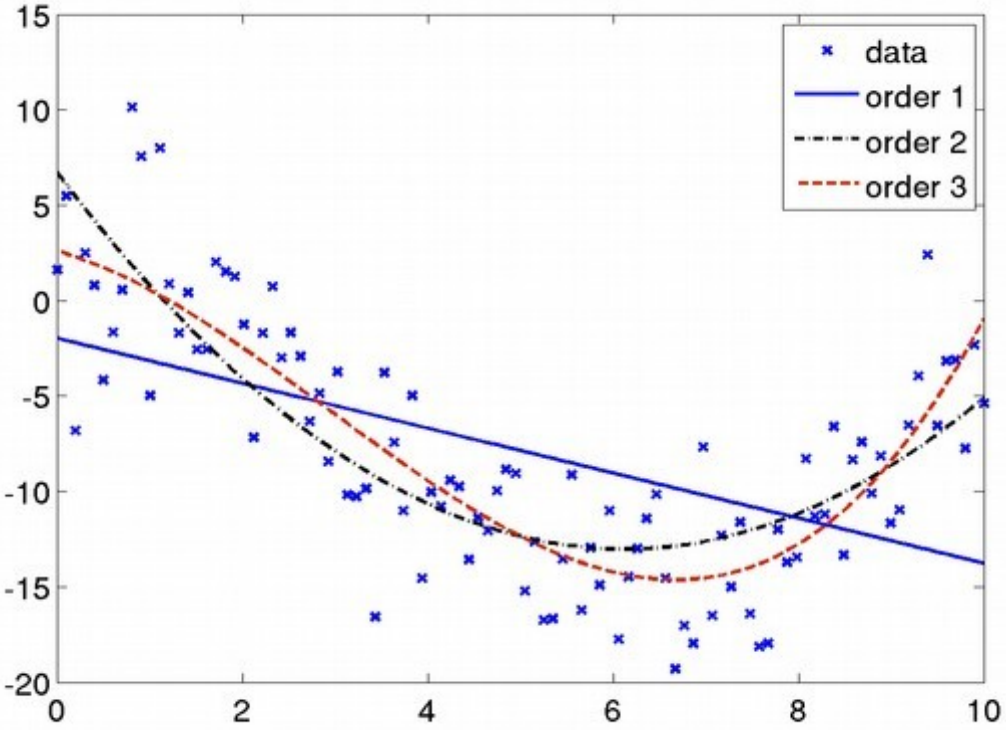
Машинное обучение
Основные понятия



Содержание лекции

- Задача обучения
- Матрица объектов-признаков
- Модель алгоритмов и метод обучения
- Функционал качества
- Вероятностная постановка задачи обучения
- Проблема переобучения

Литература

- http://www.machinelearning.ru
- Курс К.В.Воронцова
- https://www.kaggle.com/

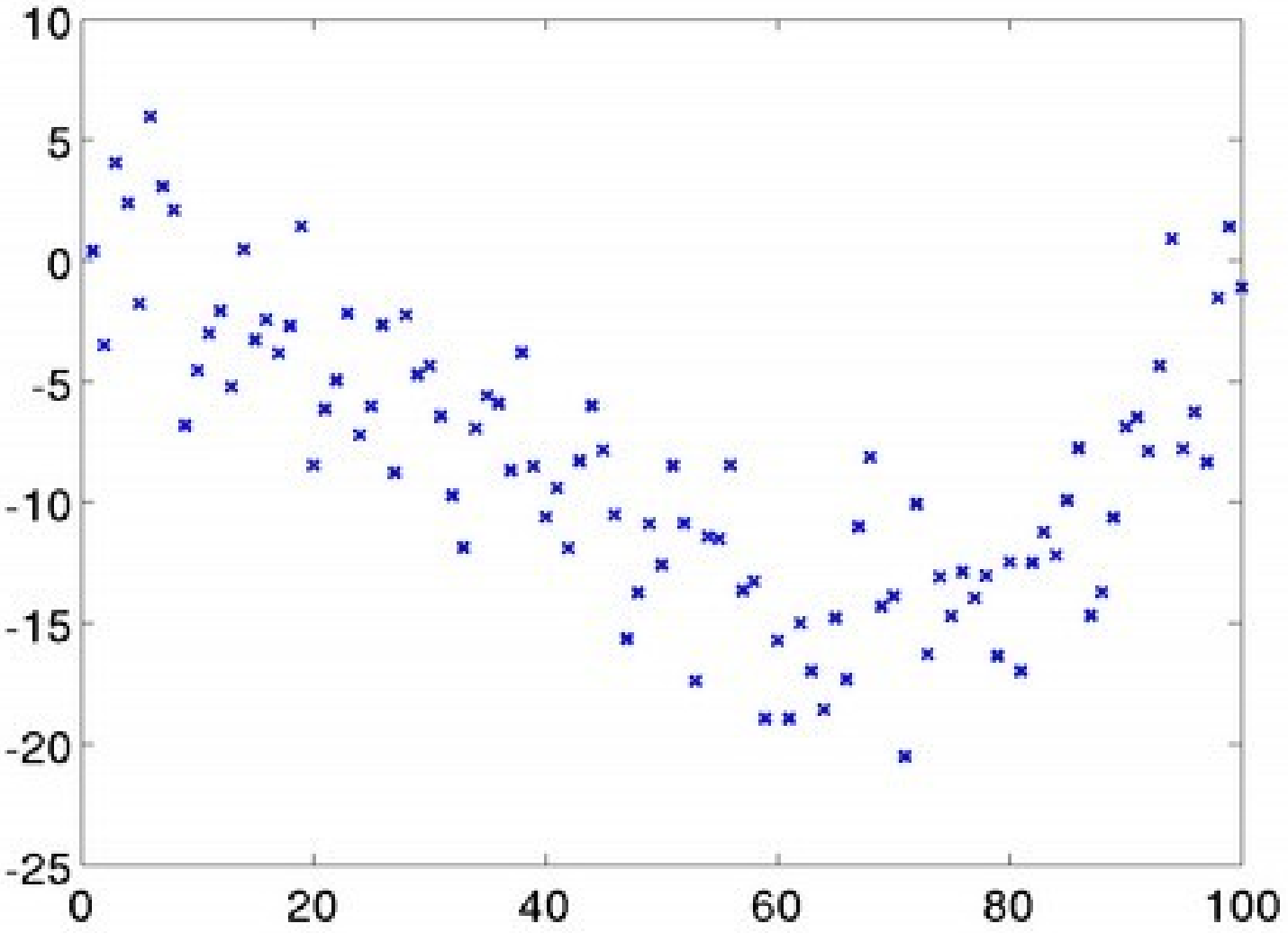
Задача обучения

X — множество объектов
Y — множество ответов
$y : X \to Y$ — неизвестная зависимость (target function)

Дано:

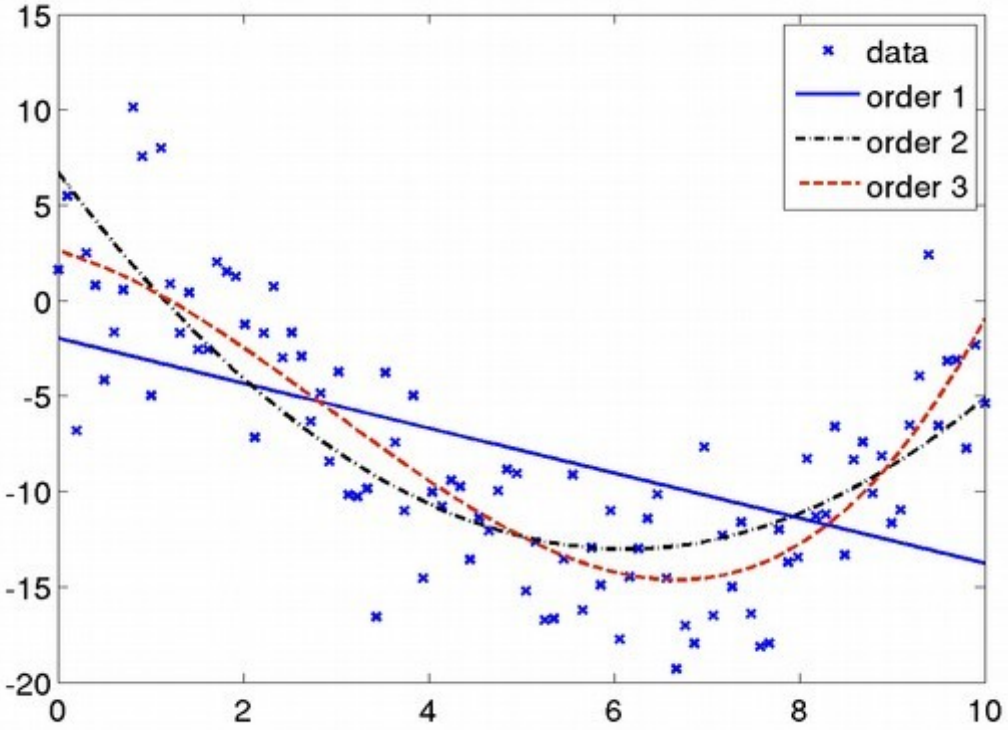
$\{x_1, \ldots, x_\ell\} \subset X$ — обучающая выборка (training sample)
$y_i = y(x_i)$, $i = 1, \ldots, \ell$ — известные ответы

Задача обучения



Задача обучения

Найти:
$a : X \to Y$ — алгоритм, решающую функцию (decision function), приближающую y на всём множестве X



Типы задач

Задачи классификации (classification):

$Y = \{-1, +1\}$ — классификация на 2 класса

$Y = \{1, \ldots, M\}$ — на M непересекающихся классов (multi-class classification)

$Y = \{0, 1\}^M$ — на M классов, которые могут пересекаться (multi-label classification).

Задачи восстановления регрессии (regression):

$Y = R$ или $Y = R^m$

Задачи ранжирования (ranking):

Y — конечное упорядоченное множество

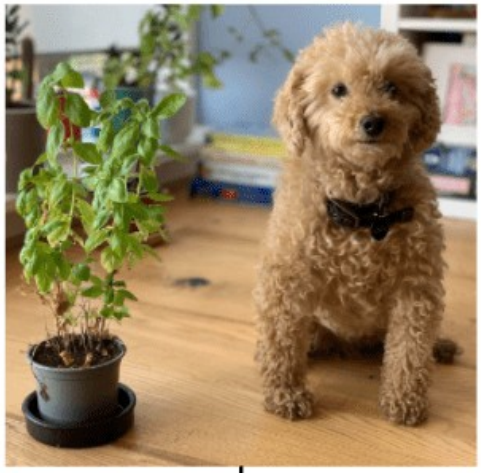
Типы классификаций

Binary Classification

Dog 0.9

Not Dog 0.1

Multiclass Classification



Dog 0.5

Cat 0.09

Bus 0.01

Plant 0.4

Multilabel Classification

Dog 0.8

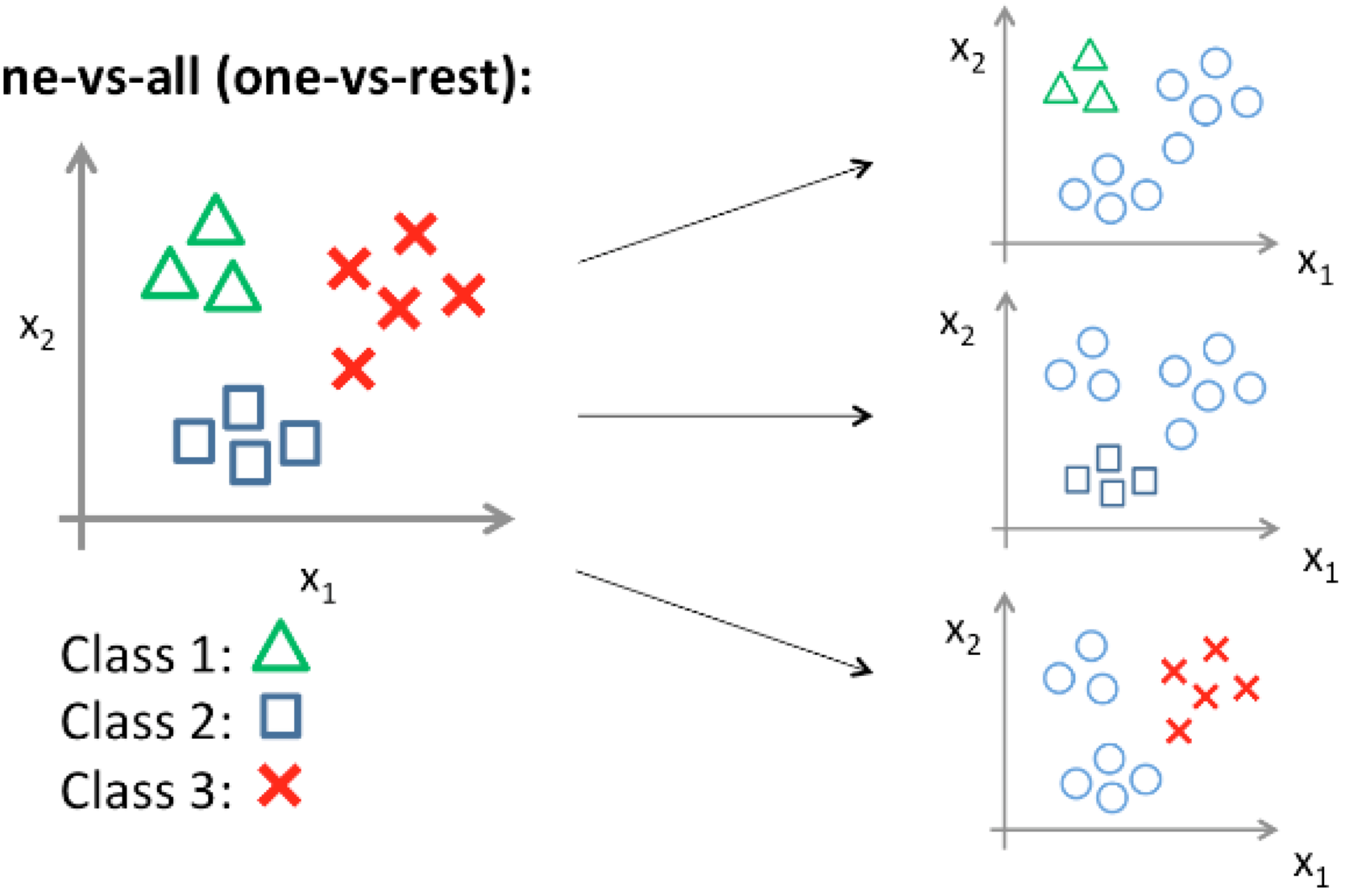
Cat 0.2

Bus 0.04

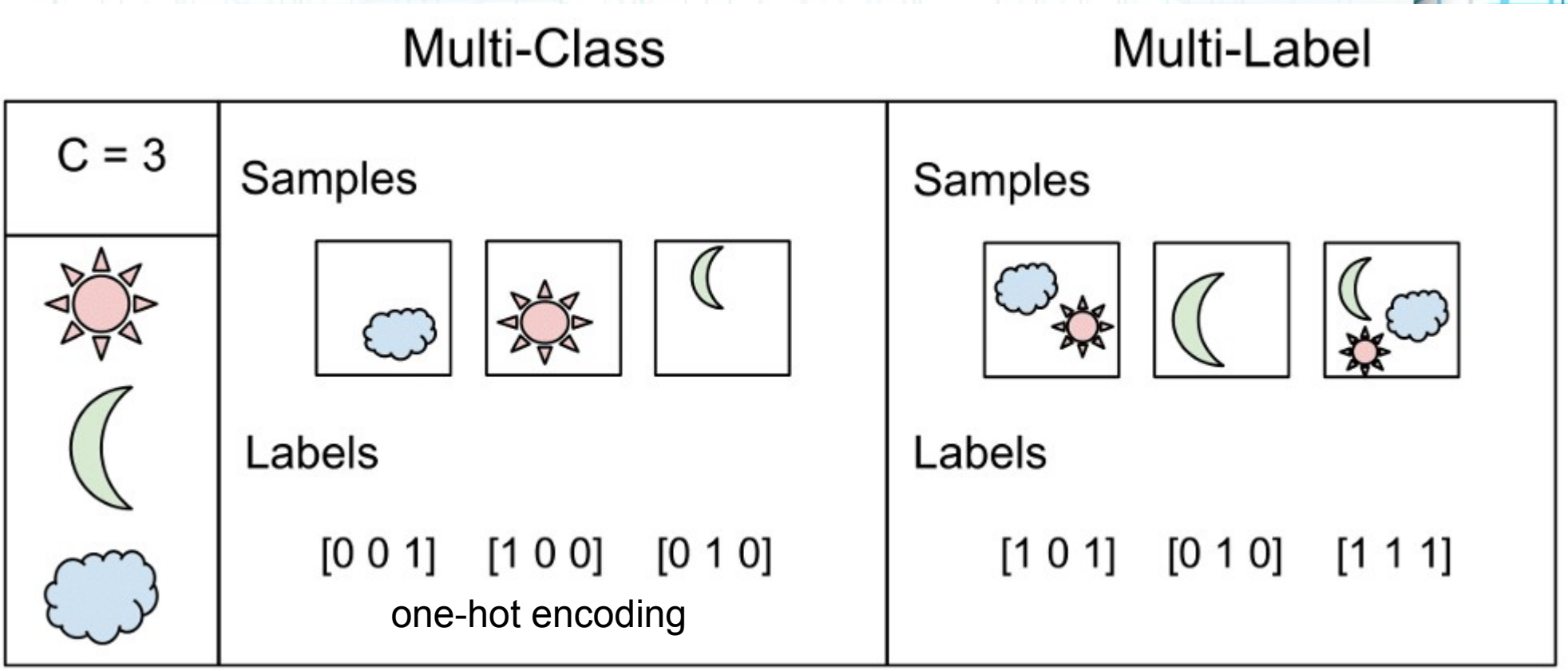
Plant 0.7

Сведение многоклассовой к бинарной классификации

One-vs-all (one-vs-rest):



Кодирование класса



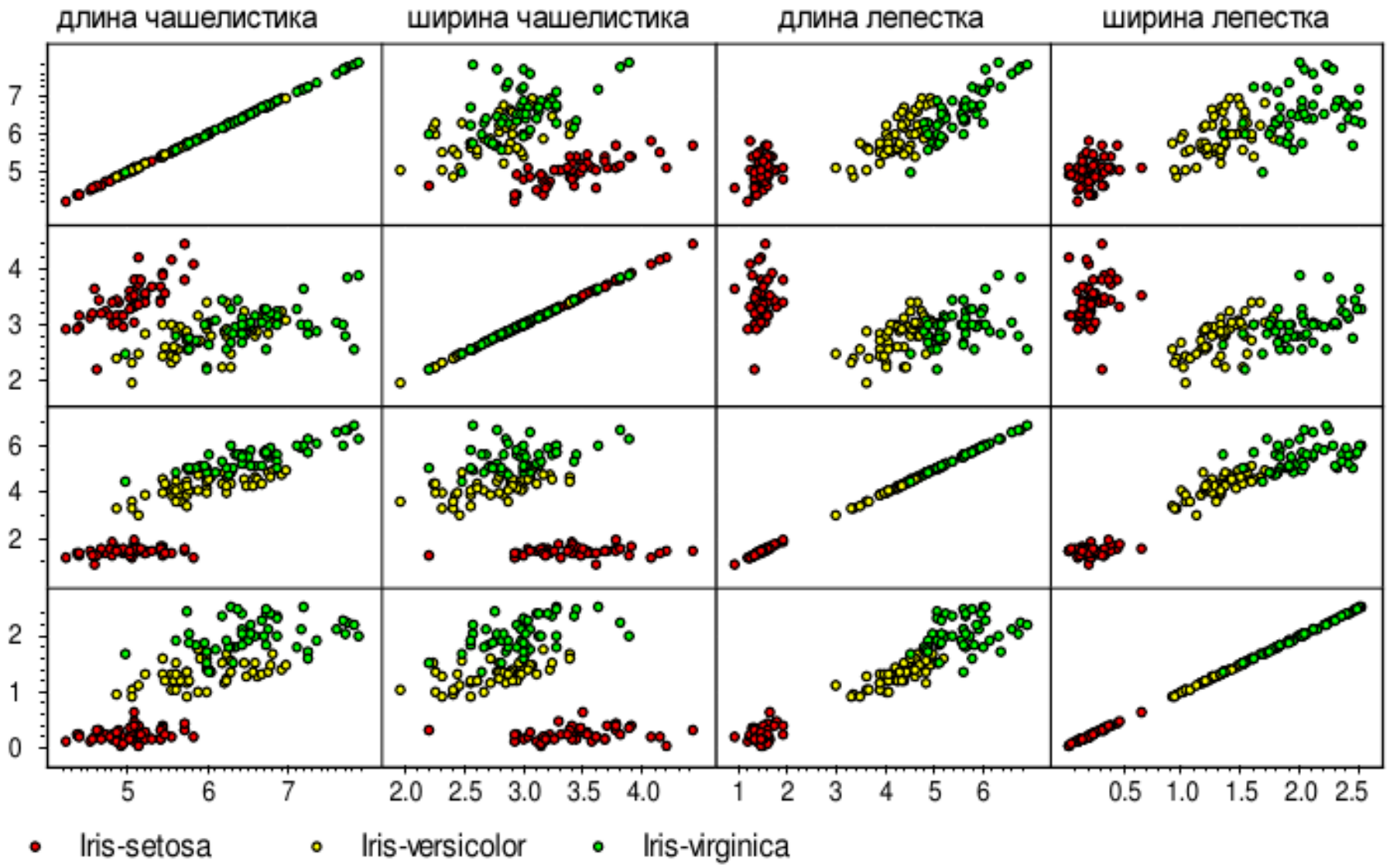
Признаки

- Компьютер всегда имеет дело с признаковым описанием объектов. Например: пациента можно описать признаками: имя, возраст, номер полиса, жалобы, давление, температура, результаты анализов
- $f : X \to D_f$
- Типы признаков:
  - бинарный
  - номинальный
  - порядковый
  - количественный

Матрица объектов-признаков:

$$\begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}$$

Пример. Задача классификации видов ириса (Фишер 1936)



Модель и алгоритм обучения

- **Модель** – это семейство “гипотез”

$$A = \{g(x, \theta) \mid \theta \in \Theta\}$$

одна из которых (как мы надеемся) хорошо приближает целевую функцию

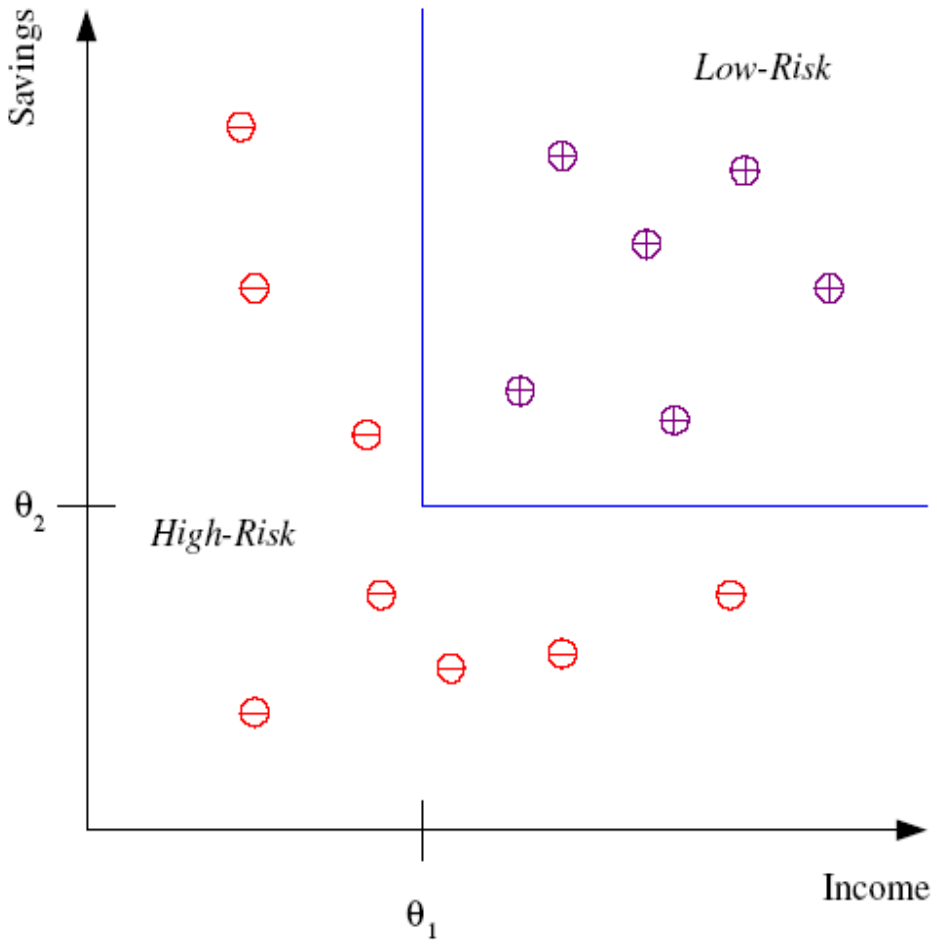
- **Алгоритм обучения**

$$\mu \colon (X \times Y)^{\ell} \to A$$

находит гипотезу в модели, которая наилучшим образом приближает целевую функцию, используя известные значения (обучающую выборку)

Пример - классификация

- Кредитный скоринг
- Разделение клиентов на low-risk и high-risk по их зарплате и сбережениям



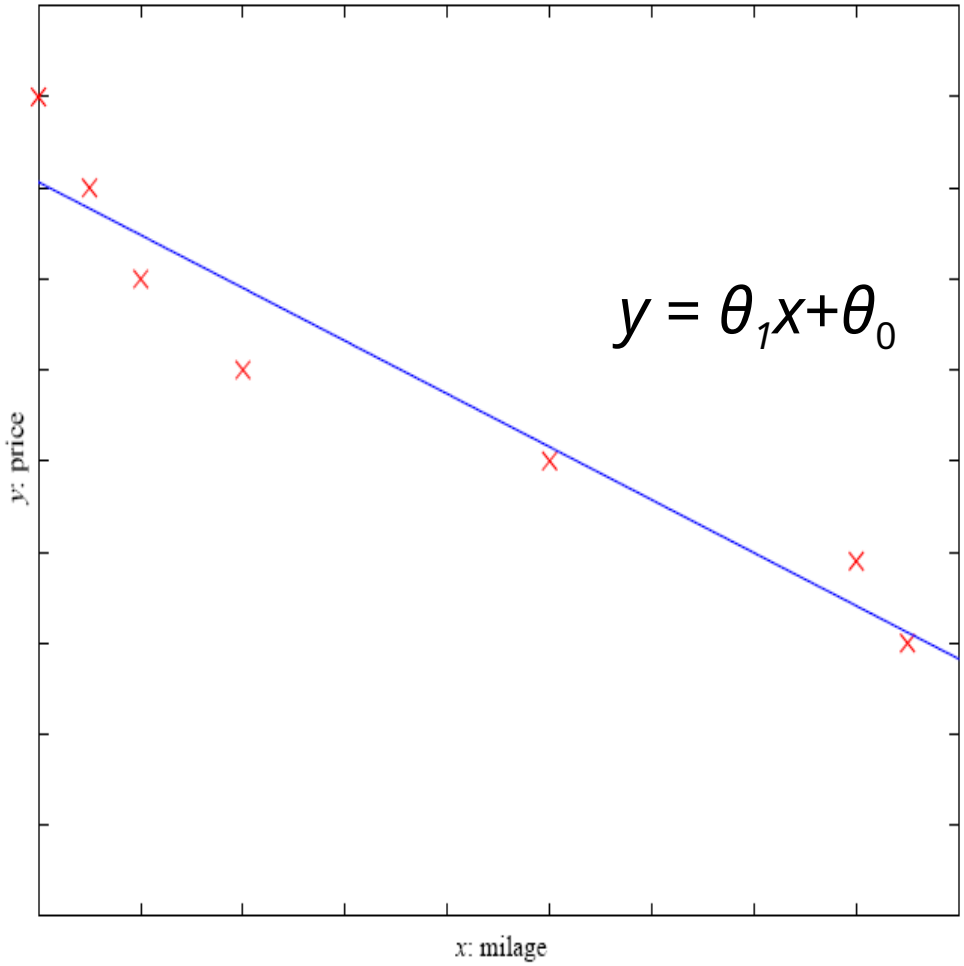
IF *income* > θ_1 AND *savings* > θ_2
THEN low-risk ELSE high-risk

Модель

14

Пример - регрессия

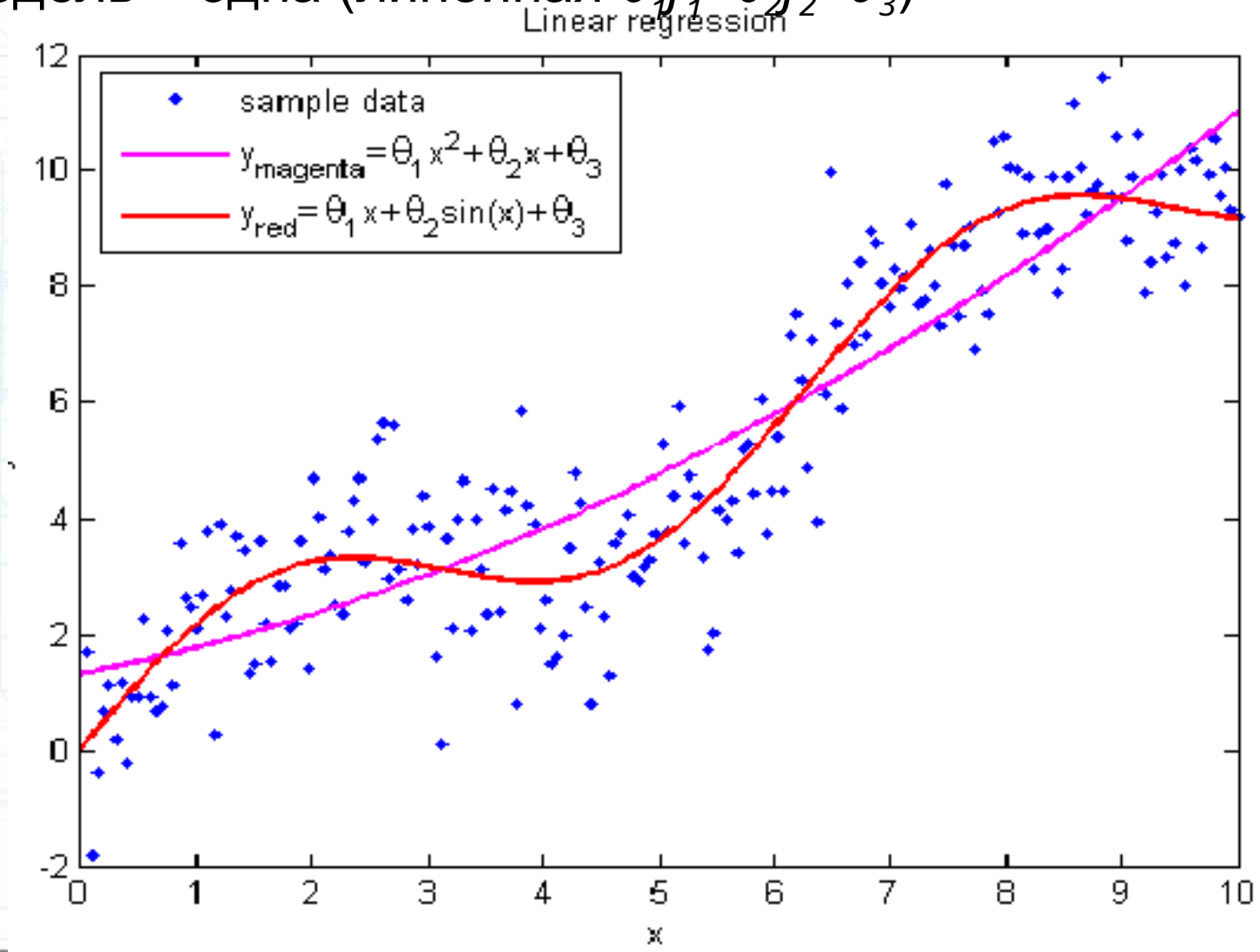
- y - цена автомобиля
- *x* – пробег
- $y = \theta_1 x + \theta_0$ – модель
- θ_0, θ_1 - *параметры*



15

Пример – две точки зрения

1. x – один признак, $\theta_1 x^2+\theta_2 x+\theta_3$ и $\theta_1 x+\theta_2 sin(x)+\theta_3$ – две модели
2. {x^2,x}, {x,sin(x)} – два набора разных признаков, модель – одна (линейная $\theta_1 f_1+\theta_2 f_2+\theta_3$)



Обучение на основе минимизации эмпирического риска

- Функция потерь $\mathscr{L}(a(x), y^*(x))$ - величина ошибки гипотезы a на объекте x. Примеры:
  - бинарная (где используется?)
  - $\mathscr{L}(a(x), y^*(x)) = |a(x) - y^*(x)|$
  - $\mathscr{L}(a(x), y^*(x)) = (a(x) - y^*(x))^2$
- Эмпирический риск: $Q(a, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}(a(x_i), y_i)$
- Самый популярный алгоритм обучения – минимизация эмпирического риска:

$$\mu(X^\ell) = \arg\min_{a \in A} Q(a, X^\ell)$$

Проблемы реальных задач

- Одинаковые признаковые описания могут соответствовать разным объектам
- Объекты с похожими (даже одинаковыми) значениями признаков могут иметь различные значения целевой функции
- Необходимый заказчику функционал качества (прибыль компании) не оптимизируется во время обучения модели

Вероятностная постановка задачи

- p(x,y) – неизвестная точная плотность распределения на X×Y
- X^ℓ - выборка из случайных, независимых и одинаково распределенных прецендентов
- $p(X^\ell) = p\big((x_1, y_1), \ldots, (x_\ell, y_\ell)\big) = p(x_1, y_1) \times \cdots \times p(x_\ell, y_\ell)$
- $\varphi(x, y, \theta)$ - модель
- Принцип максимума правдоподобия:

$$L(\theta, X^\ell) = \prod_{i=1}^{\ell} \varphi(x_i, y_i, \theta) \to \max_{\theta}$$

Decision function

- Предположим, что мы нашли вероятность p(y|x)=p(x,y)/p(x). Какое значение y нужно предсказать для заданного x ?
- Минимизация среднего риска:

$$a(x) = \arg\min_{s} E_y \mathscr{L}(s, y)$$

- Упражнение:

y	2	3	4	5
p(y|x)	0.1	0.2	0.3	0.4

примите правильные решения a(x) для каждой функции потерь со слайда 14

Расчет потерь/прибыли

Банк решает вопрос о выдаче кредита 1 млн. руб клиенту под 15% годовых. Алгоритм МО предсказал вероятность возврата кредита: 80%

Прибыль	Вернёт	Не вернёт	Средняя прибыль
Вероятность	0.8	0.2	
Выдать	150 т.р.	-1000 т.р.	-80 т.р.
Не выдать	-150 т.р.	0	-120 т.р.

Правильно?

Расчет потерь/прибыли

Банк решает вопрос о выдаче кредита 1 млн. руб клиенту под 15% годовых. Алгоритм МО предсказал вероятность возврата кредита: 80%

Прибыль	Вернёт	Не вернёт	Средняя прибыль
Вероятность	0.8	0.2	
Выдать	150 т.р.	-1000 т.р.	-80 т.р.
Не выдать	-150 т.р.	1000 т.р.	80 т.р.

Если мы учитываем упущенную выгоду, то нужно учитывать и упущенные потери.

Проблемы реальных задач: имеющиеся в распоряжении данные не позволяют оценить упущенную выгоду. Там есть информация только о клиентах, которым кредит выдали.

Расчет потерь/прибыли

Прибыль	Вернёт	Не вернёт	Средняя прибыль
Вероятность	0.8	0.2	
Выдать	150 т.р.	-1000 т.р.	-80 т.р.
Не выдать	-150 т.р.	1000 т.р.	80 т.р.

Ответьте на вопросы:

1) Можно ли оценить прибыль/потери в случае не выдачи кредита?

2) Являются ли правильными вероятности 0.8 и 0.2, возвращаемые новым алгоритмом, натренированным на выборке «принятых» старым алгоритмом банка клиентов?

Подсказка: рассмотрите две ситуации:
a) мы внедряем наш алгоритм, не убирая старый, а после старого вторым этапом;
b) мы внедряем новый алгоритм на замену старому

Расчет потерь/прибыли

Прибыль	Вернёт	Не вернёт	Средняя прибыль
Вероятность	0.8	0.2	
Выдать	150 т.р.	-1000 т.р.	-80 т.р.
Не выдать	-150 т.р.	1000 т.р.	80 т.р.

Проблемы реальных задач 2: после внедрения нашего алгоритма любым из двух упомянутых способов статистика работы банка изменится и мы получим третий вариант вероятностей*.

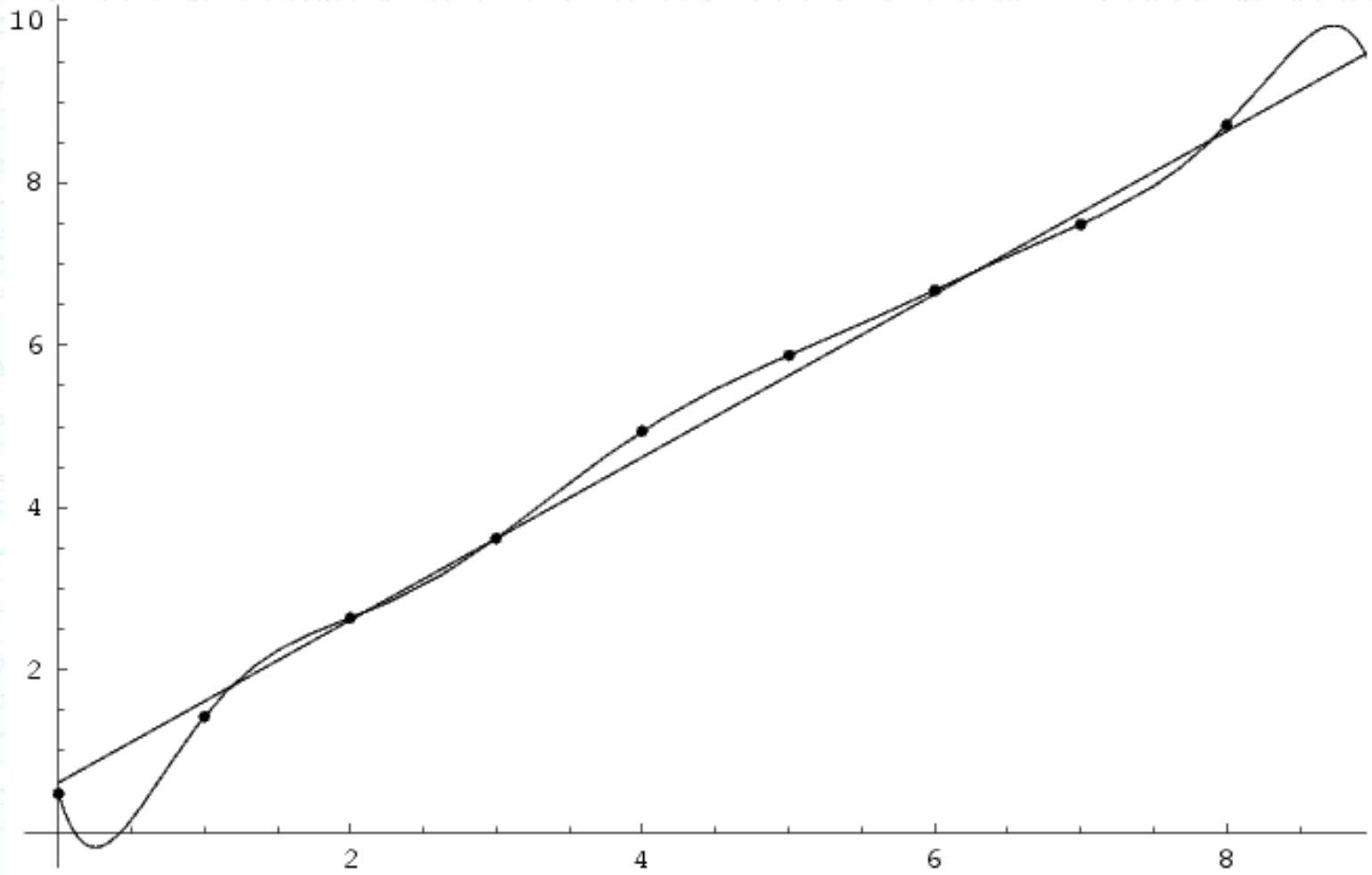
Ответьте на вопрос: Нужно ли каждый год регулярно обновлять алгоритм на новой статистике и как это делать? Сойдется ли этот процесс когда-нибудь к стационарному алгоритму и статистике?

1 вариант: вероятности на выходе старого алгоритма; 2-й — на входе; 3-й — на выходе нового алгоритма

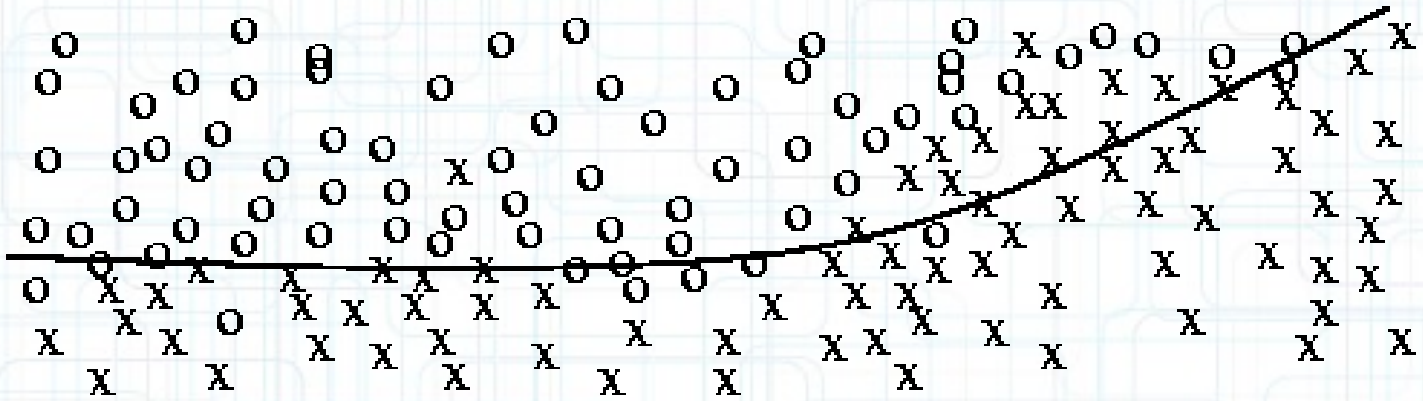
Степени обученности модели

- Недообученная модель
  - Модель, слишком сильно упрощающая закономерность $X \rightarrow Y$.
- Переобученная модель
  - Модель, слишком сильно настроенная на особенности обучающей выборки (на шум в наблюдениях), а не на реальную закономерность $X \rightarrow Y$.

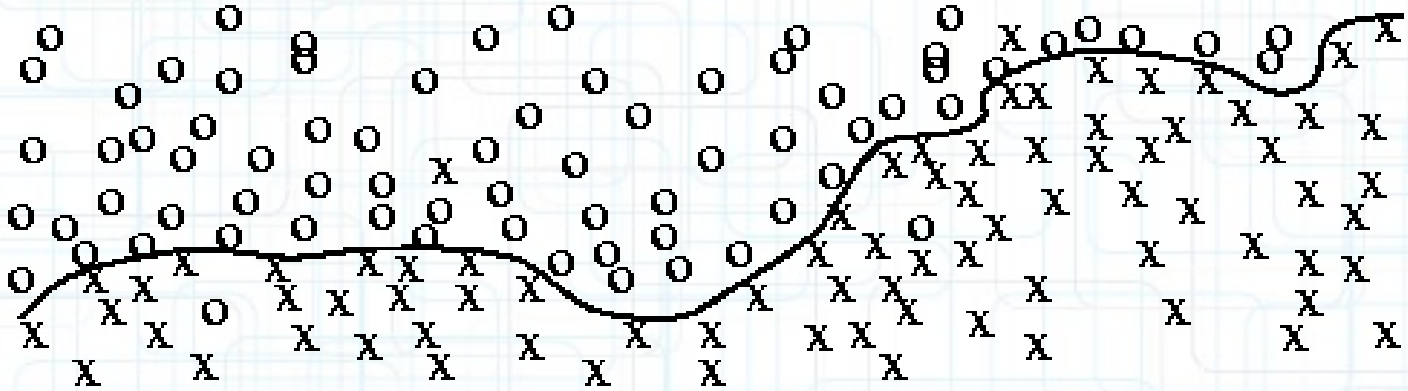
Переобучение



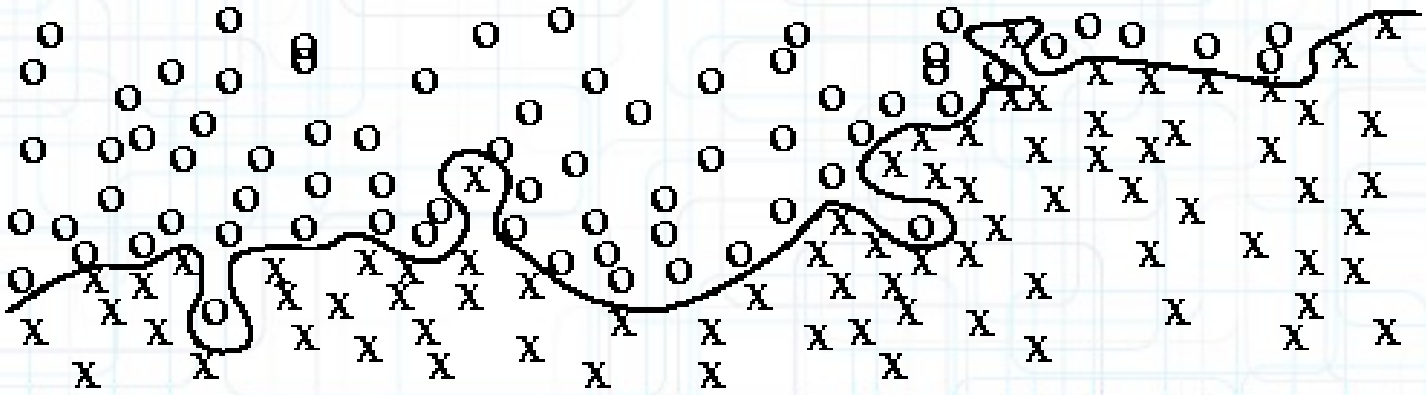
Переобучение



Under-Trained

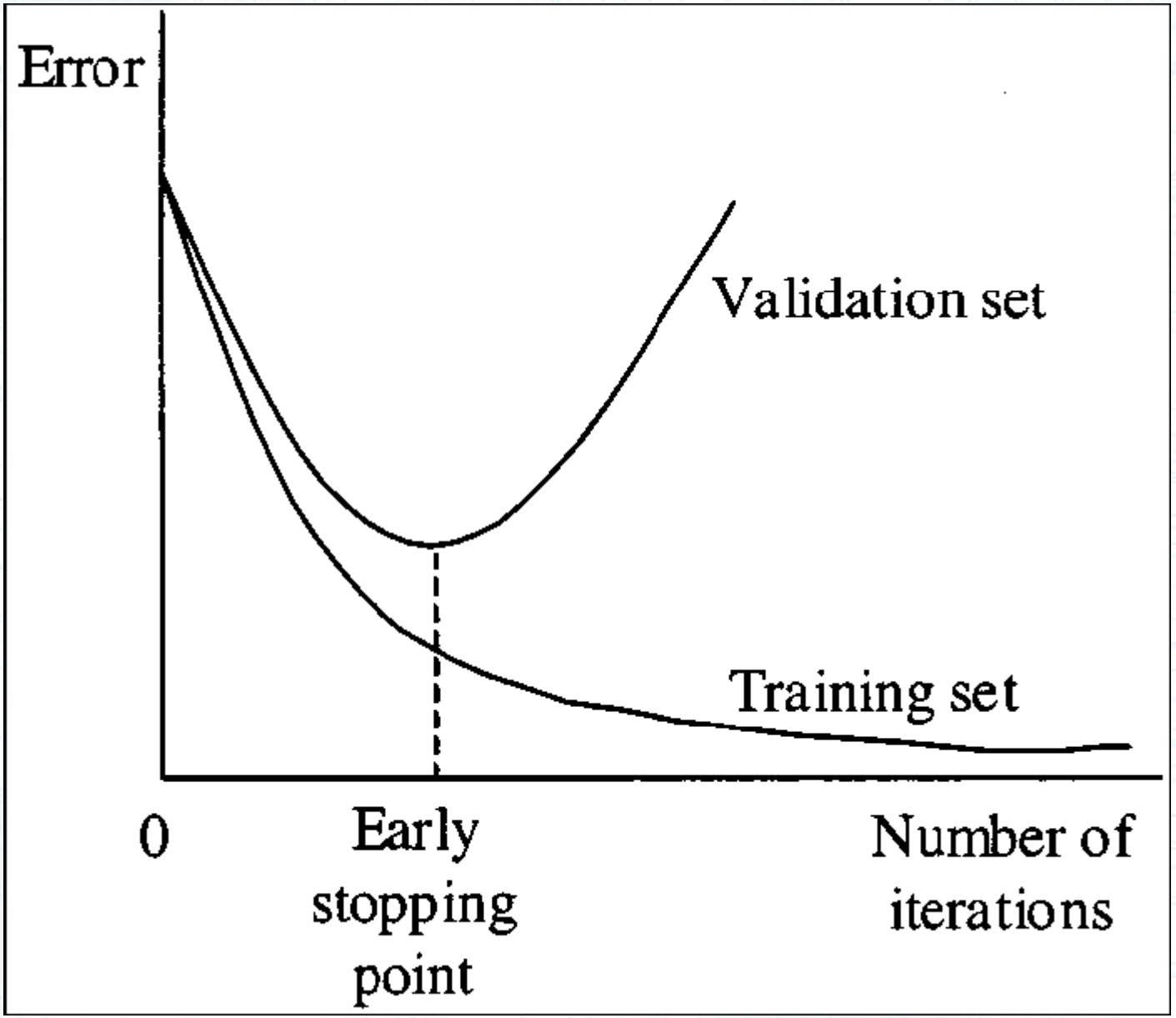


Well-Trained



Overfitted

Когда нужно заканчивать обучаться?



Контроль переобучения

- Для оценки обобщающей способности алгоритма обучения μ используют:
  - Эмпирический риск на тестовых данных (hold-out):

$$\mathrm{HO}(\mu, X^\ell, X^k) = Q(\mu(X^\ell), X^k) \to \min$$

  - Скользящий контроль (leave-one-out), L=l+1:

$$\mathrm{LOO}(\mu, X^\ell) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathscr{L}\Big(\mu(X^\ell \setminus \{x_i\})(x_i), y_i\Big)$$

  - Кросс-проверка (cross-validation):

$$\mathrm{CV}(\mu, X^L) = \frac{1}{|N|}\sum_{n\in N} Q(\mu(X_n^\ell), X_n^k) \to \min$$

  - Оценка вероятности переобучения:

$$Q_\varepsilon(\mu, X^L) = \frac{1}{|N|}\sum_{n\in N}\Big[Q(\mu(X_n^\ell), X_n^k) - Q(\mu(X_n^\ell), X_n^\ell) \geqslant \varepsilon\Big] \to \min$$