

# Машинное обучение: оценка качества

Н. Поваров, И. Куралёнок

СПб,  
2018

# Задача

Хотим понять хорошо ли будет работать решающая функция на практике.

*“If you can't measure it, you can't improve it”*

**Lord Kelvin**

*“Гораздо легче что-то измерить, чем понять, что именно вы измеряете.”*

**Джон Уильям Салливан**

# График целевой метрики





# Вспомним о чём ML

- $$\operatorname{argmax}_{F, B: F_0 = B(F)} A(\Gamma, F_0)$$

$A$  — цели эксплуатации (например деньги) на всей области работы

$B^*$  — способ оптимизации, который реализуем

\* — здесь прячутся  $D$  и  $T$ .

# Грубая классификация способов оценки

- Black Box методы
  - Online
  - Offline
- Glass Box методы
  - VC-оценки
  - PAC-Bayes bounds
  - Оценки по Воронцову

# Грубая классификация способов оценки

- Black Box методы
  - Online
  - Offline
- Glass Box методы
  - VC-оценки
  - PAC-Bayes bounds
  - Оценки по Воронцову

# Online

- Наблюдение
- Эксперимент

# Online

- Наблюдение
- Эксперимент

# Online

- ~~Наблюдение~~
- Эксперимент

# Online

- + В условиях эксплуатации
- + В положительные результаты эксперимента обычно верят
- + Легко хвастаться результатом
  
- Вряд ли цель эксплуатации
- Можно навредить пользователям

# Offline

- + Нельзя навредить пользователям
- + Обычно можно проводить сильно больше экспериментов
- Обычно нужны данные (примеры)
- Сложно «хвастаться» результатом

# Оценка качества как система принятия решений Natur

		<b>e</b>	
		<b>P</b>	<b>N</b>
<b>СП</b>	<b>P</b>	<b>TP</b>	<b>FP</b>
<b>P</b>	<b>N</b>	<b>FN</b>	<b>TN</b>

# Offline

# Offline на данных

- Hold Out
- Cross-fold Validation
- Bootstrap

# Hold-out

Можно поделить множество на две части и обучить на одной, а оценить на другой:

$$DS = L \cup T, \quad L \cap T = \emptyset$$

- + Расскажет о качестве предсказания
- + Можно посмотреть на качество на L и на T
- Использует меньше данных в обучении
- Если исходное множество непоказательно, то всё плохо

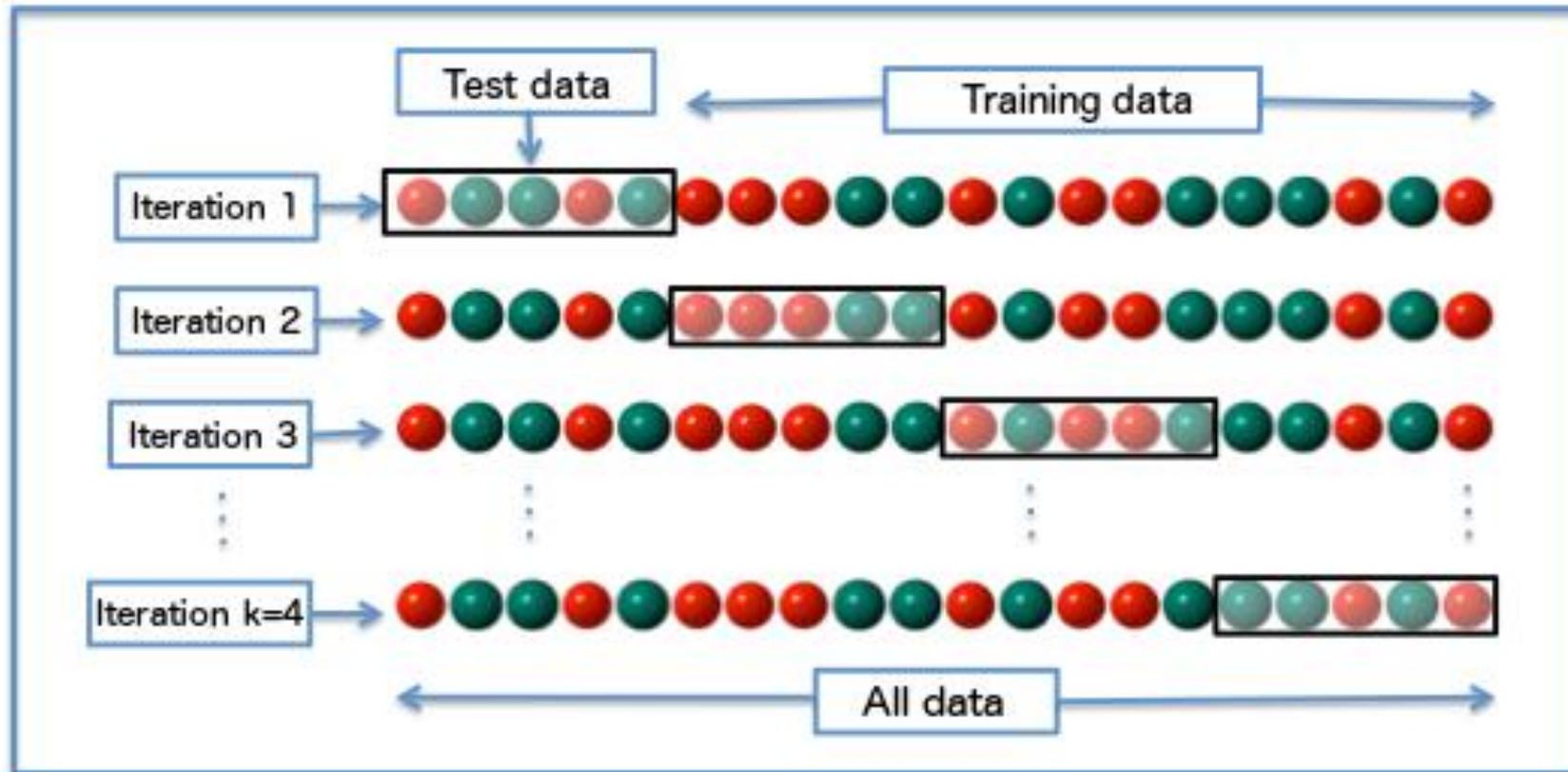
# Cross-fold Validation

Можно поделить множество на  $k$  частей и обучить на  $k-1$ , а оценить на оставшейся:

$$DS = L \cup T, \quad L \cap T = \emptyset$$

Проделать так  $k$  раз.

# Cross-fold Validation



# Cross-fold Validation

Можно поделить множество на  $k$  частей и обучить на  $k-1$ , а оценить на оставшейся:

$$DS = L \cup T, \quad L \cap T = \emptyset$$

Проделать так  $k$  раз.

- + Расскажет о качестве предсказания
- + Можно посмотреть на качество на  $L$  и на  $T$
- Использует меньше данных в обучении
- Если исходное множество непоказательно, то всё плохо
- Надо обучающим показать всё множество
- Живёт такая система не очень долго

# Повторные выборки

Можно посэмпливать.

- $|L| = |T| = |D|$
- Random sampling with replacement

+ Объем в обучении максимальный

+ T можно делать много раз

- Между L и T возникли зависимости

# Где в offline система принятия решений

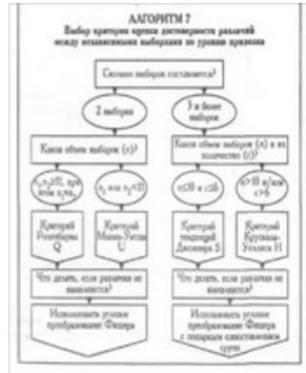
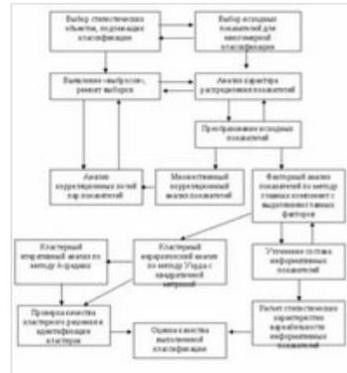
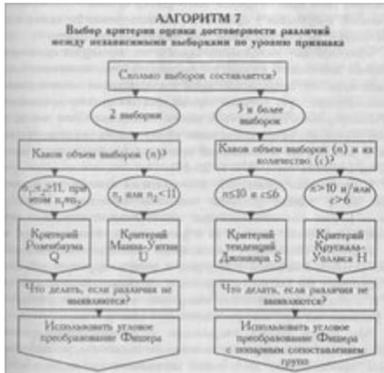
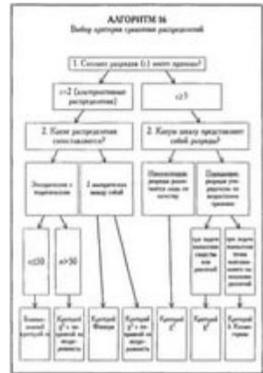
		Верная гипотеза	
		$H_0$	$H_1$
Результат применения критерия	$H_0$	$H_0$ верно принята	$H_0$ неверно принята (Ошибка второго рода)
	$H_1$	$H_0$ неверно отвергнута (Ошибка первого рода)	$H_0$ верно отвергнута

# Где в offline система принятия решений

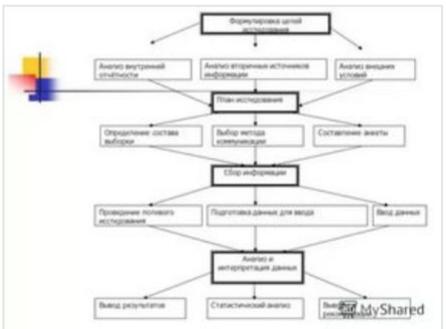
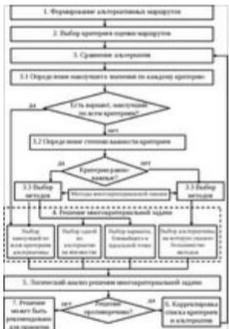
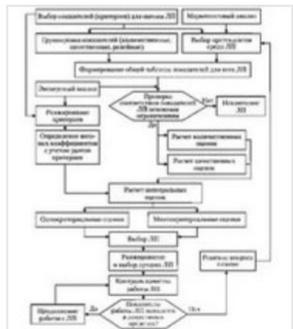
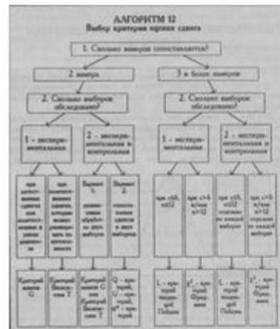
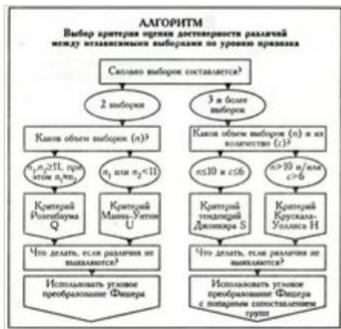
- В случае hold-out — «много» наблюдений в T.
- В случае cross-fold — большое k.
- В случае повторных выборок — можно сделать много раз.

		Верная гипотеза	
		$H_0$	$H_1$
Результат применения критерия	$H_0$	$H_0$ верно принята	$H_0$ неверно принята (Ошибка второго рода)
	$H_1$	$H_0$ неверно отвергнута (Ошибка первого рода)	$H_0$ верно отвергнута

# Как выбрать статтест?

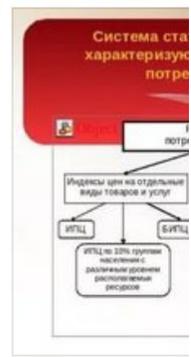


4.14  
Выявление  
Выявление  
Сравнение  
Принятие



**Таблица 4.20**  
Результаты дисперсионного анализа

Выборки	Среднее	Дисперсия	Среднее	Дисперсия
1	10,0	1,0	10,0	1,0
2	10,0	1,0	10,0	1,0
3	10,0	1,0	10,0	1,0
4	10,0	1,0	10,0	1,0
5	10,0	1,0	10,0	1,0
6	10,0	1,0	10,0	1,0
7	10,0	1,0	10,0	1,0
8	10,0	1,0	10,0	1,0
9	10,0	1,0	10,0	1,0
10	10,0	1,0	10,0	1,0



# Классический подход что надо ПОМНИТЬ

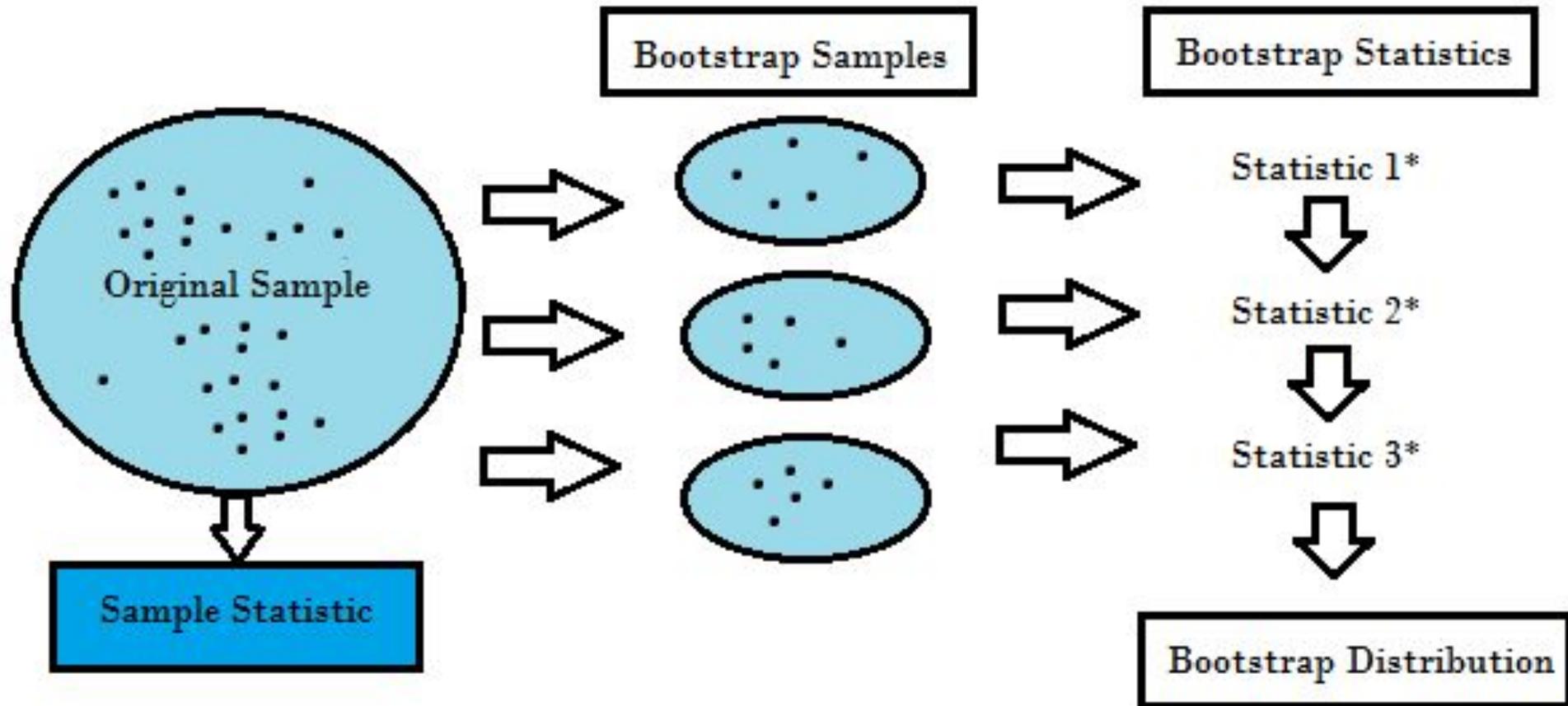
- Направленность
- Шкала измерений (отношений, порядка, номинальная, бинарная)
- Связанные/независимые наблюдения
- Параметрические/непараметрические
- ...

**Есть метод проще!**

# Правильный подход на примере hold-out

- A **bootstrap sample** is a random sample taken with replacement from the original sample, of the same size as the original sample.
- A **bootstrap statistic** is the statistic computed on bootstrap sample.
- A **bootstrap distribution** is the distribution of many bootstrap statistic.

# Правильный подход на примере hold-out



**Common knowledge**

Model Selection

Обобщающая  
способность  
алгоритма  
Flexibility

Bias-Variance trade-off

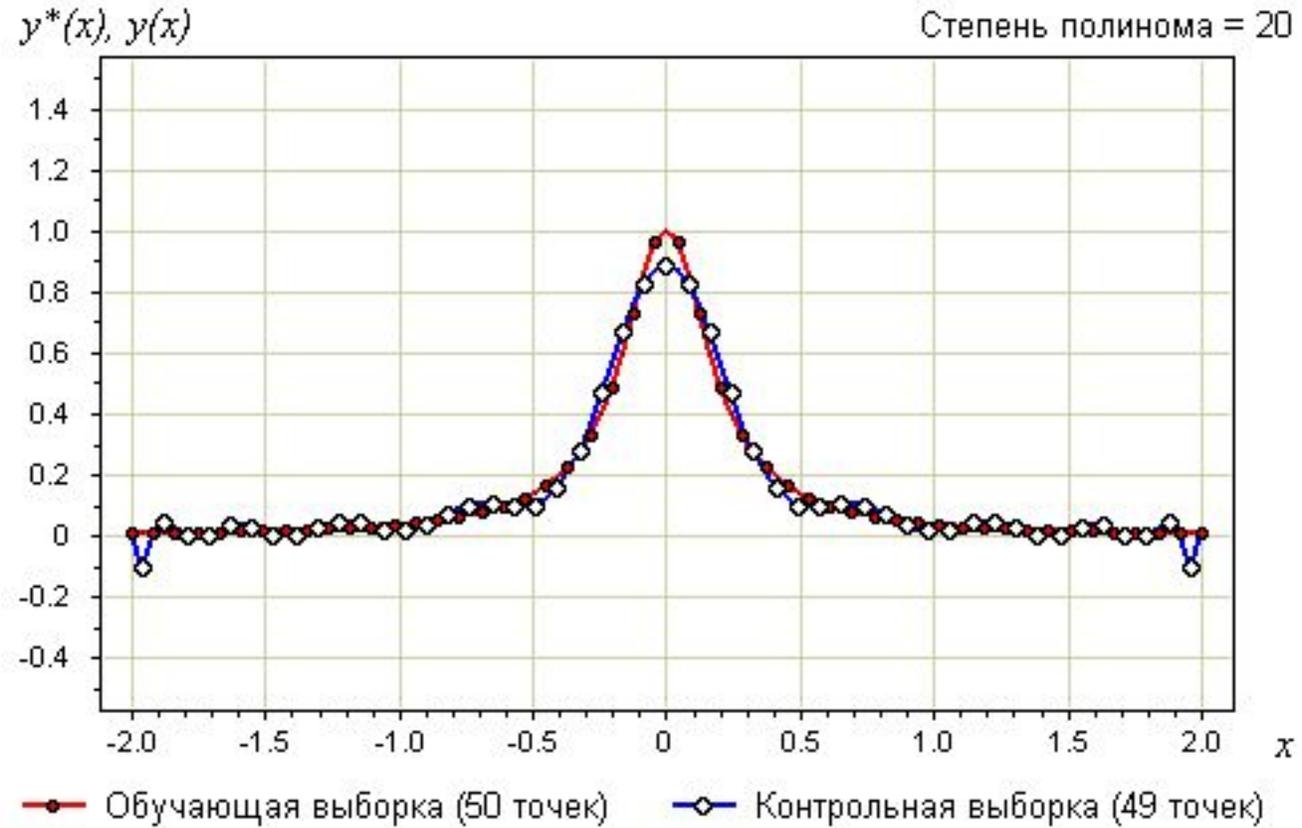
Выбор сложности  
модели

Evaluation

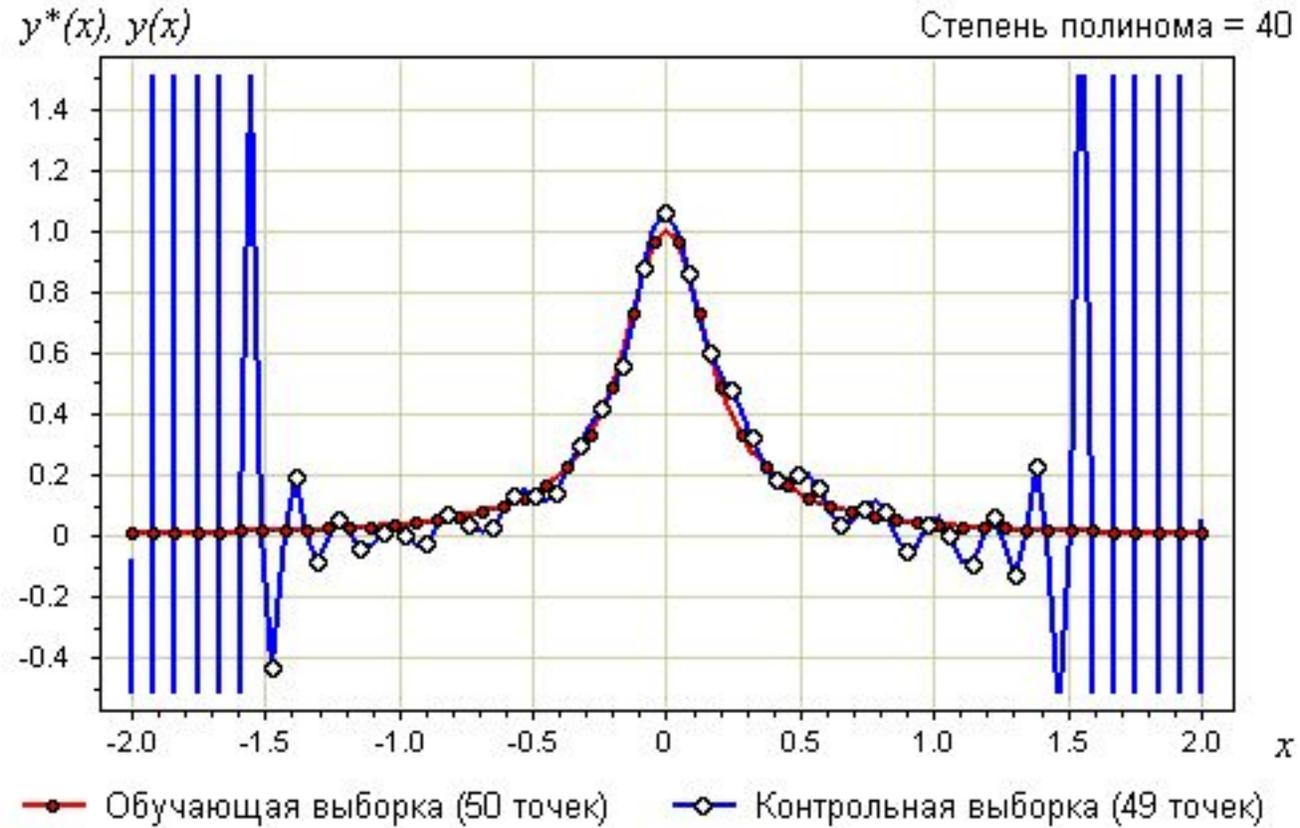
# На примере



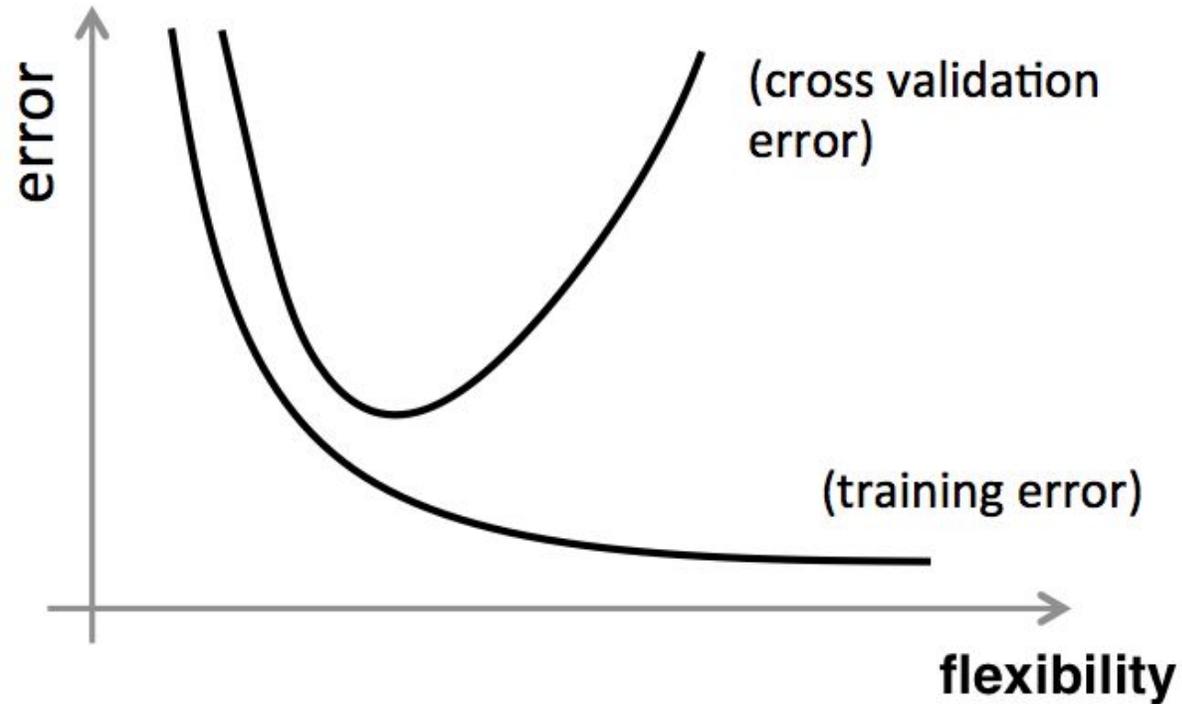
# На примере



# На примере



# Формальная картинка же



# Сложность (гибкость) модели

*Чем больше в модели параметров, тем больше информации она может нести.*

— Ваш К. О.

## Какая бывает информация в параметрах:

- про генеральную совокупность;
- про выборку;
- про random seed.

# Определение I

Переобучение, переподгонка (overfitting, high variance) — нежелательное явление, возникающее при решении задач обучения по прецедентам, когда вероятность ошибки обученного алгоритма на объектах тестовой выборки оказывается существенно выше, чем средняя ошибка на обучающей выборке.

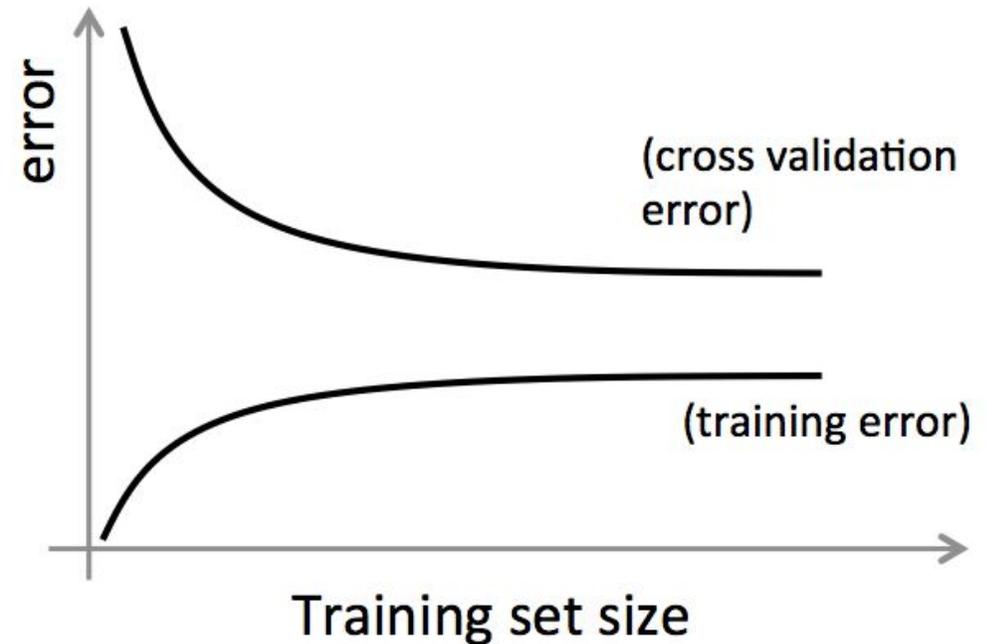
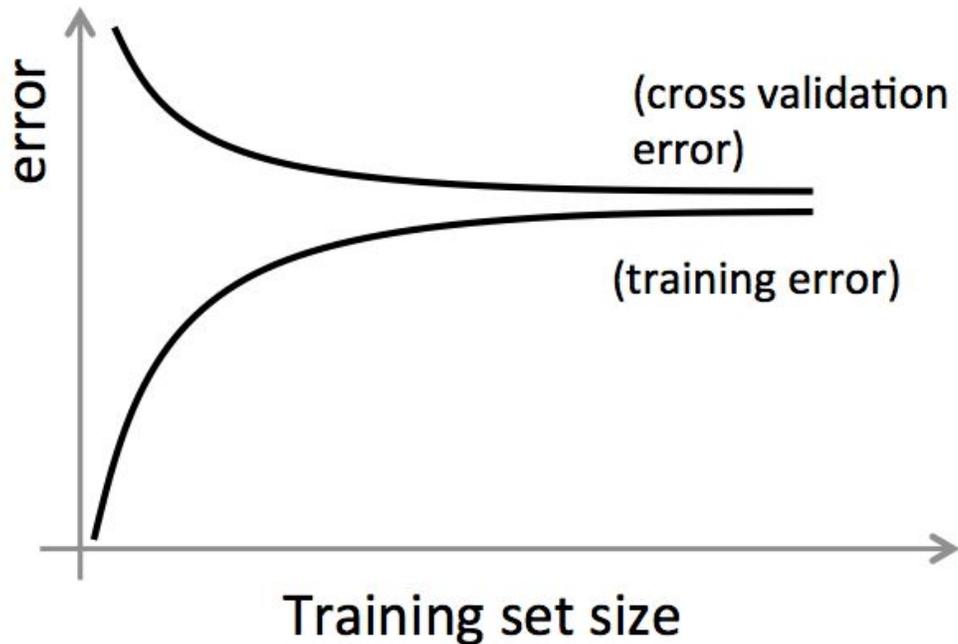
# Определение II

Недообучение (underfitting, high bias) — нежелательное явление, возникающее при решении задач обучения по прецедентам, когда алгоритм обучения не обеспечивает достаточно малой величины средней ошибки на обучающей выборке.

# Зачем знать

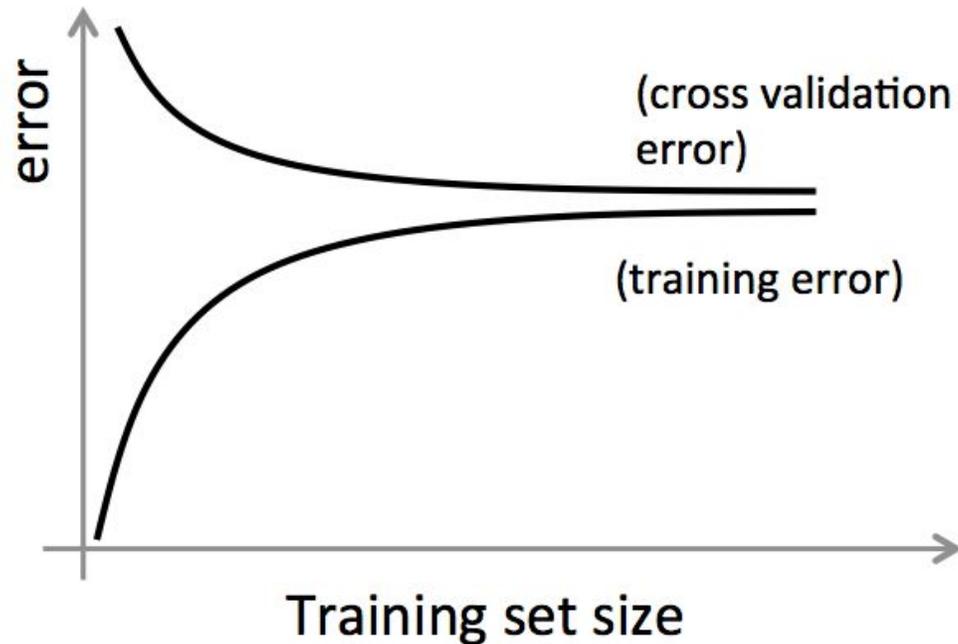
- Увеличение числа примеров для обучения исправляет high variance
- Меньшее число факторов исправляет high variance
- Уменьшение сложности модели исправляет high variance
- Увеличение числа факторов исправляет high bias
- Увеличение сложности модели исправляет high bias

# Как понять где мы находимся

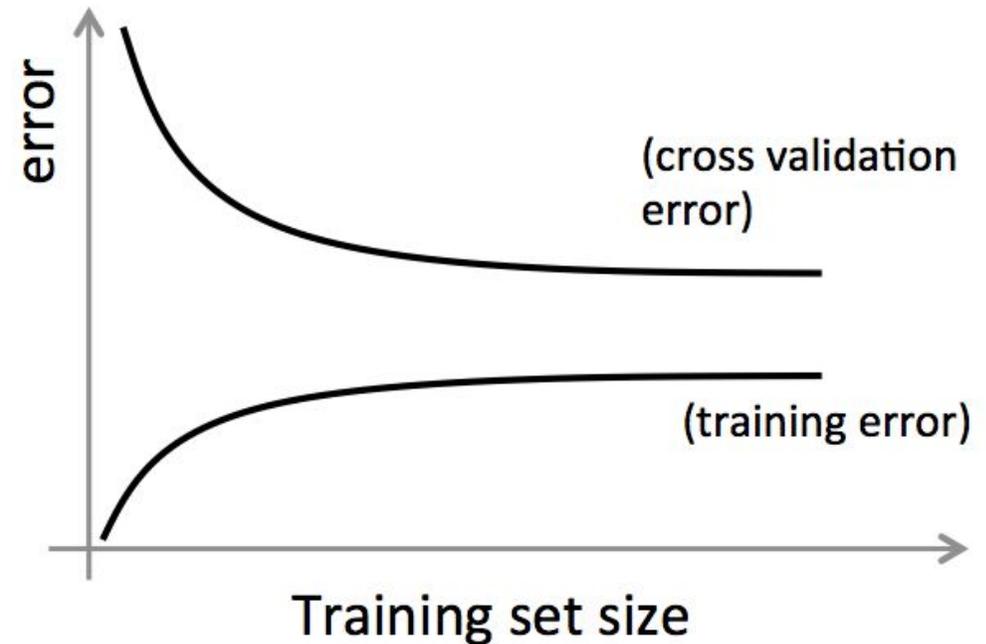


# Как понять где мы находимся

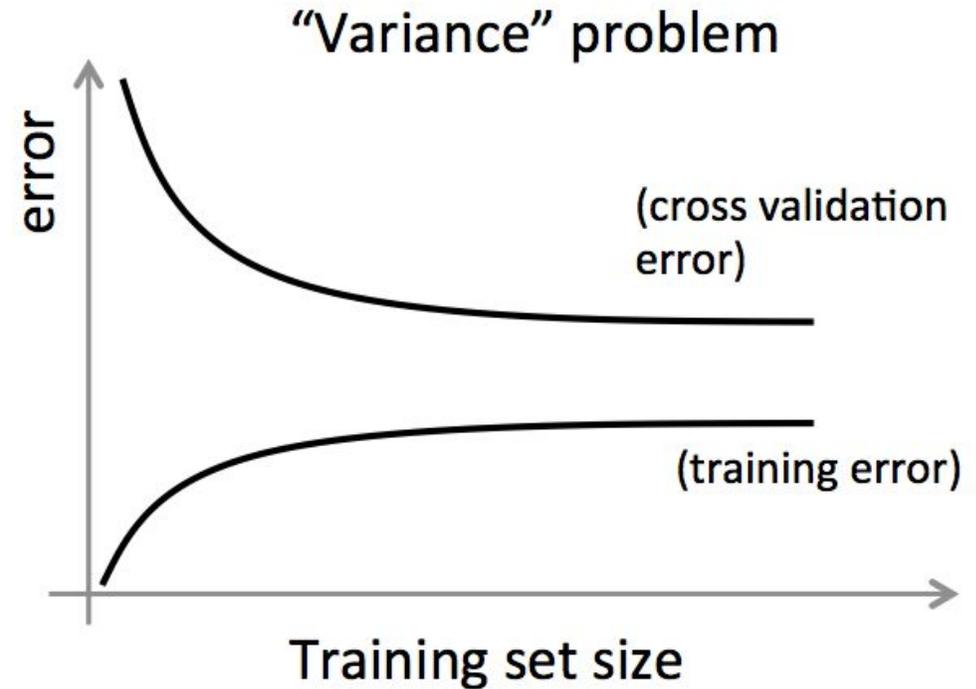
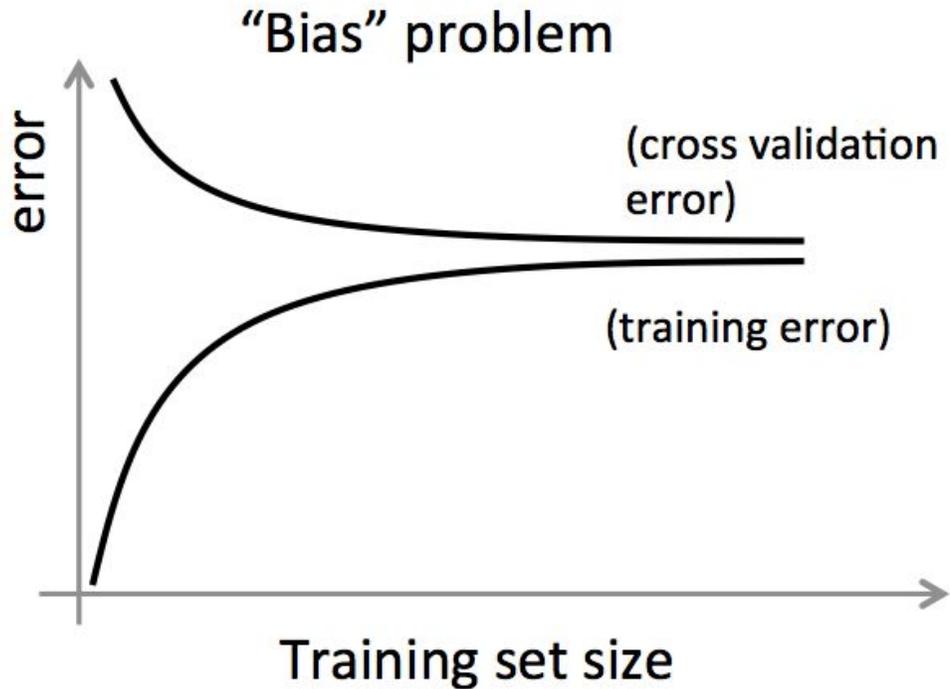
## Underfit



## Overfit



# Как понять где мы находимся



# Как можно переобучиться

- Линейные модели: степень полинома
- Деревья решений: глубина дерева
- Нейронные сети: ширина и глубина
- SVM: Kernel trick
- ...

# Итого offline

- В академической среде чаще всего это про сложность модели
- В промышленности это про выбор лучшей решающей функции
- Смотреть на это надо как на СПР
- У нас есть проверка гипотез

# Вопросы?