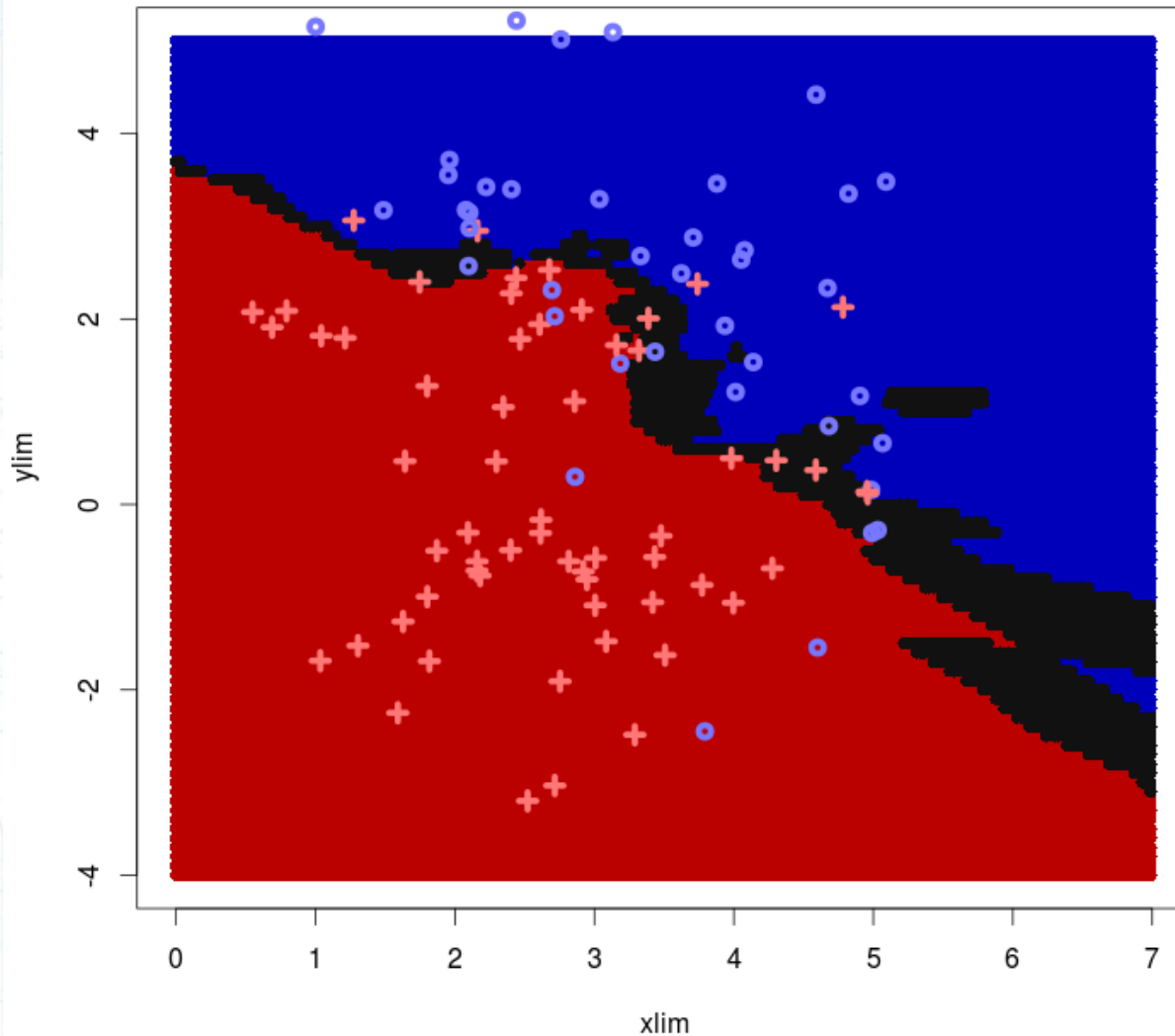


# Машинное обучение

## Лекция 3. Метрические алгоритмы



# Содержание лекции

- Обобщенный алгоритм
- Примеры частных алгоритмов:
  - метод ближайших соседей
  - метод окна Парзена
- Понятие выступа объекта
- Алгоритм отбора эталонов
- Проклятие размерности
- Выбор метрики

# Гипотезы

- Задачи классификации и регрессии:
  - $X$  — объекты,  $Y$  — ответы;
  - $X_\ell = (x_i, y_i)$  — обучающая выборка;
- Гипотеза компактности (для классификации):
  - Близкие объекты, лежат в одном классе.
- Гипотеза непрерывности (для регрессии):
  - Близким объектам соответствуют близкие ответы.
- Формализация понятия «близости»:
  - Задана функция расстояния  $\rho : X \times X \rightarrow [0, \infty)$ .
- Пример. Евклидово расстояние и его обобщение:

$$\rho(x, x_i) = \left( \sum_{j=1}^n |x^j - x_i^j|^2 \right)^{1/2} \quad \rho(x, x_i) = \left( \sum_{j=1}^n w_j |x^j - x_i^j|^p \right)^{1/p}$$



# Обобщенный алгоритм

- Для заданного  $x \in X$  отсортируем объекты  $x_1, \dots, x_\ell$ :

$$\rho(x, x^{(1)}) \leq \rho(x, x^{(2)}) \leq \dots \leq \rho(x, x^{(\ell)}),$$

–  $x^{(i)}$  —  $i$ -тый ближайший сосед объекта  $x$

- Метрический алгоритм классификации:

$$a(x) = \arg \max_{y \in Y} \Gamma_y(x) = \arg \max_{y \in Y} \sum_{\substack{i=1 \\ y_i=y}}^{\ell} w(i, x)$$

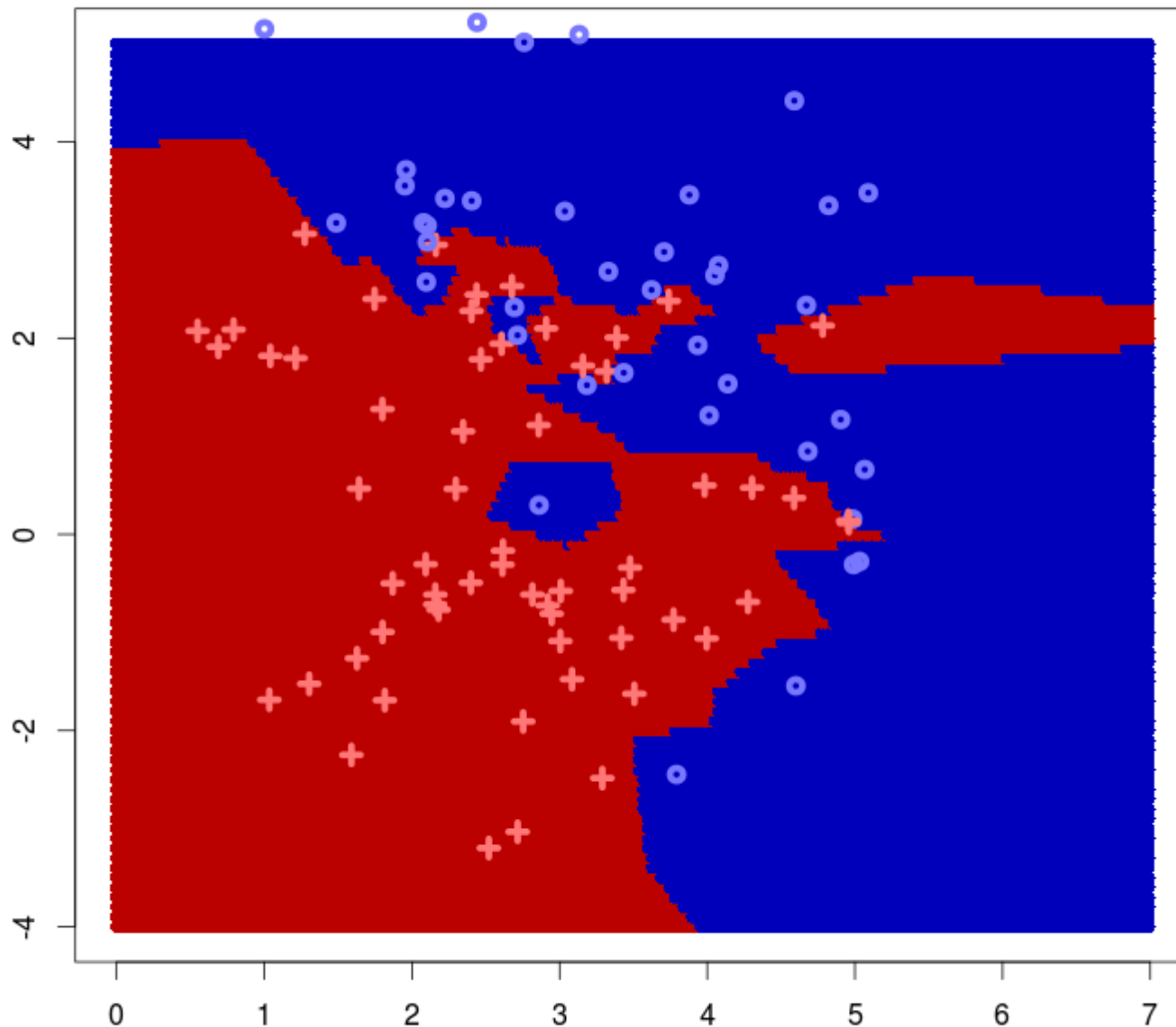
- $w(i, x)$  — вес (степень важности)  $i$ -го соседа объекта  $x$ ,  $\geq 0$ ,  $\searrow$  по  $i$
- $\Gamma_y(x)$  — близость объекта  $x$  к классу  $y$

# Lazy learning

- Это так называемое ленивое обучение, в котором нет этапа тренировки параметров модели. Сразу происходит этап предсказания.
- Подходит для задач, в которых сложно сформулировать набор признаков, но легко сравнивать объекты (пример: сравнительная геномика)
- Недостаток: медленный процесс предсказания

# Метод ближайшего соседа

- $w(i,x) = 1$ , если  $i = 1$
- $w(i,x) = 0$ , в противном случае

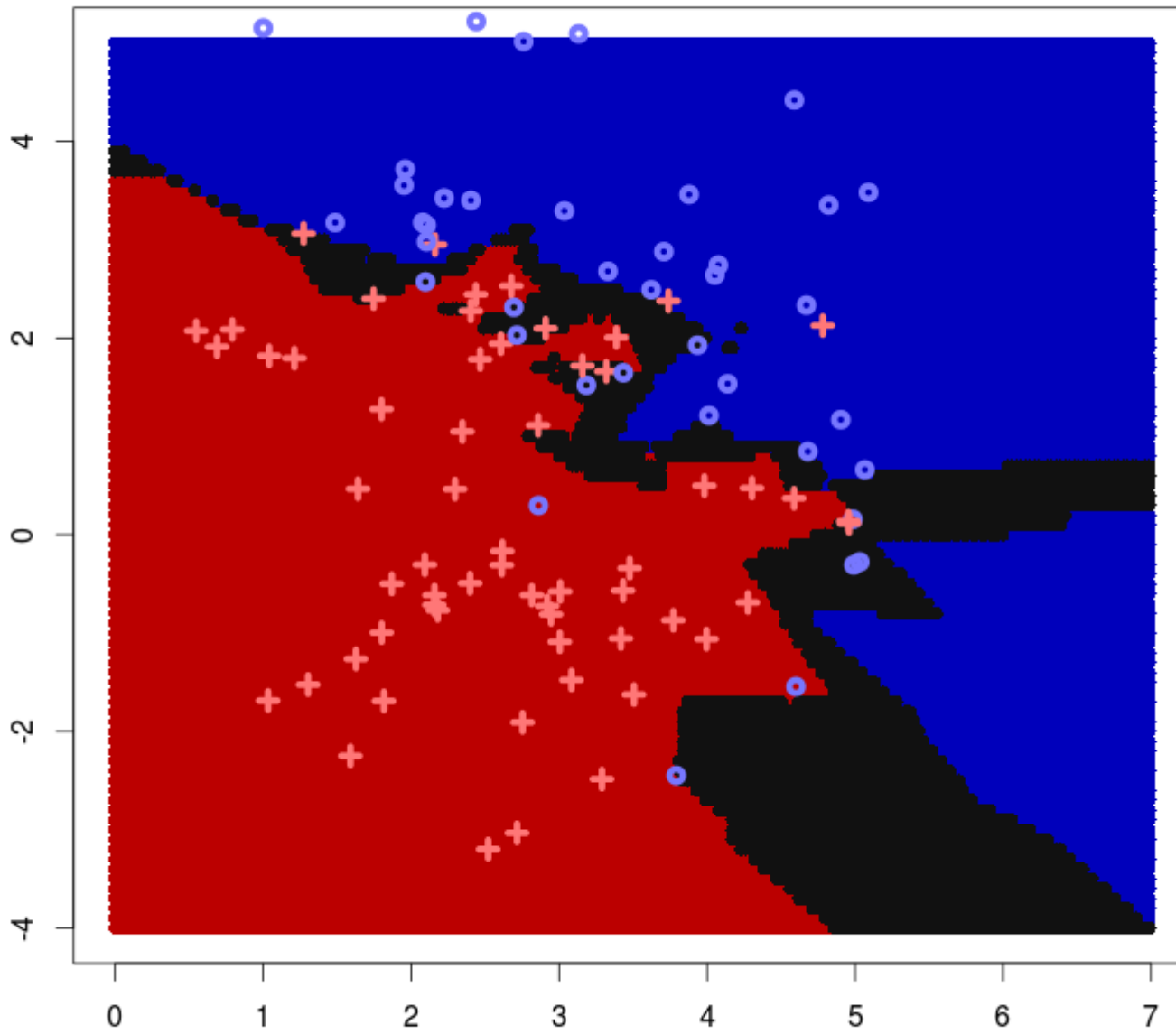




# Метод k ближайших соседей

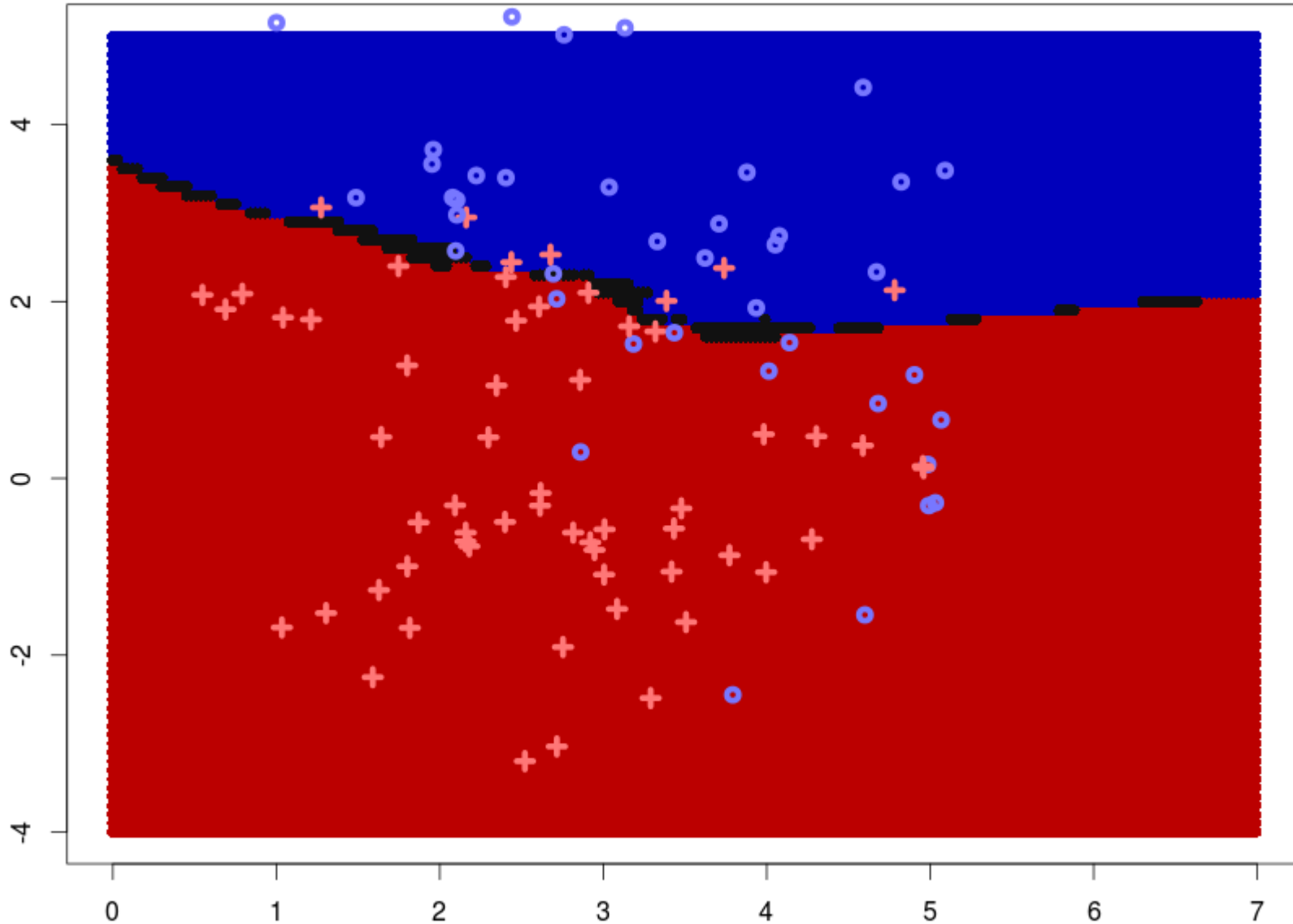
- $w(i,x) = 1$ , если  $i \leq k$
- $w(i,x) = 0$ , в противном случае

$k = 4$



# Метод k ближайших соседей

- $k = 60$





# Оптимизация k

- Функционал скользящего контроля (leave-one-out):
- Для каждого  $x_i$ 
  - сортируем всех его соседей (за исключением его самого) по возрастанию расстояния
  - делаем предсказание для  $x_i$
- Цель: минимизировать число ошибок

$$\text{LOO}(k, X^\ell) = \sum_{i=1}^{\ell} \left[ a(x_i; X^\ell \setminus \{x_i\}, k) \neq y_i \right] \rightarrow \min_k$$

# Метод $k$ взвешенных ближайших соседей

Проблема метода ближайших соседей – близкие и далекие учитываются с одним весом

$$w(i, x) = [i \leq k] w_i,$$

где  $w_i$  — вес, зависящий только от номера соседа;

**Возможные эвристики:**

$w_i = \frac{k+1-i}{k}$  — линейные убывающие веса;

$w_i = q^i$  — экспоненциально убывающие веса,  $0 < q < 1$ ;

# Метод k взвешенных ближайших соседей

Проблема метода ближайших соседей – близкие и далекие учитываются с одним весом

$$w(i, x) = [i \leq k] w_i,$$

где  $w_i$  — вес, зависящий только от номера соседа;

**Возможные эвристики:**

$w_i = \frac{k+1-i}{k}$  — линейные убывающие веса;

$w_i = q^i$  — экспоненциально убывающие веса,  $0 < q < 1$ ;

**Проблема. Две ситуации:**

1)  $\rho(x, x_i) = 1; 1.1; 1.2; 1.3$

2)  $\rho(x, x_i) = 1; 1.1; 50; 51$

приведут к одним и тем же весам



# Метод окна Парзена

$w(i, x) = K\left(\frac{\rho(x, x^{(i)})}{h}\right)$ , где  $h$  — ширина окна,  
 $K(r)$  — ядро, не возрастает и положительно на  $[0, 1]$

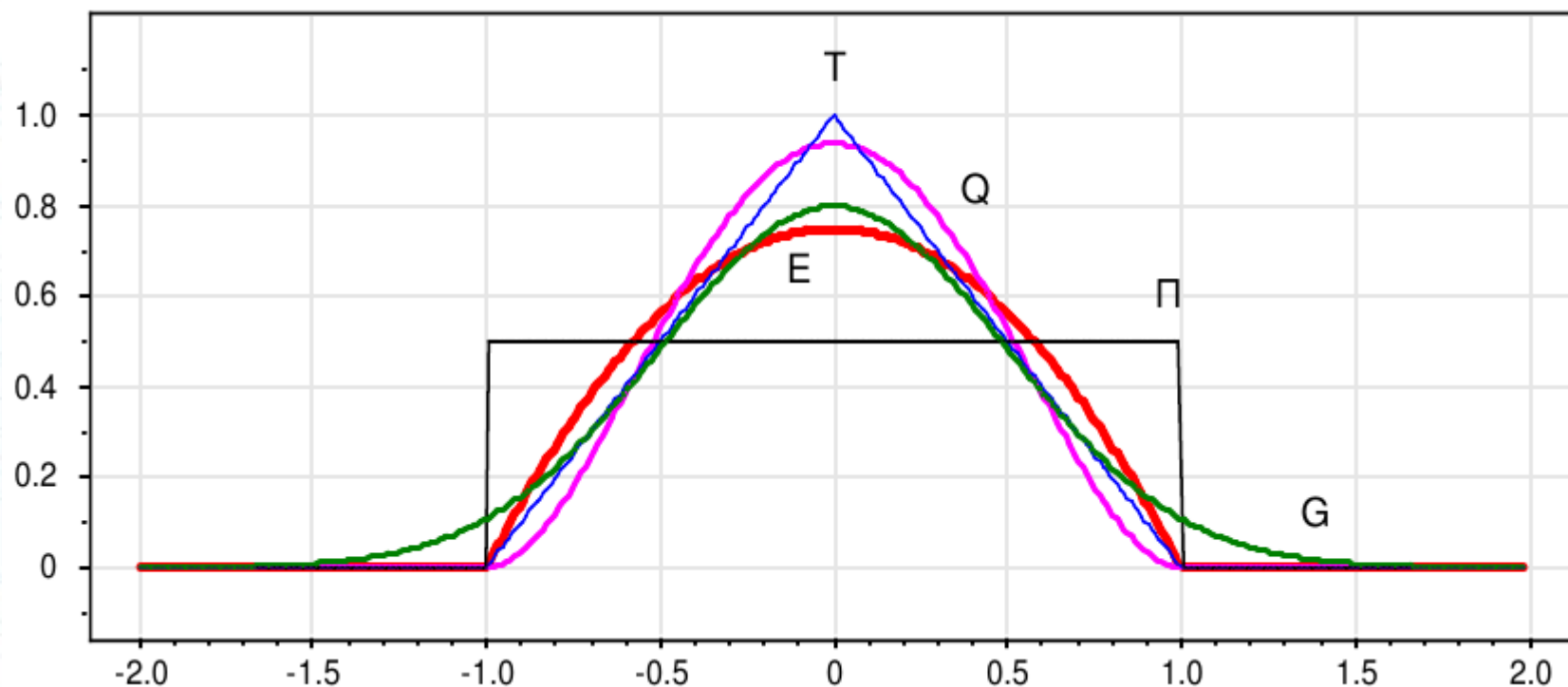
# Метод окна Парзена

$w(i, x) = K\left(\frac{\rho(x, x^{(i)})}{h}\right)$ , где  $h$  — ширина окна,  
 $K(r)$  — ядро, не возрастает и положительно на  $[0, 1]$

При фиксированной ширине окна качество классификатора сильно зависит от плотности точек.

Выход: положить ширину  $h$  равной расстоянию до  $k$ -того соседа

# Часто используемые ядра



$\Pi(r) = [ |r| \leq 1 ]$  — прямоугольное

$T(r) = (1 - |r|) [ |r| \leq 1 ]$  — треугольное

$E(r) = (1 - r^2) [ |r| \leq 1 ]$  — квадратичное (Епанечникова)

$Q(r) = (1 - r^2)^2 [ |r| \leq 1 ]$  — кватрическое

$G(r) = \exp(-2r^2)$  — гауссовское



# Отступ (выступ) объекта

- Пусть классификатор  $a(x)$  работает по правилу:

$$a(x) = \arg \max_{y \in Y} \Gamma_y(x)$$

- Отступом (margin) объекта  $x_i$  обучающей выборки называется величина

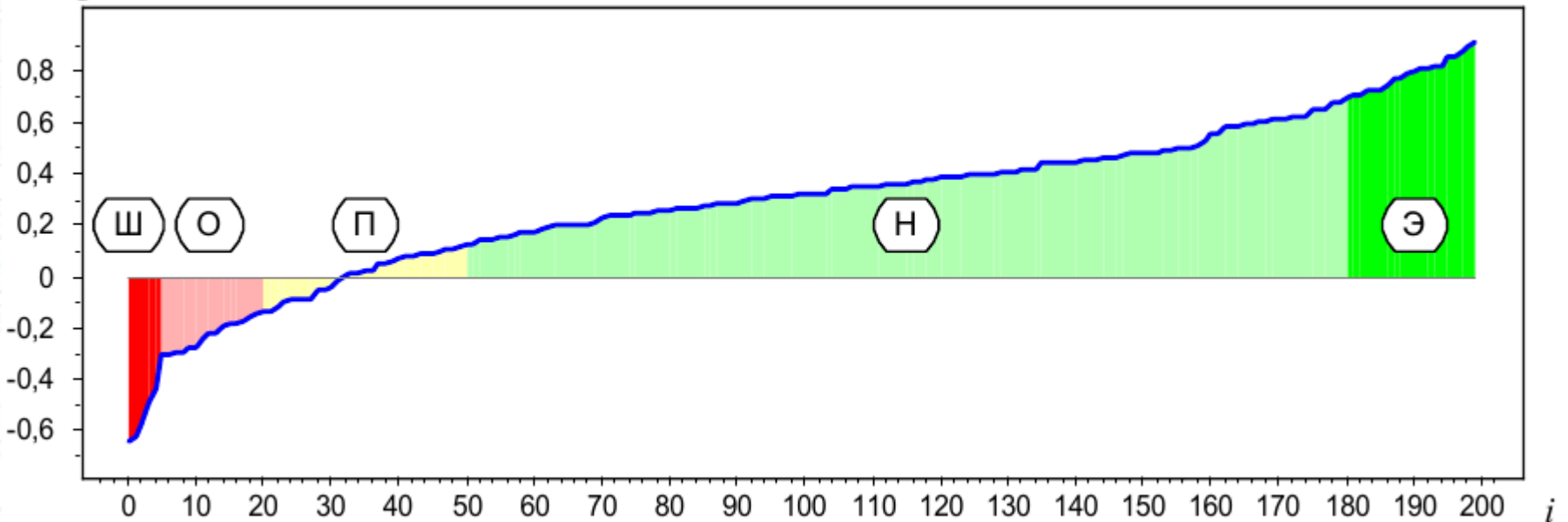
$$M(x_i) = \Gamma_{y_i}(x_i) - \max_{y \in Y \setminus y_i} \Gamma_y(x_i)$$

- Отступ показывает степень типичности объекта: чем больше  $M(x_i)$ , тем «глубже»  $x_i$  в своём классе;  $M(x_i) < 0 \Leftrightarrow a(x_i) \neq y_i$

# Типы объектов в зависимости от выступления

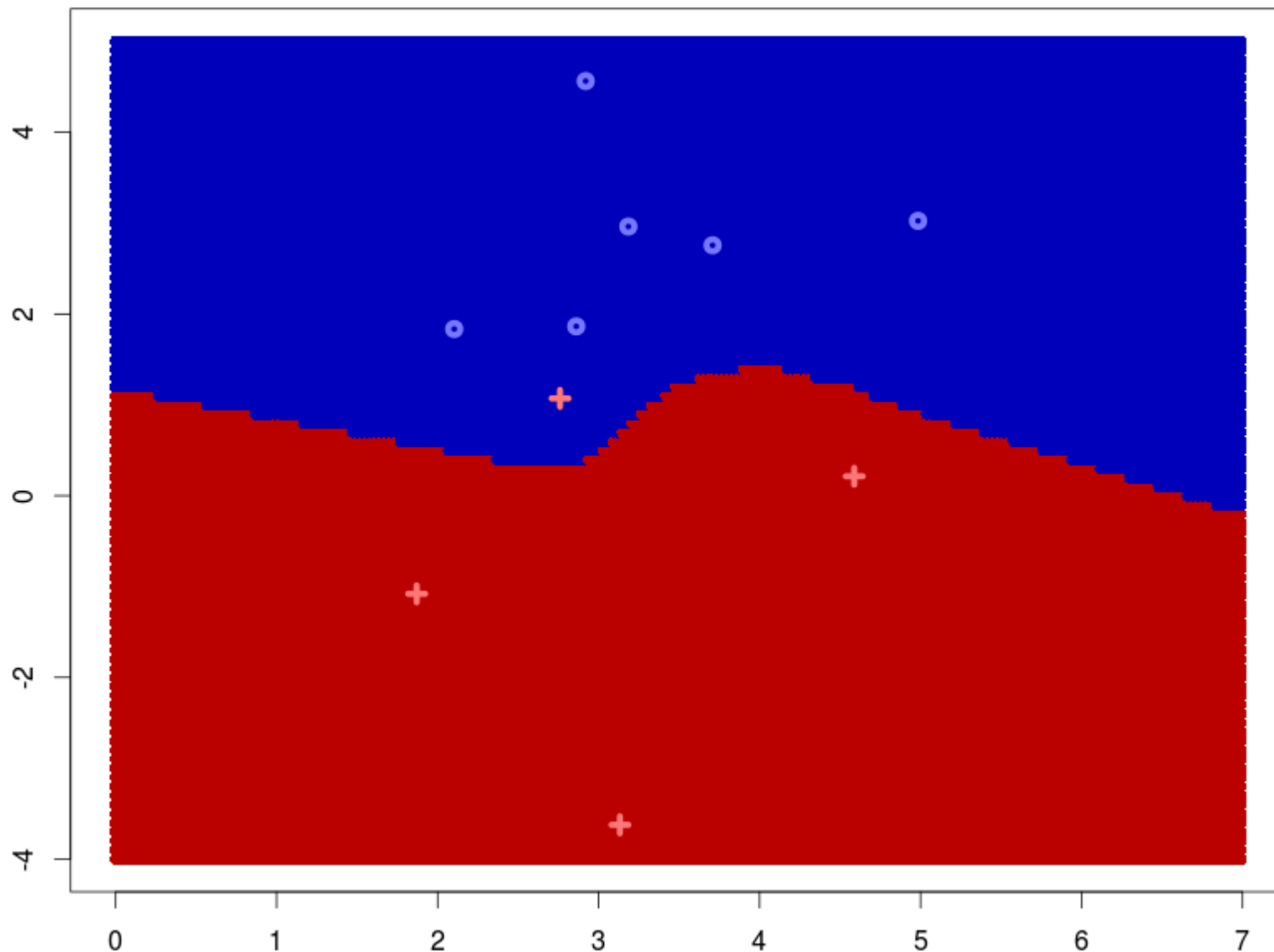
- Э — эталонные (можно оставить только их);
- Н — неинформативные (можно удалить из выборки);
- П — пограничные (их классификация неустойчива);
- О — ошибочные (причина ошибки — плохая модель);
- Ш — шумовые (причина ошибки — плохие данные).

*Margin*



# Отступ (выступ) объекта

Вычислите отступы для всех объектов для метода 3NN





# Отбор эталонов

- Задача: выбрать оптимальное подмножество эталонов  $\Omega$  из обучающей выборки. Классификатор будет иметь вид:

$$a(x, \Omega) = \arg \max_{y \in Y} \sum_{\substack{x^{(i)} \in \Omega \\ y_i = y}} w(i, x)$$

- Алгоритм STOLP. Три основных этапа:
  - исключить выбросы и, возможно, пограничные объекты;
  - найти по одному эталону в каждом классе;
  - добавлять эталоны, пока есть отрицательные отступы;

# Алгоритм STOLP

- Исключаем из  $X_\ell$  выбросы  $x_i : M(x_i) < \delta$
- Инициализируем множество эталонов  $\Omega$ , выбирая по одному элементу из каждого класса с максимальным выступом
- Цикл: пока процент ошибок классификации велик и эталонов  $\Omega$  не слишком много
  - добавляем в  $\Omega$  объект с наименьшим выступом



# Проклятие размерности

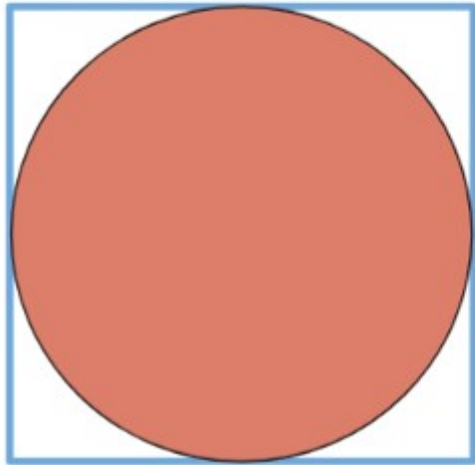
- Проклятие размерности - усреднение значений метрики при большом количестве признаков. Почти до всех ближайших соседей расстояние одинаково
- Почему это происходит:
  - Шар радиуса  $R$  имеет объем  $V(R) \sim R^D$
  - Объем шара радиуса 0.9 в 20-мерном пространстве составляет всего 12% от объема шара радиуса 1.  
Т.е. 88% точек лежит на сфере:  $0.9 < R < 1$

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

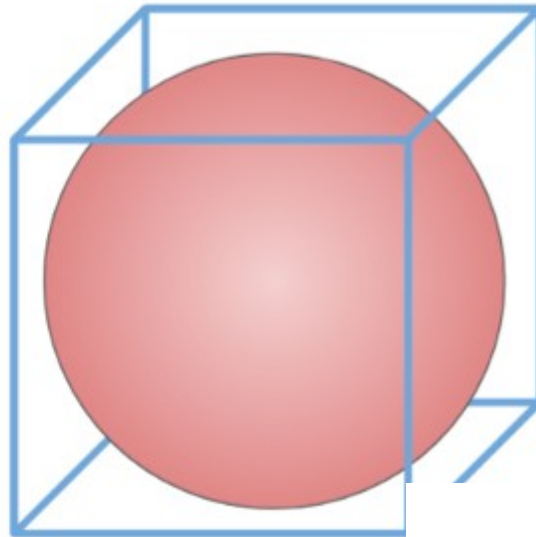


# Проклятие размерности

A

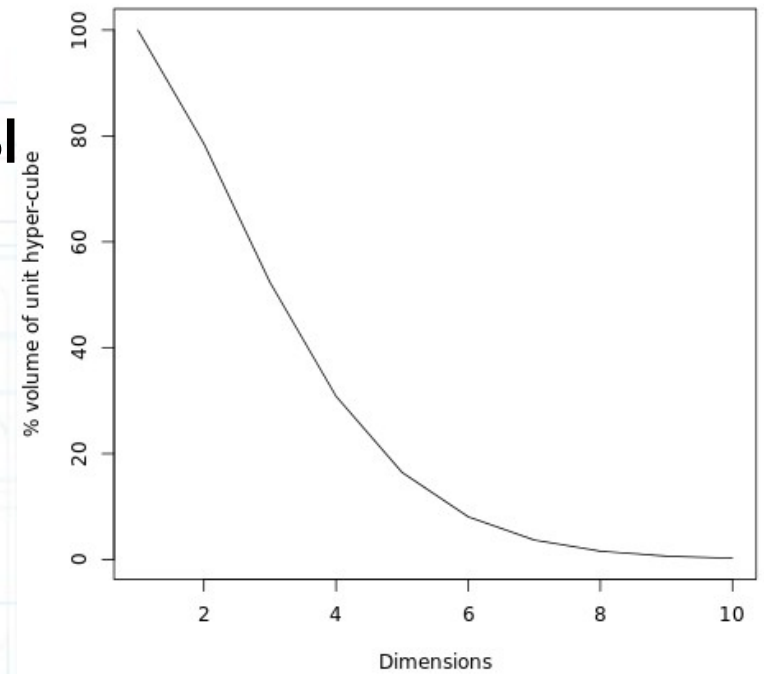


B



Объем вписанной в куб сферы  
в многомерном пространстве  
во много раз меньше объема  
куба!

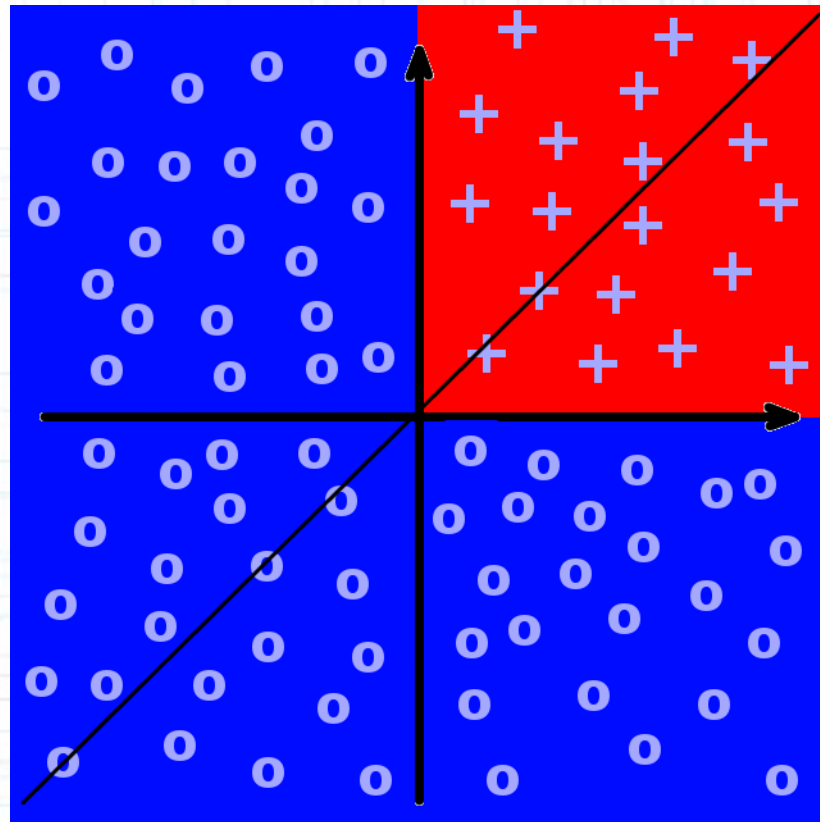
Volume of hyper-sphere



# Проклятие размерности

## Пример

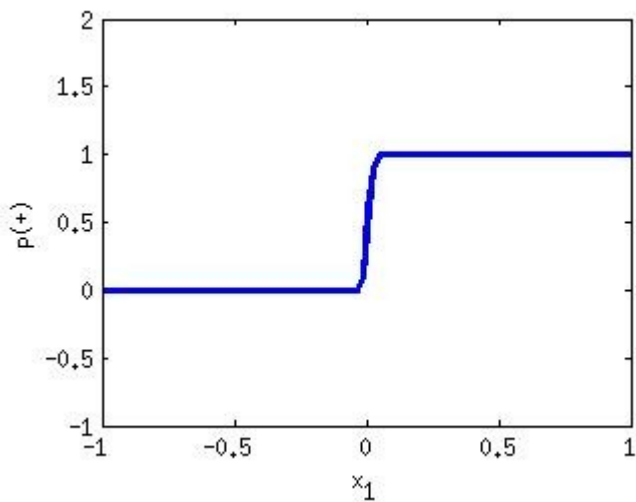
- Пространство признаков:  $\mathbb{R}^n$ .
- Класс +: область  $x_{1,2} > 0$  (остальные координаты произвольны)
- $X_\ell$  - равномерно распределена



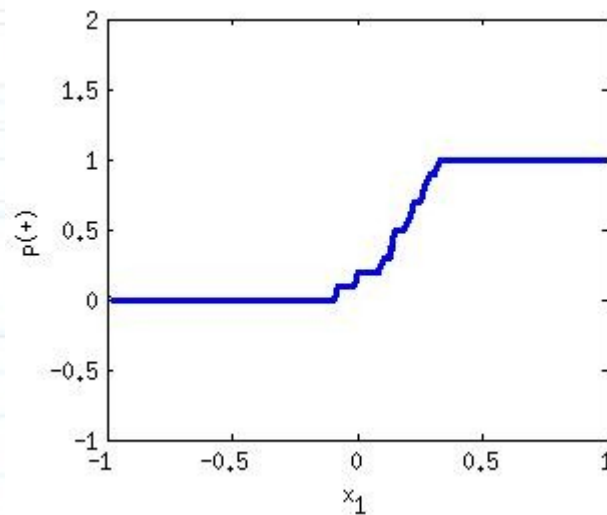
# Проклятие размерности

## Пример

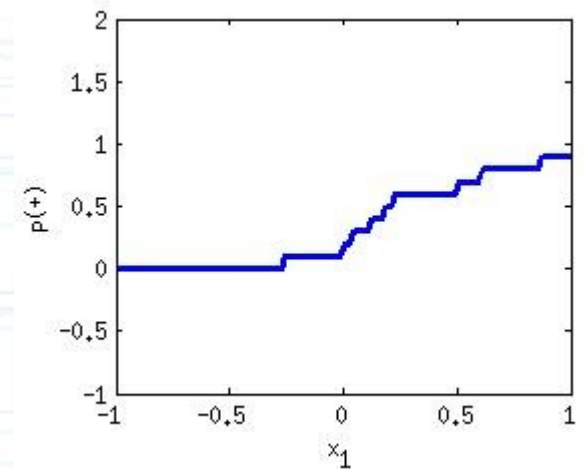
- Метод 10 ближайших соседей.  $\ell = 10000$
- Относительная частота класса “+” на прямой:  $x_1 = x_2$ ,  $x_3 = 0$ ,  $x_4 = 0, \dots$



$n=2$



$n=5$



$n=20$

Вывод: для больших размерностей метрические алгоритмы сглаживают границы областей классов



# Выбор метрики

Взвешенная метрика Минковского:

$$\rho(x, x_i) = \left( \sum_{j=1}^n w_j |f_j(x) - f_j(x_i)|^p \right)^{\frac{1}{p}},$$

где  $w_j$  — неотрицательные веса признаков,  $p > 0$ .

В частности, если  $w_j \equiv 1$  и  $p = 2$ , то имеем евклидову метрику.

**Роль весов  $w_j$ :**

- 1) нормировка признаков;
- 2) степень важности признаков;
- 3) отбор признаков (какие  $w_j = 0$ ?);

# Жадное добавление признаков

1. А вдруг одного признака уже достаточно?

Расстояние по  $j$ -му признаку:  $\rho_j(x, x_i) = |x^j - x_i^j|$ .

Выберем наилучшее расстояние:  $\text{LOO}(j) \rightarrow \min$ .

2. Добавим к расстоянию  $\rho$  ещё один признак  $j$ :

$$\rho^p(x, x_i) := \rho^p(x, x_i) + w_j \rho_j^p(x, x_i), \quad w_j \geq 0.$$

Найдём признак  $j$  и вес  $w_j$ , при которых  $\text{LOO}(j, w_j) \rightarrow \min$  (два вложенных цикла перебора).

3. Можно корректировать вес признака  $k$ , уже вошедшего в  $\rho$ :

$$\rho^p(x, x_i) := \rho^p(x, x_i) + w'_k \rho_k^p(x, x_i), \quad w'_k \geq -w_k.$$

4. Будем добавлять признаки, пока LOO уменьшается.

# Пример необычной метрики

Задача поиска подходящих по цвету вещей

Сравнить!

54281 14605

Загрузить

Загрузить



# Пример необычной метрики 2

Задача сравнения генных последовательностей

