

# Машинное обучение

## Лекция 2. Метрические методы классификации

Катя Тузова

## Разбор ДЗ. Общий алгоритм.

```
standard_deviation = sys.maxsize
for i in range(len(learnX)):
    polynom = OLS.polyfit(learnX, learnY)
    dev = calc_dev(polynom, learnX, learnY)
    if dev < standard_deviation:
        standard_deviation = sd
    else:
        break
```

# Разбор ДЗ. Вопросы.

- Максимальная степень полинома?
- Как использовать test.txt?

# Разбор ДЗ. Число обусловленности.

$$\mu(A) = \|A^{-1}\| \|A\|$$

Число обусловленности матрицы показывает насколько матрица близка к вырожденной (для квадратных матриц).

$$Ax = b \quad \det(A) = 0$$

Если матрица  $A$  вырожденная, то для некоторых  $b$  решение  $x$  не существует, а для других  $b$  оно будет неединственным.

Если  $A$  почти вырожденная, то малые изменения в  $A$  и  $b$  вызывают очень большие изменения в  $x$ .

# Cross-fold validation

Разобьем исходное множество  $X$  на два  $L$  и  $T$  случайным образом.

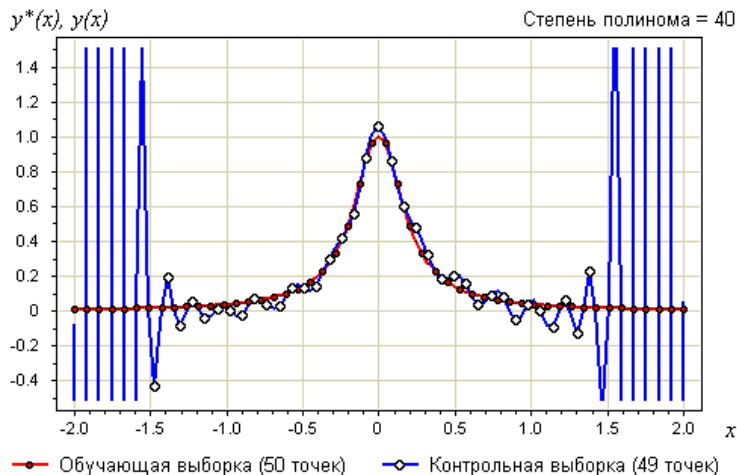
Будем обучаться на  $L$  а проверять результат обучения на  $T$ .

- + Простой и надежный
- + Позволяет оценить распределение на множестве решений
- Последовательные эксперименты зависимы
- Используем мало данных для обучения
- Непонятно как подбирать соотношения  $\frac{|L|}{|T|}$

# $k$ -fold validation

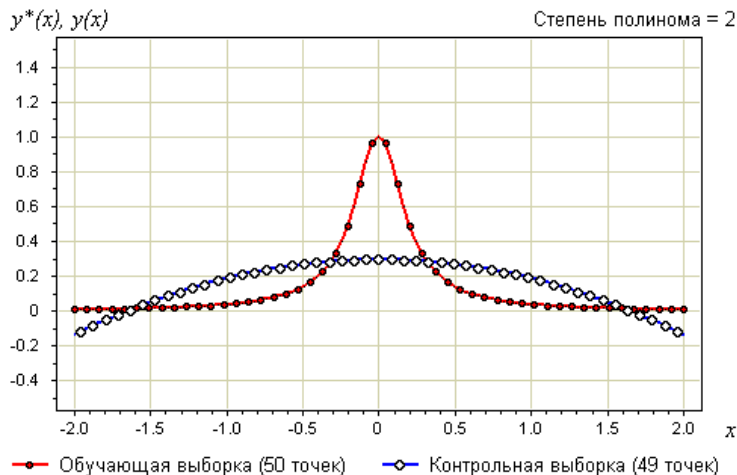
$X$  разбивается на  $k$  частей. Затем на  $k - 1$  частях данных производится обучение модели, а оставшаяся часть данных используется для тестирования.

# Переобучение на полиномах



картинка с [machinelearning.ru](http://machinelearning.ru)

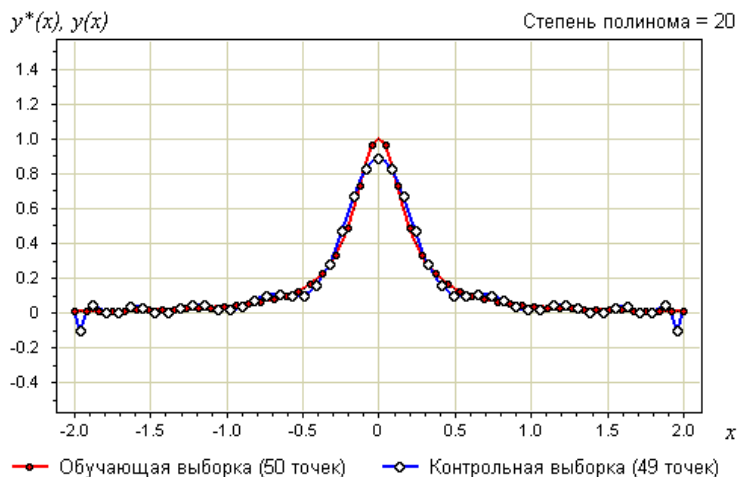
# Недообучение на полиномах



картинка с [machinelearning.ru](http://machinelearning.ru)

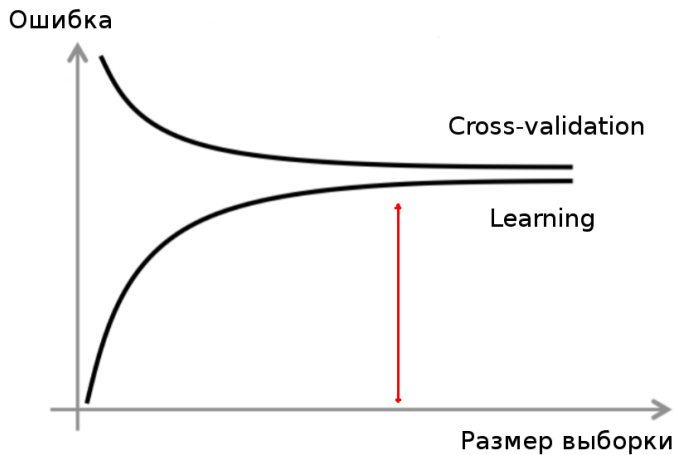


# Fit на полиномах

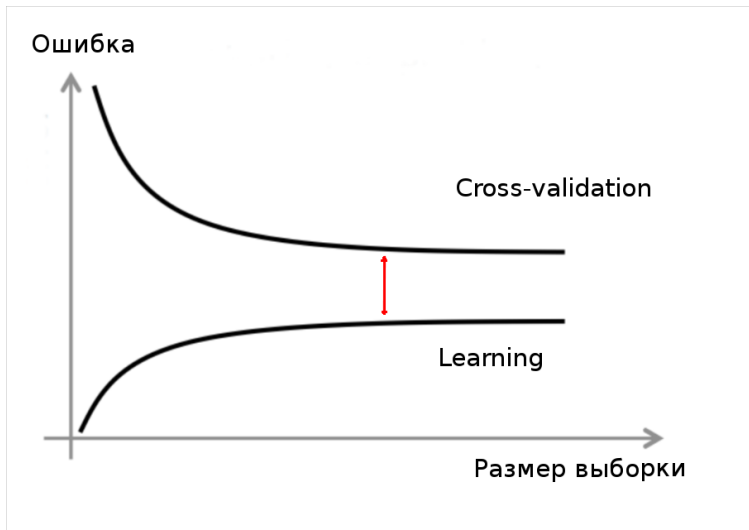


картинка с [machinelearning.ru](http://machinelearning.ru)

# Как понять какая ситуация. Underfitting



# Как понять какая ситуация. Overfitting



# Что в какой ситуации делать

## Переобучение:

- Увеличение числа объектов для обучения
- Введение штрафа для определенных степеней полинома
- Уменьшение количества параметров

## Недообучение:

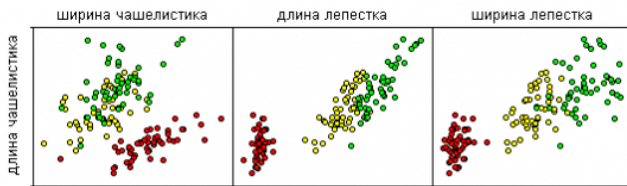
- Добавление степеней полиному
- Увеличение количества параметров

# Пример. Ирисы Фишера

Признаки:

- длина/ширина чашелистика
- длина/ширина лепестка

Задача – разделить на 3 класса



# Гипотеза компактности

**Задача классификации:**

$X$  - объекты,  $Y$  - ответы (классы)

$$X^l = (x_i, y_i)_{i=1}^l$$

**Гипотеза компактности:**

Схожие объекты, как правило, лежат в одном классе.

**"Схожесть":**

Функция расстояния =  $\rho : X \times X \rightarrow [0, \infty)$

# Примеры функции расстояния

Евклидово расстояние:

$$X \in \mathbb{R}^n$$

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

Признаковые описания объектов:

$$u = \{u^1, u^2, \dots, u^n\}$$

$$x_i = \{x_i^1, x_i^2, \dots, x_i^n\}$$

# Обобщенный метрический классификатор

$u \in X$  - произвольный объект, который собираемся классифицировать.

Отсортируем объекты  $x_1, x_2, \dots, x_l$  относительно  $u$ :

$$\rho(u, x_u^1) \leq \rho(u, x_u^2) \leq \dots \leq \rho(u, x_u^l)$$

$x_u^i$  -  $i$ -й сосед объекта  $u$

$y_u^i$  - ответ на  $i$ -м соседе объекта  $u$



# Метрический алгоритм классификации

$$a(u, X^l) = \arg \max_{y \in Y} \underbrace{\sum_{i=1}^l [y_u^i = y] w(i, u)}_{\Gamma_y(u)}$$

$w(i, u)$  - вес  $i$ -го соседа  $u$ , неотрицателен

$\Gamma_y(u)$  - оценка близости объекта  $u$  к классу  $y$

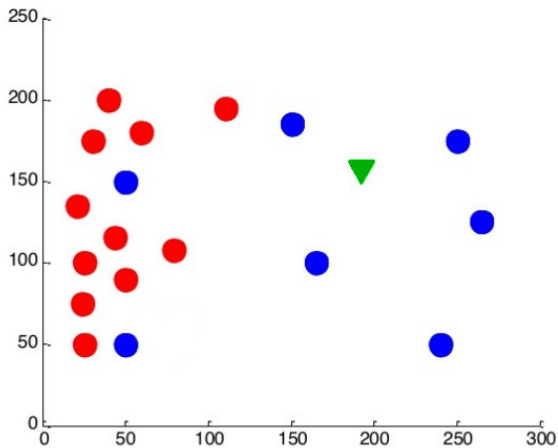
# Метод ближайшего соседа

Объект относится к тому классу, к которому относится ближайший в выборке.

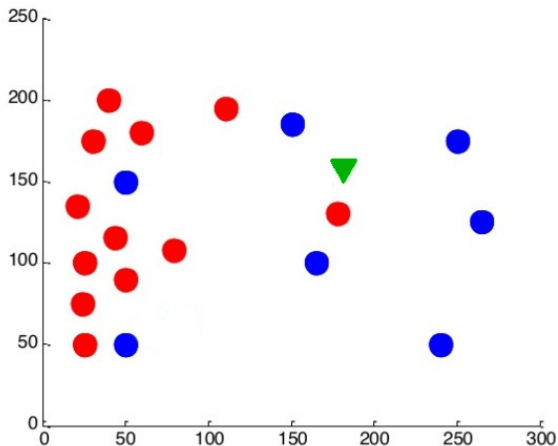
$$w(i, u) = [i = 1]$$

- + Простота
- + Интерпретируемость решения
- Неустойчивость к шуму
- Отсутствие настраиваемых параметров
- Низкое качество классификации
- Надо хранить всю выборку целиком

# Пример.



# Пример неустойчивости к шуму



# Метод k ближайших соседей

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

- + Менее чувствителен к шуму
- + Появляется настраиваемый параметр  $k$
- Неоднозначность классификации при  $\Gamma_y(u) = \Gamma_s(u), y \neq s$

# Как выбрать $k$

Функционал скользящего контроля (leave-one-out):

$$LOO(k, X^l) = \sum_{i=1}^l [a(x_i; X^l \setminus \{x_i\}, k) \neq y] \rightarrow \min_k$$

# Вопрос

Правда ли надо выбрасывать один объект?

# Метод $k$ взвешенных соседей

$w(i, u) = [i \leq k] * w_i$ , где  $w_i$  это 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

Проблема:

как более обоснованно задать веса?



# Ядерная оценка плотности

## Метод окна Парзена

$$w_i = K\left(\frac{\rho(u, x_u^i)}{h}\right)$$

$K(r)$  – ядро, невозрастающее, положительное на  $[0, 1]$

Фиксированной ширины:

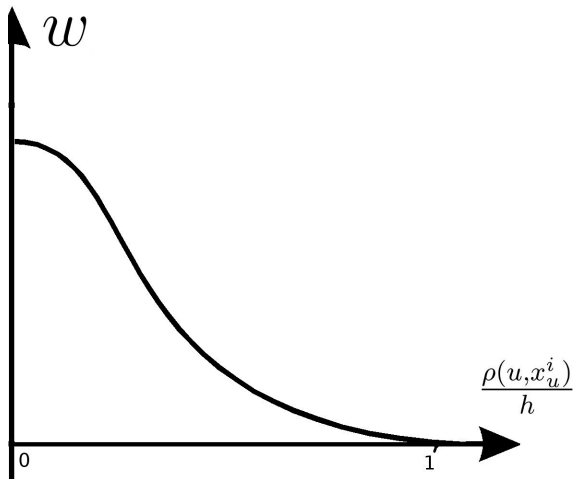
$$a(u, X^l, h, K) = \arg \max_{y \in Y} \sum_{i=1}^l [y_u^i = y] K\left(\frac{\rho(u, x_u^i)}{h}\right)$$

$h$  – ширина окна

Переменной ширины:

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

# Метод окна Парзена



# Выбор метрики

Какие метрики вам известны?  
Как выбрать подходящую?

# Выбор метрики. MMC

Максимизировать сумму расстояний между объектами разных классов при этом сохраняя сумму расстояний между объектами одного класса небольшой.

$$\max \sum_{x_i, x_j \in D} \rho(x_i, x_j)$$

$$\sum_{x_i, x_j \in S} \rho^2(x_i, x_j) \leq 1$$

# Проклятие размерности

Если используемая метрика  $\rho(u, x_u^i)$  основана на суммировании различий по всем признакам, а число признаков очень велико, то все точки выборки могут оказаться практически одинаково далеки друг от друга.

Пример:

Набор признаков объекта генерируется подбрасыванием честной монетки  $n$  раз. Соответственно каждый объект описывается вектором  $[0, 1]^n$ . При таких условиях все объекты будут равноудалены.

# Предобработка данных

Что делать если разные шкалы признаков?

# Предобработка данных

Все признаки должны быть представлены "в одном масштабе".

В противном случае признак с наибольшими числовыми значениями будет доминировать в метрике

# Жадное добавление признаков

- $\rho_j(u, x_i) = |u^j - x_i^j|$  – расстояние по j-му признаку  
 $LOO(j) \rightarrow \min$
- Добавляем признак и строим  $\rho'$   
 $\rho'(u, x_i) = \rho(u, x_i) + w_j \rho_j(u, x_i)$   
 $LOO(j, w_j) \rightarrow \min$
- Замена признака:  
 $\rho'(u, x_i) = \rho(u, x_i) - w_k \rho_k(u, x_i) + w_j \rho_j(u, x_i)$
- Добавляем признаки, пока LOO не увеличивается



# Сверхбольшие выборки

- Проблема хранения
- Проблема быстрого поиска ближайших соседей

# Отступ

Отступ показывает степень "типичности объекта".

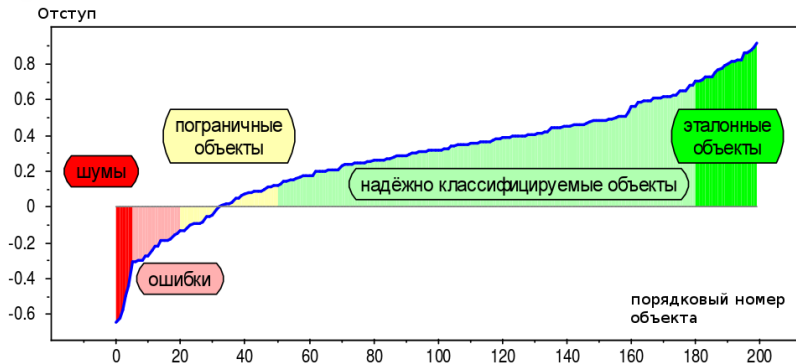
Отступом объекта  $x_i \in X^l$  относительно классификатора  $a$  называется величина

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

# Типы объектов

- Эталонные
- Неинформативные
- Пограничные
- Ошибочные
- Шумовые

# Типы объектов



картинка с [machinelearning.ru](http://machinelearning.ru)

# Отбор эталонных объектов

Задача:

выбрать оптимальное подмножество эталонов  $\Omega \subseteq X^l$

Классификатор будет иметь вид:

$$a(u, \Omega) = \arg \max_{y \in Y} \sum_{x_i \in \Omega} [y_u^i = y] w(i, u)$$

# Алгоритм STOLP

- Исключить выбросы и пограничные объекты
- Найти по одному эталону в каждом классе
- Добавлять эталоны, пока есть отрицательные отступы

# Алгоритм STOLP

- + Сокращается число хранимых объектов
- + Сокращается время классификации
- + Объекты разделяются по величине отступа
  
- Выбор параметра для определения выбросов
- Не высокая эффективность

# Как быстро искать ближайших соседей

- граф ближайших соседей
- k-d дерево
- хеширование (LSH)



<http://www.machinelearning.ru>

Пособие

# Вопросы по курсовым проектам

# На следующей лекции

- Кластеризация. K-means.
- Цели кластеризации.
- Типы кластерных структур.
- Функционал качества кластеризации
- K-средних
- Иерархическая кластеризация.