longleftrightarrow x = y$.
- $\rho(x, y) = \rho(y, x)$,
- $\rho(x, y) \leq \rho(x, z) + \rho(z, y)$.

Примеры метрик

Можно привести следующие примеры метрик:

- Евклидова метрика:

$$\rho(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

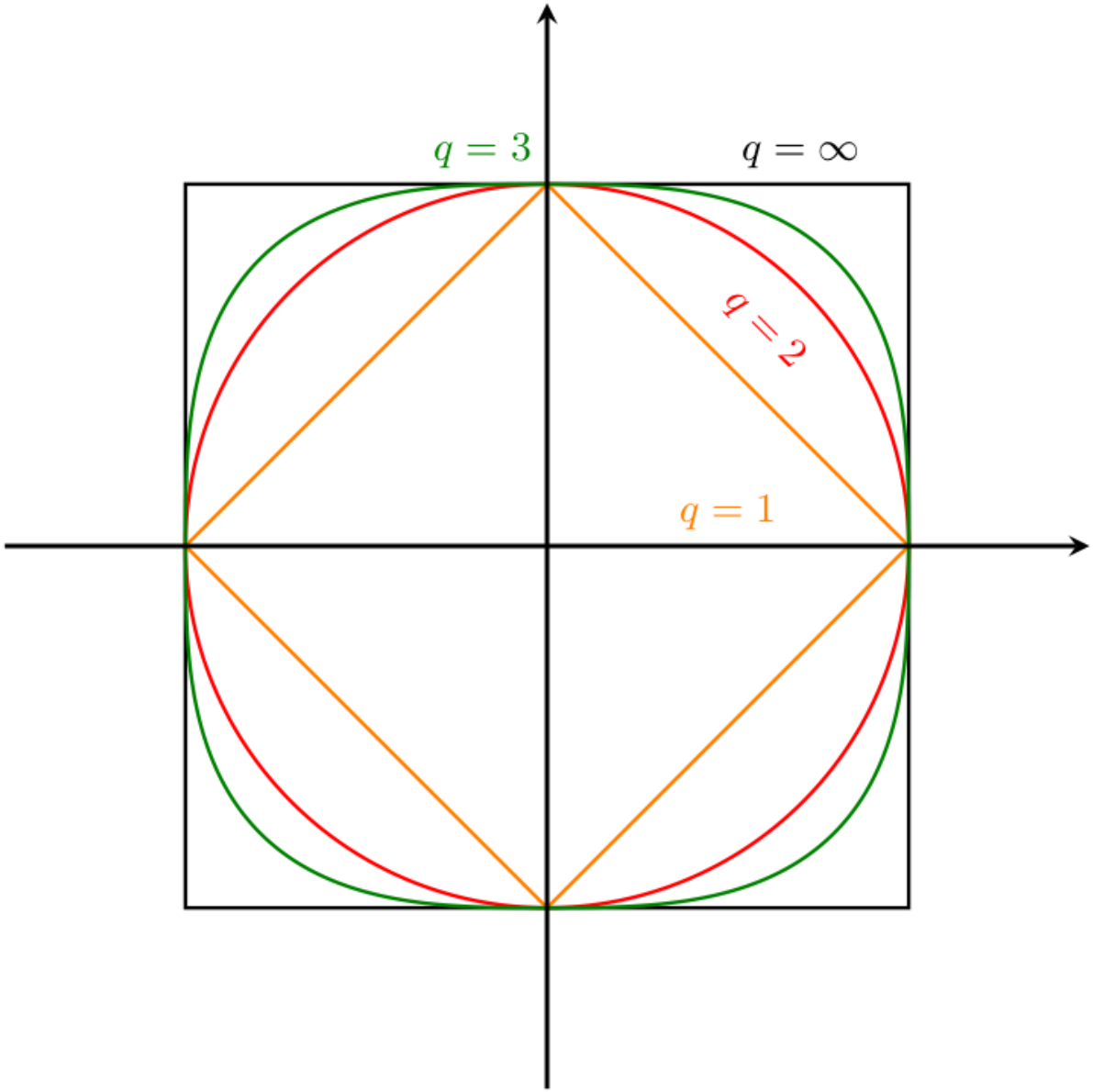
- Манхэттенская метрика:

$$\rho(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

- Метрика Минковского (обобщение Евклидовой и Манхэттенской метрик):

$$\rho(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^q \right)^{\frac{1}{q}}.$$

Примеры метрик



Функции близости

В задачах анализа текста, используется так называемая косинусная мера, которая представляет собой косинус угла между векторами:

$$similarity = \cos\theta = \frac{x \cdot y}{\|x\| \cdot \|y\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Функции близости

- Скалярное произведение:

$$\sum x_i y_i$$

- Коэффициент Дайса:

$$\frac{2\sum x_i y_i}{\sum x_i^2 + \sum y_i^2}$$

- Косинусная мера:

$$\frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$$

- Коэффициент Жаккара:

$$\frac{\sum x_i y_i}{\sum x_i^2 + \sum y_i^2 - \sum x_i y_i}$$

Проклятие размерности

Небольшие отличия в большом числе координат

$$x_1 = (a_1, a_2, ..., a_N),$$
$$x_2 = (a_1 + \varepsilon, a_2 + \varepsilon, ..., a_N + \varepsilon),$$
$$x_3 = (a_1, a_2 + \Delta, ..., a_N).$$

Когда признаков очень много, незначительные различия в каждом признаке могут значить больше, чем одно большое различие в одном

Почти одинаковые расстояния

Когда количество объектов сравнимо с количеством признаков, может возникнуть ситуация, что расстояния между двумя любыми объектами будет почти одинаковым.

Экспоненциальный рост необходимых данных

Пусть X — вектор в признаковом пространстве из N бинарных признаков, например:

$$X = (0, 0, 1, 0, 1, 1, \ldots, 1)$$

Всего в этом пространстве 2^N различных векторов,размер обучающей выборки, необходимый, чтобы покрыть все возможные комбинации эти признаков будет также порядка 2^N.

Экспоненциальный рост необходимых данных

Пусть в N-мерном пространстве дан куб с ребром 1 и меньший куб, длина ребер которого равна $\ell < 1$. Меньший куб вложен в больший таким образом, что они имеют общую вершину и их грани попарно параллельны. Доля объема меньшего куба от объема большего выражается формулой:

$$\frac{v}{V} = \ell^N \to 0, \qquad N \to \infty,$$

Рекомендации фильмов с помощью kNN

User-based подход: среди пользователей ищутся наиболее похожие на того пользователя, для которого делается прогноз.

	Пила	Улица Вязов	Ванильное небо	1 + 1
Маша	5	4	1	2
Юля		5	2	
Вова			3	5
Коля	3		4	5
Петя				4
Ваня		5	3	3

Похожесть пользователей

В качестве меры похожести $w_{i,j}$ двух пользователей можно использовать коэффициент корреляции:

$$w_{i,j} = \frac{\sum_a (r_{i,a} - \bar{r}_i)(r_{j,a} - \bar{r}_j)}{\sqrt{\sum_a (r_{i,a} - \bar{r}_i)^2} \sqrt{\sum_a (r_{j,a} - \bar{r}_j)^2}},$$

$\bar{r}_i = \frac{1}{N_i} \sum_a r_{i,a}$ — средние оценки i-го пользователя, N_i — количество просмотренных им фильмов. Суммирование ведется только по тем фильмам, которые смотрели оба пользователя.

Прогнозирование рейтинга

$$\hat{r}_{i,a} = \bar{r}_i + \frac{\sum_j (r_{j,a} - \bar{r}_j) w_{i,j}}{\sum_j |w_{i,j}|}.$$

$$\hat{r}_{i,a} = \bar{r}_i + \frac{\sum_{j \in kNN(i)} (r_{j,a} - \bar{r}_j) w_{i,j}}{\sum_{j \in kNN(i)} |w_{i,j}|}.$$

Таким образом метод kNN может быть адаптирован к задаче рекомендации.

Метод опорных векторов (SVM)

Это просто линейный классификатор

$$a(x) = \text{sign}(\langle w, x \rangle - w_0),$$

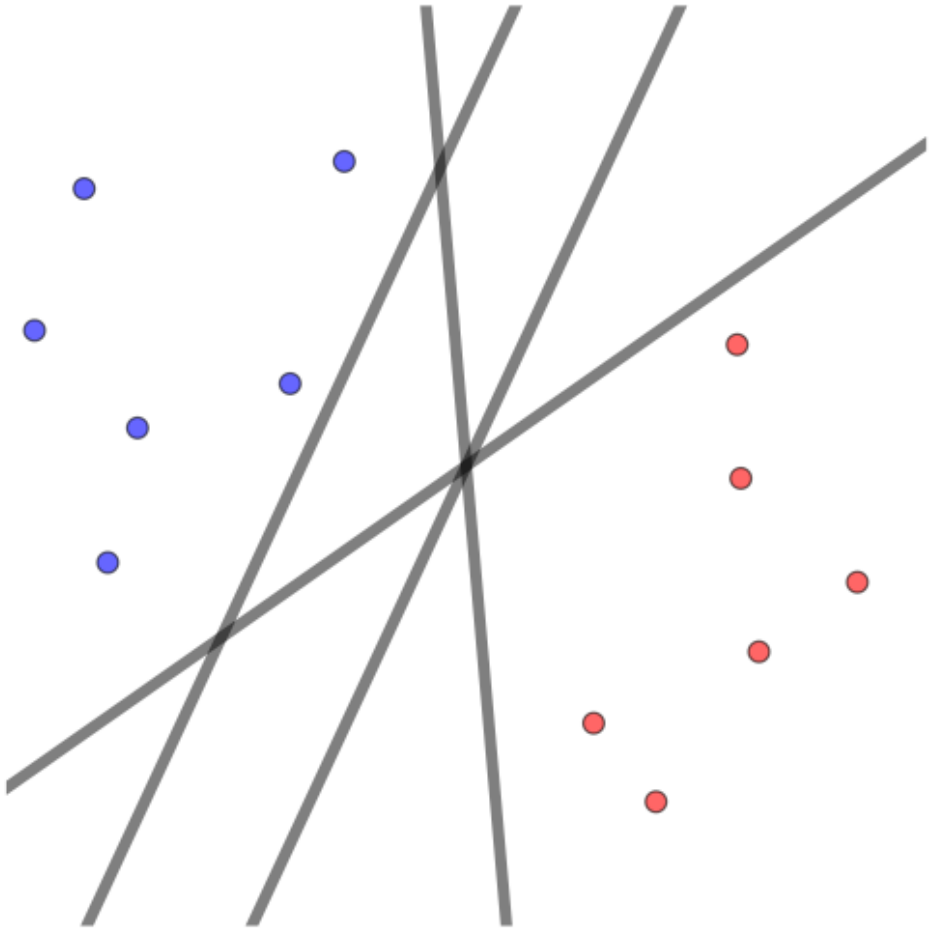
использующий кусочно-линейную функцию потерь

$$L(M_i) = \max\{0, 1 - M_i\} = (1 - M_i)_+$$

и L 2 -регуляризатор:

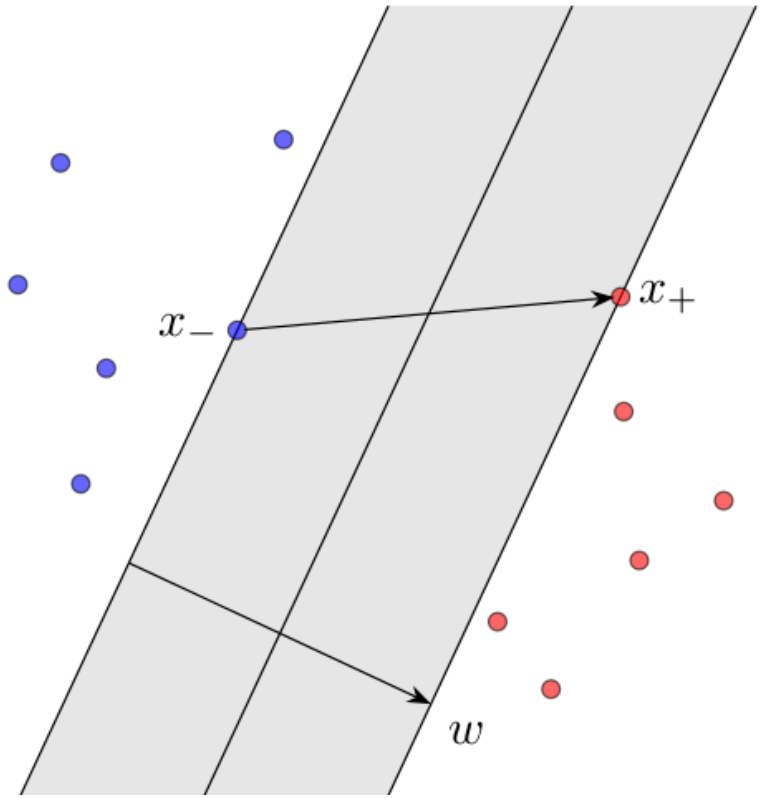
$$\sum_{i=1}^{\ell} \underbrace{L(M_i)}_{\text{Функция потерь}} + \underbrace{\gamma \|w\|^2}_{\text{Квадратичный регуляризатор}} \to \min_w .$$

Разделяющая полоса в случае линейно разделимой выборки



Разделяющая полоса в случае линейно разделимой выборки

$$\langle w, x \rangle - w_0 = 0$$



Разделяющая полоса в случае линейно разделимой выборки

Всегда можно выбрать (отнормировать) w и w 0 таким образом, чтобы уравнения граничных плоскостей имели вид:

$$\langle w, x \rangle - w_0 = \pm 1$$

Это условие нормировки можно также сформулировать следующим образом:

$$\min_{i=1,\ldots,\ell} y_i(\langle w, x \rangle - w_0) = 1.$$

Разделяющая полоса в случае линейно разделимой выборки

На каждой из двух граничных плоскостей будет лежать как минимиум один объект из соответствующего ей класса (иначе расстояние между плоскостями можно увеличить). Пусть x_+ и x_- — два таких вектора, лежащие на построенных плоскостях и принадлежащие соответствующим классам.
Ширины разделяющей полосы

$$\left\langle (x_+ - x_-), \frac{w}{\|w\|} \right\rangle = \frac{2}{\|w\|}$$

Разделяющая полоса в случае линейно разделимой выборки

Задача построения такой разделяющей гиперплоскости, что расстояние между соответствующими ей граничными плоскостями будет максимальным:

$$\begin{cases} \langle w, w \rangle \to \min, \\ y_i(\langle w, x \rangle - w_0) \geq 1, \quad i = 1, ..., \ell. \end{cases}$$