

## **Лекция 1. Введение в машинное обучение**

## Как строится курс

- Курс ознакомительный
- Ориентирован на практику

План:

- Введение. Обзор основных алгоритмов
- Задачи обработки изображений
- Задачи обработки текстов
- Практики по каждой лекции - важнейшая часть курса

## Как строится курс

Лекции будут читать:

- Владимир Борисов
- Александр Кузнецов
- Илья Лось
- Борис Филиппов

В лекциях активно использовались материалы:

- [Специализации яндекса по машинному обучению](https://www.coursera.org/learn/supervised-learning/)  
(<https://www.coursera.org/learn/supervised-learning/>)
- [Курса К. Воронцова по машинному обучению](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%28%D0%BA%D1%83%D1%80%D1%81_%D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D0%B9%2C_%D0%9A.%D0%92.%D0%92%D0%BE%D1%80%D0%BE%D0%BD%D1%86%D0%BE%D0%B2%29)  
([http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5\\_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5\\_%28%D0%BA%D1%83%D1%80%D1%81\\_%D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D0%B9%2C\\_%D0%9A.%D0%92.%D0%92%D0%BE%D1%80%D0%BE%D0%BD%D1%86%D0%BE%D0%B2%29](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%28%D0%BA%D1%83%D1%80%D1%81_%D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D0%B9%2C_%D0%9A.%D0%92.%D0%92%D0%BE%D1%80%D0%BE%D0%BD%D1%86%D0%BE%D0%B2%29))

## **О чём эта лекция**

- Что такое машинное обучение
- Какие бывают виды задач
- Что такое переобучение
- Разбор реальной задачи

## Что такое машинное обучение

- У термина есть много определений
- Как правило, под ним понимают извлечение закономерностей из примеров
- Применяют в задачах где трудно запрограммировать явные правила:
  - Распознать человека на фотографии
  - Рекомендация фильма на сайте

## Что такое машинное обучение

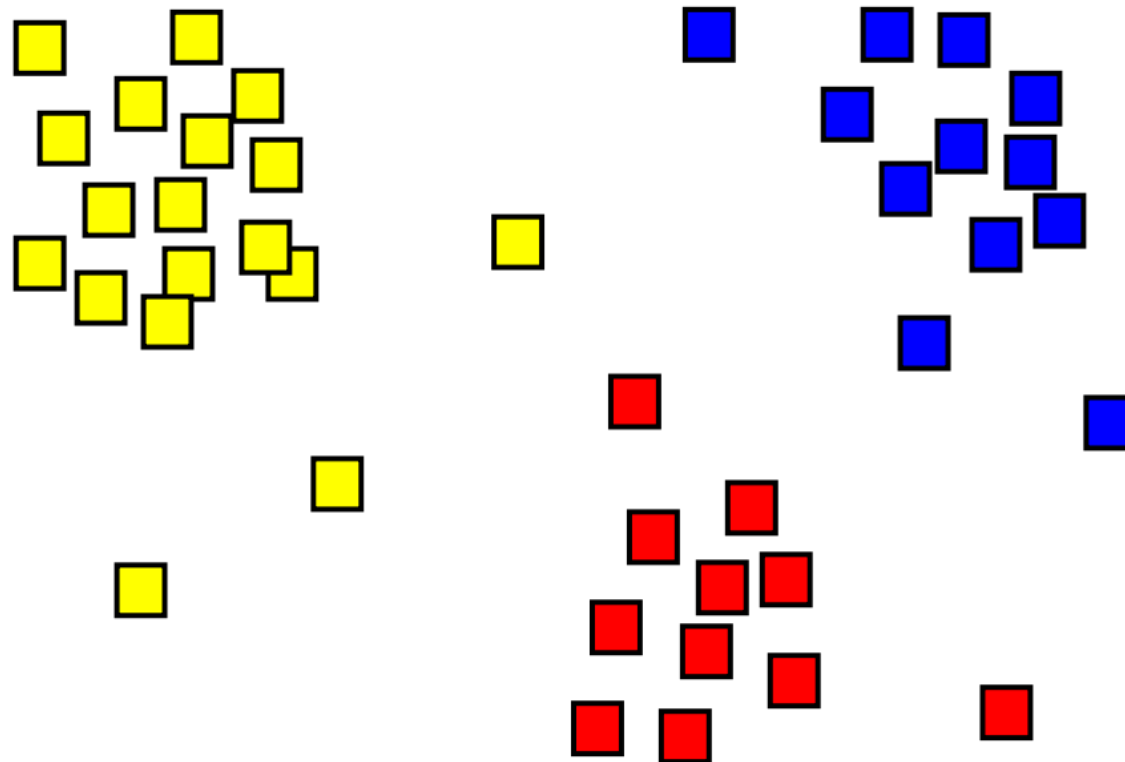
Существует много разных классов задач машинного обучения, часто упоминаемые:

- Обучение с учителем - по известным примерам и ответам (восстановление зависимости)
- Обучение без учителя - только по примерам (выделение закономерностей\структуры в данных)
- Обучение с подкреплением - когда алгоритм-агент обучается взаимодействуя с средой

Далее перечислим задачи обучения без учителя.

## Кластеризация

- Задача: найти группы похожих объектов.
- ... не зная правильных групп (ответов)







## **Кластеризация**

Примеры:

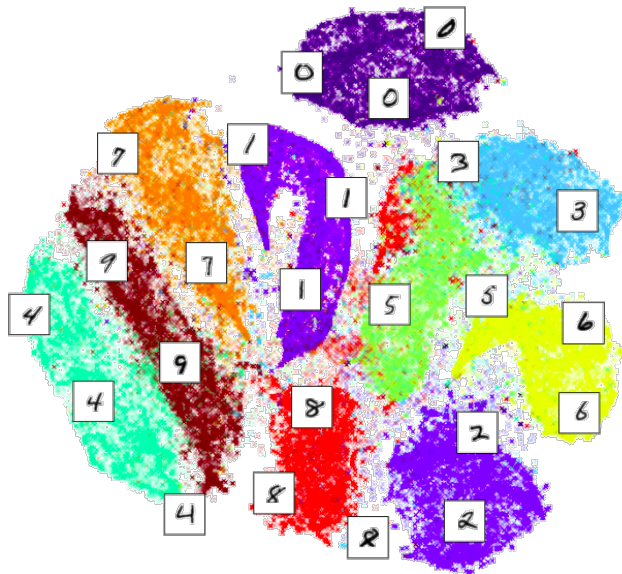
- Сегментация пользователей
- Поиск схожих пользователей

## Понижение размерности

- Задача: уменьшить количество признаков сохранив как можно больше информации.
- В случае размерности 2 или 3 получаем задачу визуализации

Пример:

- Из фотографии числа 30x30 (900 признаков) получить два числа (tSNE):



## **Поиск аномалий**

- Задача: определить объект, который не похож на остальных.
- Особенность: аномалий мало или их вообще нет

Перейдем к задаче обучения с учителем.

## Формальная постановка задачи обучения с учителем

$X$  - множество объектов

$Y$  - множество ответов

$f : X \rightarrow Y$  - неизвестная зависимость (целевая функция)

Дано:

$\{x_1, x_2, x_3, \dots, x_l\} \subset X$  - обучающая выборка

$y_i = f(x_i), \quad i = 1, \dots, l$  - известные ответы (может отсутствовать)

Найти:

$a : X \rightarrow Y$  - алгоритм (модель), решающую функцию, приближающую  $f$  на всем множестве  $X$ .

## Признаки

$x \in X$  - объект

$x = (x^1, x^2, \dots, x^d)$  - признаковое описание

$c_i \in M$  - конкретный признак, может быть разных типов

- $M = \{0, 1\}$  - бинарный
- $M$  - конечное множество - категориальный (номинальный)
- $M$  - конечное упорядоченное множество - порядковый
- $M = \mathbb{R}$  - количественный

Меняя множество ответов  $Y$  - получаем разные задачи. Рассмотрим их.

## Классификация

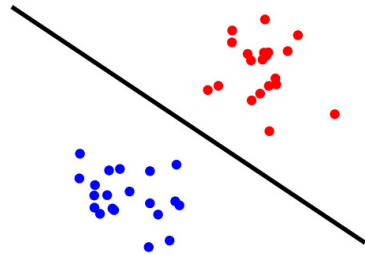
$Y = \{-1, +1\}$  - классификация на два класса (бинарная)

$Y = \{1, 2, \dots, M\}$  - классификация на  $M$  непересекающихся классов

$Y = \{0, 1\}^M$  - классификация на  $M$  пересекающихся классов

### Примеры

- Что изображено на картинке: кошка или собака
- Какое слово из заранее известных записано в аудиофайле
- Какие теги (из заранее известного множества) сопоставить картинке



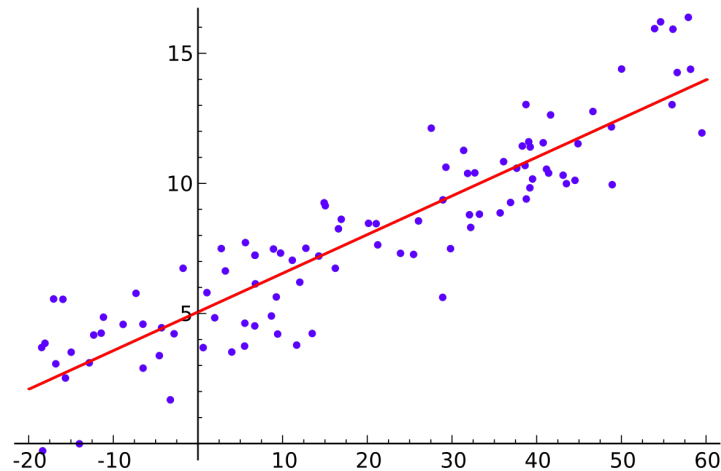
## Регрессия

$$Y = \mathbb{R}$$

$$Y = \mathbb{R}^m$$

### Примеры

- Какую ставить цену продажи для квартиры
- Прогноз погоды: температура, ветер, давление
- Предсказание спроса на товар в магазине



## Ранжирование

$Y$  - конечное упорядоченное множество.

Целевая функция  $f$  упорядочивает объекты из  $X$ .

Примеры:

- Как упорядочить поисковую выдачу?
- Какие фильмы рекомендовать посетителю?



## **Вопрос**

Нужно определить является ли данный прибор бракованным по его характеристикам

- К какой из задач это можно свести?

## Модель

Модель это семейство функций  $A = \{g(x, \theta) \mid \theta \in \Theta\}$ .  $\theta$  называют параметрами модели. При конкретном  $\theta$  получаем фиксированную функцию  $g(x)$ .

Построение фиксированной функции  $g$  по заданной обучающей выборке - это обучающий алгоритм (метод обучения). Обычно обучающий алгоритм тоже имеет параметры (например, скорость обучения) - их называют гиперпараметрами.

### Пример

- Линейная модель  $a(x, w) = \text{sign}(w_0 + w_1x^1 + w_2x^2 + \dots + w_dx^d)$
- $w$  - параметры модели

## Виды моделей

На практике используется большое количество разных типов моделей:

- линейные модели (линейная, логистическая регрессия)
- метрические (метод ближайших соседей)
- модели на основе решающих деревьев (случайный лес, бустинг)
- вероятностные модели
  - наивный байесовский классификатор
  - графические модели
  - марковские цепи
  - ...
- нейронные сети
- ...

Для разных задач хорошо себя показывают разные типы моделей (про это есть также теоретический результат - no free lunch theorem).

## Эмпирический риск

$L(x, a)$  - функция потерь: значение ошибки алгоритма  $a \in A$  на примере  $x \in X$ .

Эмпирический риск:  $Q(a, B)$  - ошибка алгоритма  $a \in A$  на выборке  $B \subset X$ .

Синонимы: функционал ошибки, ошибка алгоритма на выборке.

$$Q(a, B) = \sum_{x \in B} L(x, a)$$

## Эмпирический риск

Для классификации:

- доля неправильных ответов

Для регрессии:

- $Q(a, B) = \frac{1}{N} \sum_{i=1}^N |a(x_i) - y_i|$  - средняя абсолютная ошибка (MAE)
- $Q(a, B) = \frac{1}{N} \sum_{i=1}^N (a(x_i) - y_i)^2$  - средняя квадратичная ошибка (MSE)

## **Метод минимизации эмпирического риска**

Метод (принцип) минимизации эмпирического риска - искать тот алгоритм, который дает минимум  $Q(a, B)$  на данной выборке.

Это главный принцип, используемый в машинном обучении - он сводит задачу обучения к задаче оптимизации.

Пример: метод наименьших квадратов.

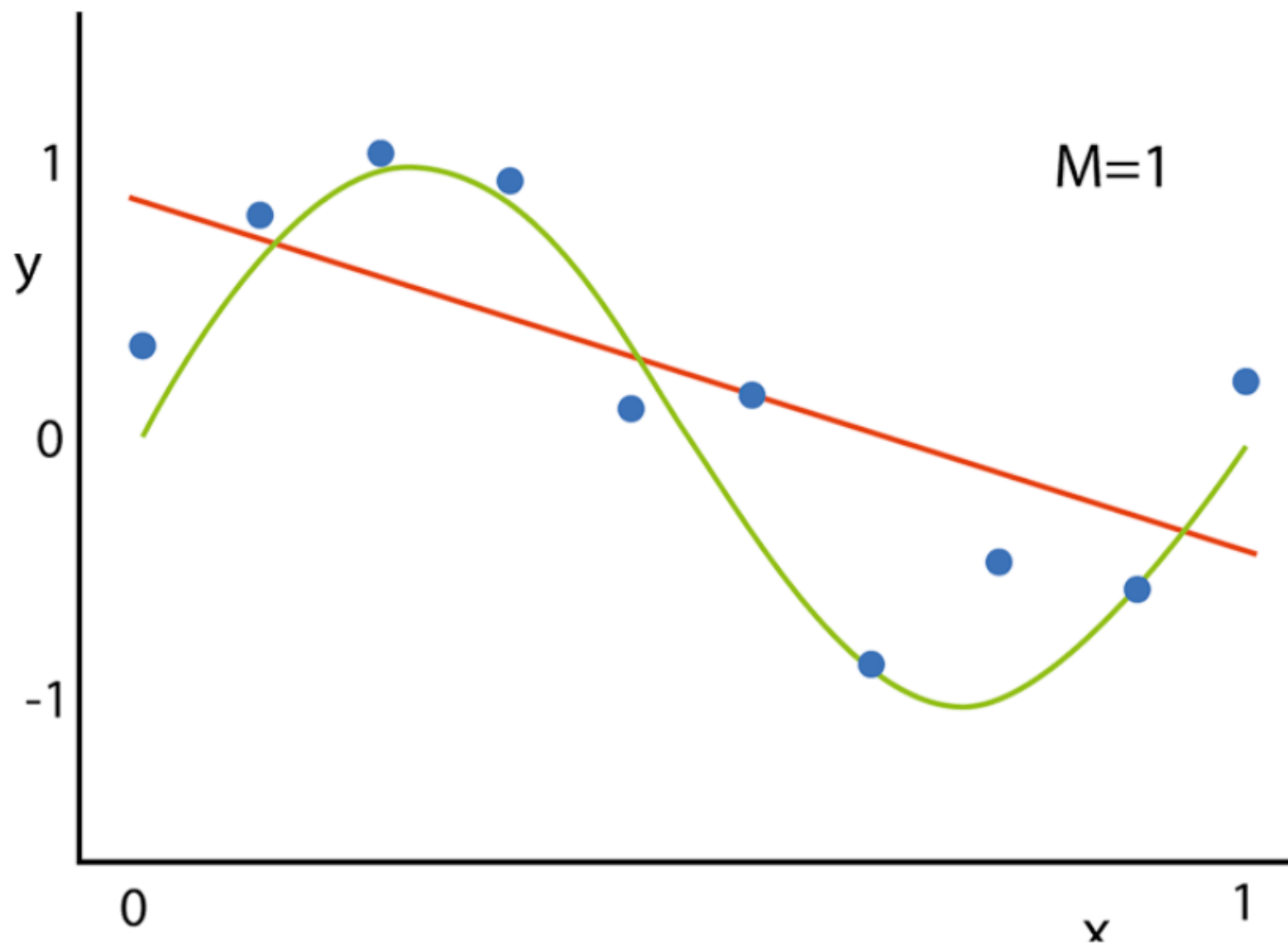
## Проблема обобщающей способности

Будет-ли алгоритм с маленькой ошибкой на множестве  $B$  давать маленькую ошибку на всем множестве  $X$ ?

Приблизит-ли он реальную закономерность или просто переобучится на множество  $B$ ?

## Недообучение

$$g(x) = w_0 + w_1x$$

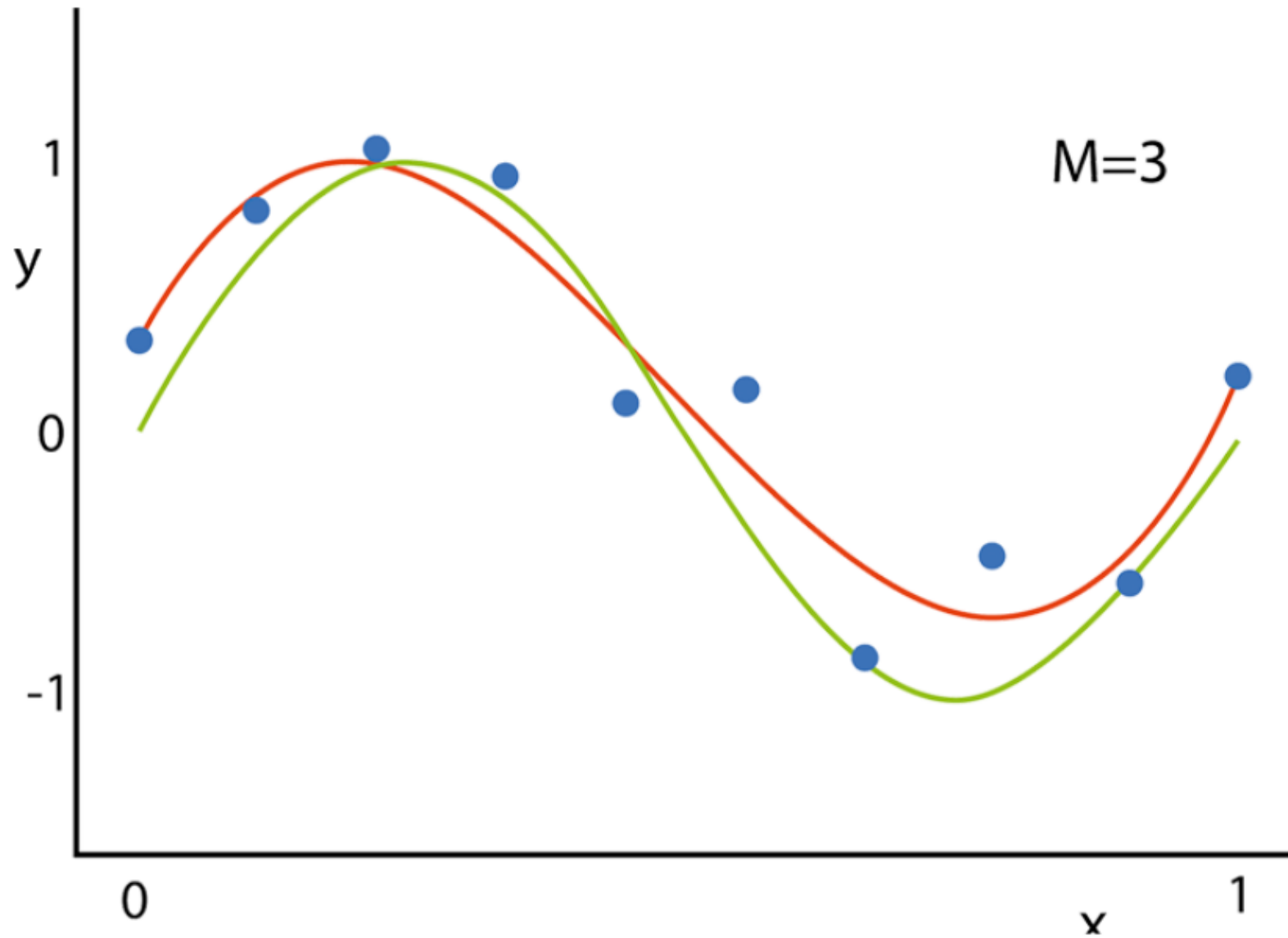






## Хороший результат

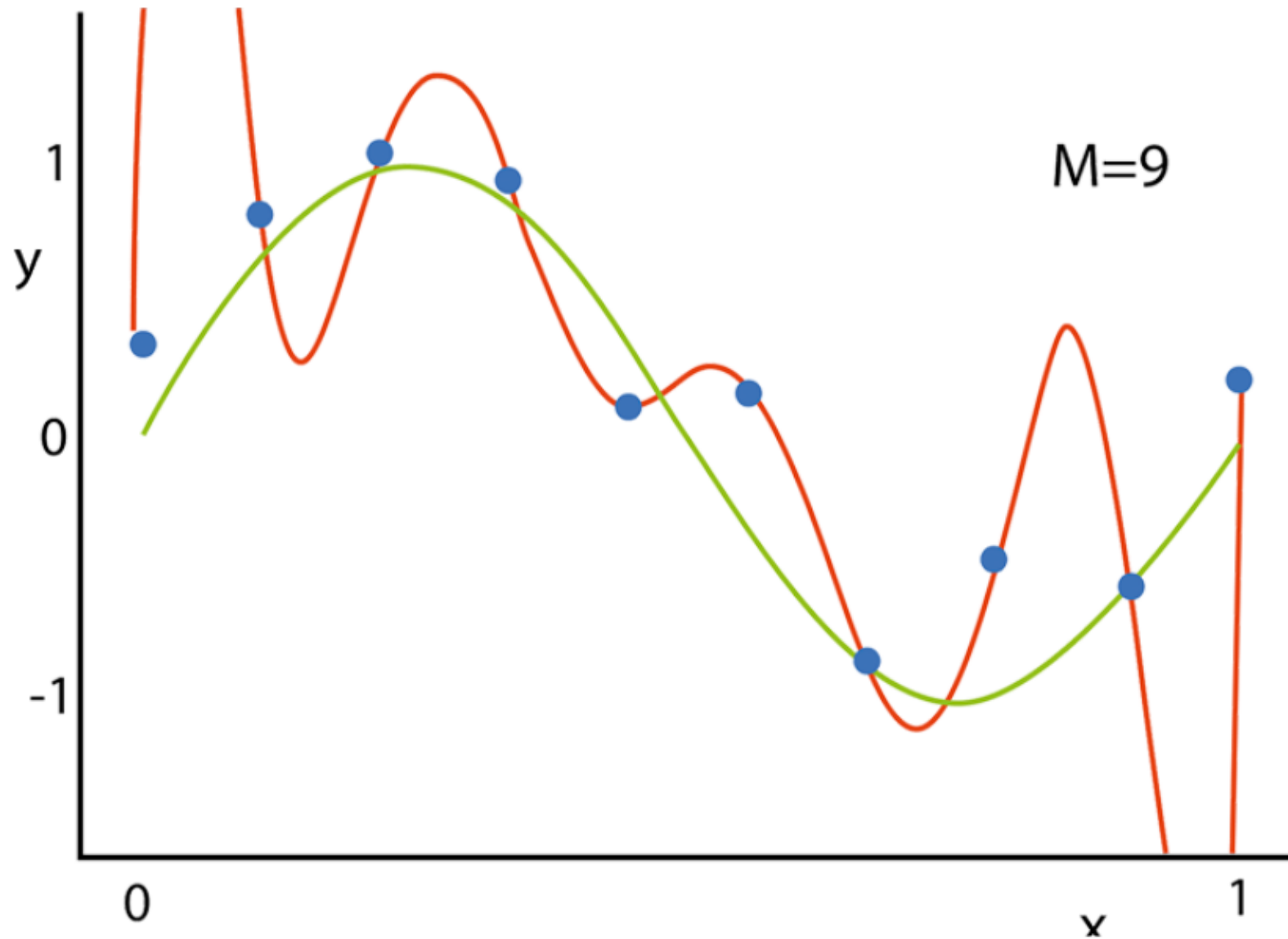
$$g(x) = w_0 + w_1x + w_2x^2 + w_3x^3$$





## Переобучение

$$g(x) = w_0 + w_1x + w_2x^2 + \dots + w_9x^9$$





## Переобучение

Переобучение есть всегда

- обучающая выборка конечна и не полна
- достоверно отличить случайные флуктуации от закономерностей на ней нельзя

*Но, также:*

- чрезмерная сложность модели (по количеству параметров, выразительной способности) поощряет подгонку под обучающее множество

## **Вопрос**

Можно-ли оценить переобучение по работе модели на обучающей выборке?

## **Как обнаружить переобучение**

Чтобы оценить переобучение мало обучающей выборки - нужны дополнительные данные. Способы:

- Разбиение выборки на две - обучающую и тестовую (hold-out set)
- Кросс-валидация



## Обучающая, валидационная и тестовые выборки

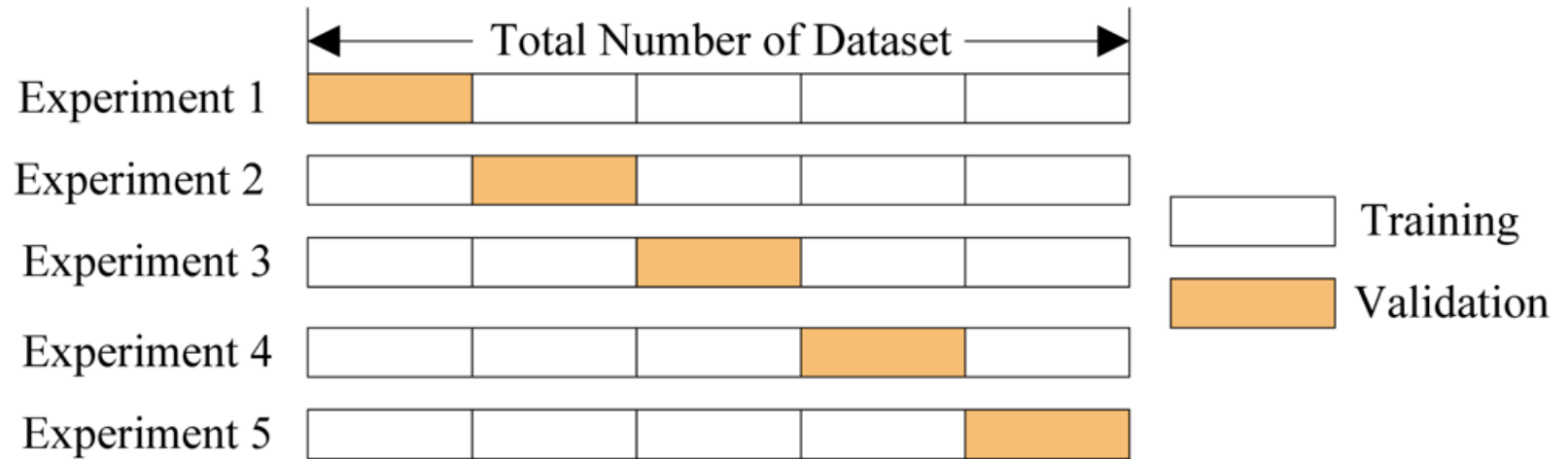
- На практике вместо разбиения на две, обычно разбивают на три выборки:
  - обучающая - используется чтобы обучить алгоритм
  - валидационная - для подбора гиперпараметров
  - тестовая - для итоговой оценки качества и обнаружения переобучения

Часто встречающиеся разбиения: 60/20/20.

## Кросс-валидация

Обучение набора моделей и их оценка на различных подмножествах обучающей выборки

Пример: K-fold кросс-валидация

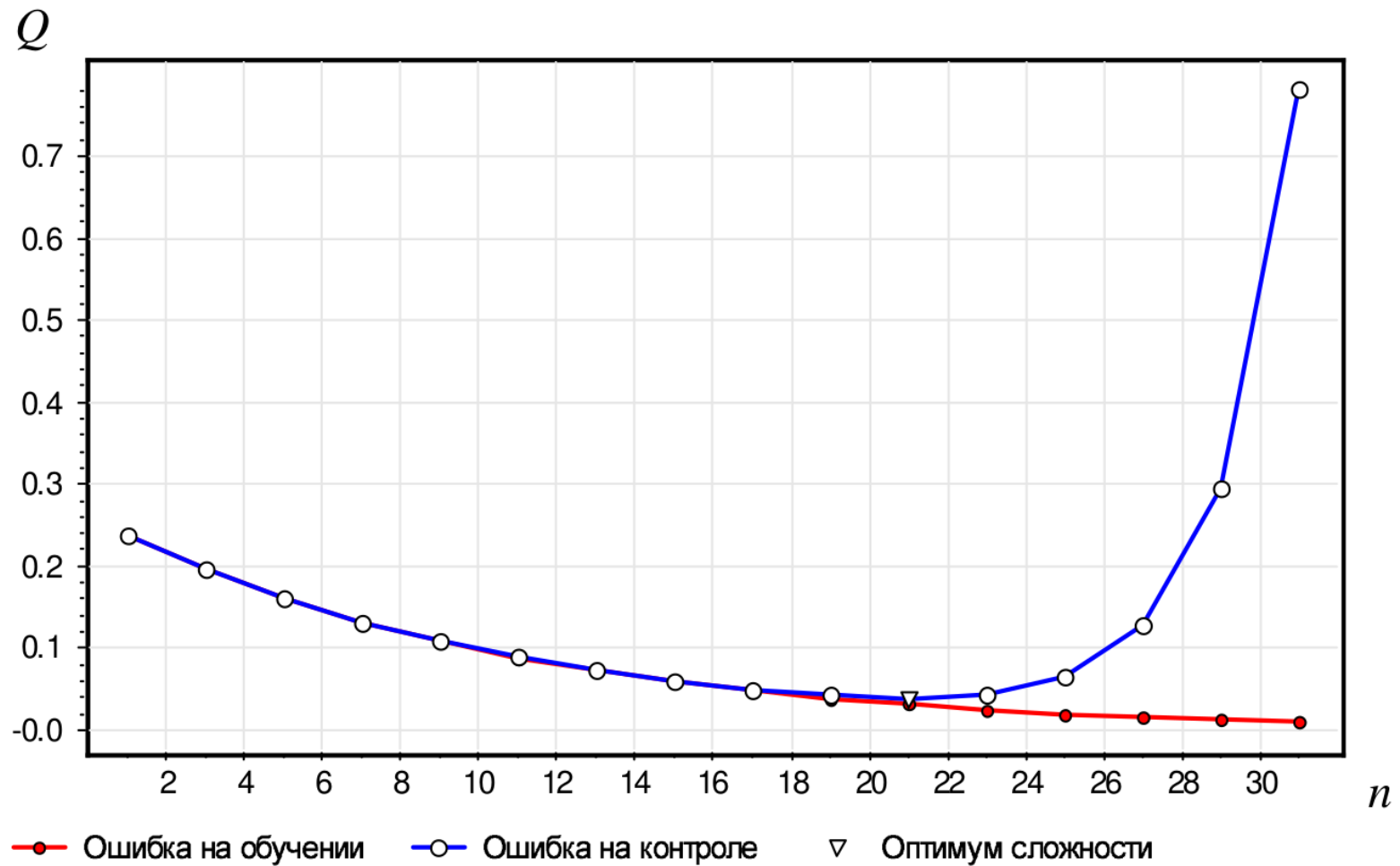


## Как бороться с переобучением

- Уменьшать мощность модели (как семейства функций  $A = \{g|\theta \in \Theta\}$ ):
  - выбрать более простую модель (другими словами более простой класс функций)
  - Ограничить параметры модели - это называется регуляризацией
  - Использовать валидационное множество/кросс-валидацию для настройки параметров регуляризации и гиперпараметров
- Увеличить обучающую выборку
- Строить ансамбли алгоритмов

## Обучающая, валидационная и тестовые выборки

На практике часто используется т.н. early stopping - во время обучения смотрят на ошибку модели на валидационном множестве и останавливают его когда ошибка растет несколько итераций подряд





**Задача: Кошка или собака?**



## **Задача: Кошка или собака?**

Дано: 25 000 изображений собак и кошек (фото собак и кошек приблизительно равное число).

Найти: Для данной фотографии сказать - на ней кошка или собака.

Вопрос: Какая это задача?

Вопрос: каким образом будем разбивать выборку?

## **Задача: Кошка или собака?**

Решаем задачу классификации.

Разобьем выборку в соотношении 60/20/20 на train, validation и test.

Классификатор: логистическая регрессия

Представление признаков:

1. Преобразуем изображения к черно-белому.
2. уменьшим изображения до размера 100x100 и возьмем вектор пикселей как вектор признаков.

Вопросы:

Как будем настраивать гиперпараметры классификатора?

Как оценим качество модели (её обобщающую способность)?



## Заключение

Общий вид процесса (пайплайна) обучения модели

1. Подготовить данные
2. Разделить обучающую выборку на train, validation и test
3. Подобрать гиперпараметры тренируя модель на train и оценивая качество на validation.
4. Обучить модель на train и validation.
5. Оценить качество модели на test.

Вместо пары множеств train и validation можно использовать кросс-валидацию.

Лекции (и задачи на практику) можно посмотреть тут: [https://github.com/frenzykryger/ssu\\_ds\\_course](https://github.com/frenzykryger/ssu_ds_course) ([https://github.com/frenzykryger/ssu\\_ds\\_course](https://github.com/frenzykryger/ssu_ds_course))