«Машинное обучение»

Сложность алгоритмов, переобучение, смещение и разброс

Александр Дьяконов

2 ноября 2021 года

План

Проблема обобщения

Переобучение / переподгонка / перенастройка

Недообучение / недоподгонка / недонастройка

Сложность алгоритмов

Смещение и разброс

Способы борьбы с переобучением

Регуляризация

В чём нас обманывают…

Проблема обобщения

напоминаем... $L(a, X_{\text{train}}) \vee L(a, X_{\text{test}})$

будет ли алгоритм также работать на новых данных?

Не путать с проблемой представительности выборки!

- неправильное разбиение на обучение и контроль
- данные меняются со временем – предсказываем будущее
- другое распределение теста (пример: ЭКГ)

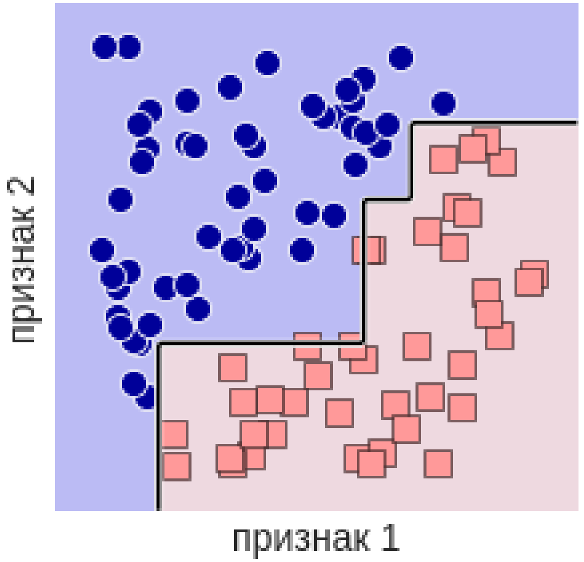
Считаем, что обучение и контроль одинаково распределены

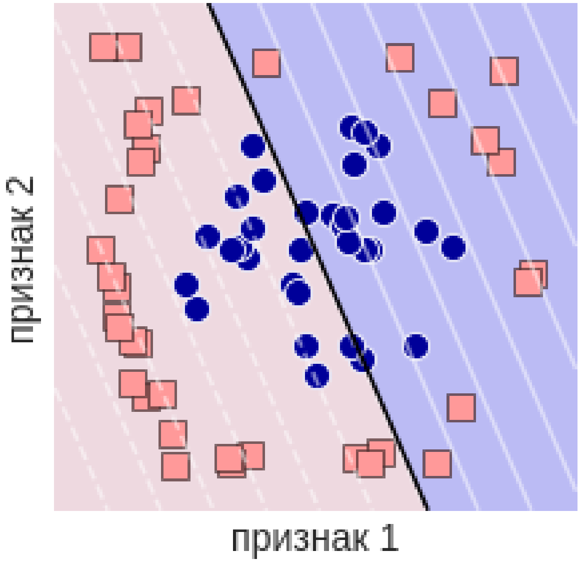
Термины: переобучение, недообучение, сложность

Переобучение / переподгонка / перенастройка (overfitting)
– явление, когда ошибка на тестовой выборке заметно больше ошибки на обучающей

Это главная проблема машинного обучения!
Если бы её не было $\Rightarrow$ минимизация эмпирического риска

Недообучение / недоподгонка / недонастройка (underfitting)
– явление, когда ошибка на тестовой выборке достаточно большая
(не удаётся «настроиться на выборку»)

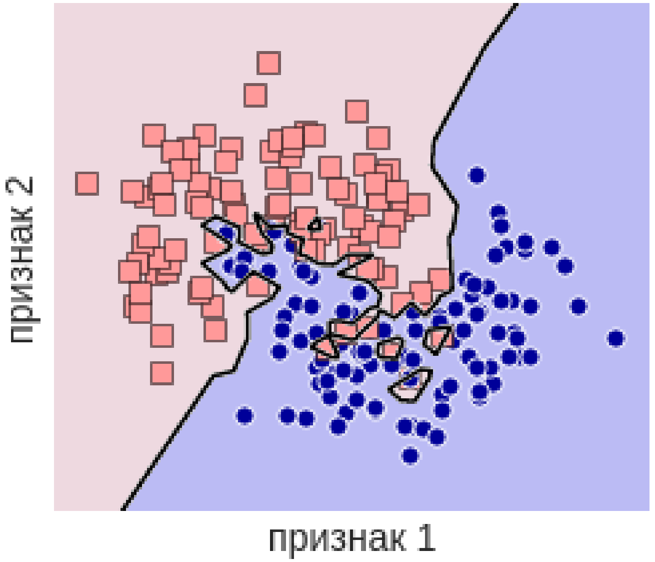


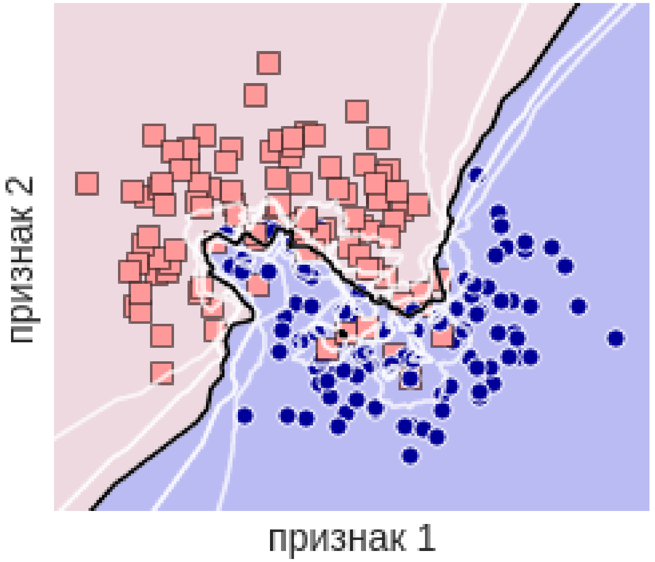


Термины: переобучение, недообучение, сложность

«Сложность» допускает много строгих формализаций…

Сложность (complexity / capacity) модели алгоритмов
– оценивает, насколько разнообразно семейство алгоритмов в модели с точки зрения их функциональных свойств
(например, способности настраиваться на выборки)



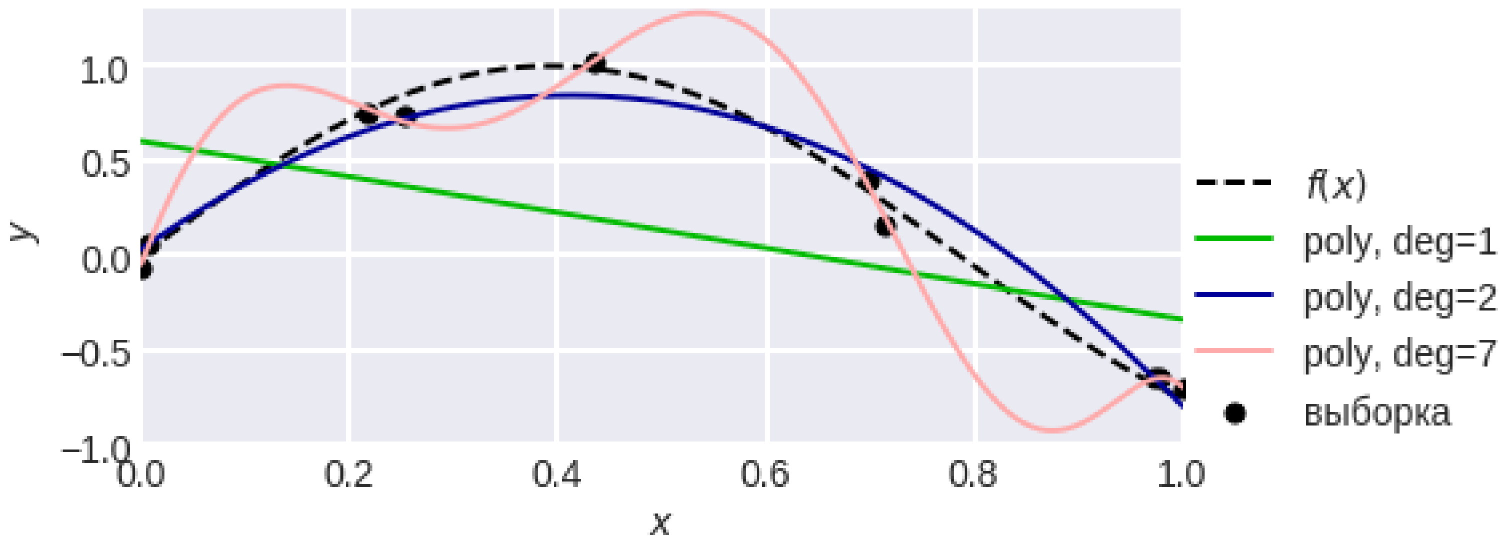


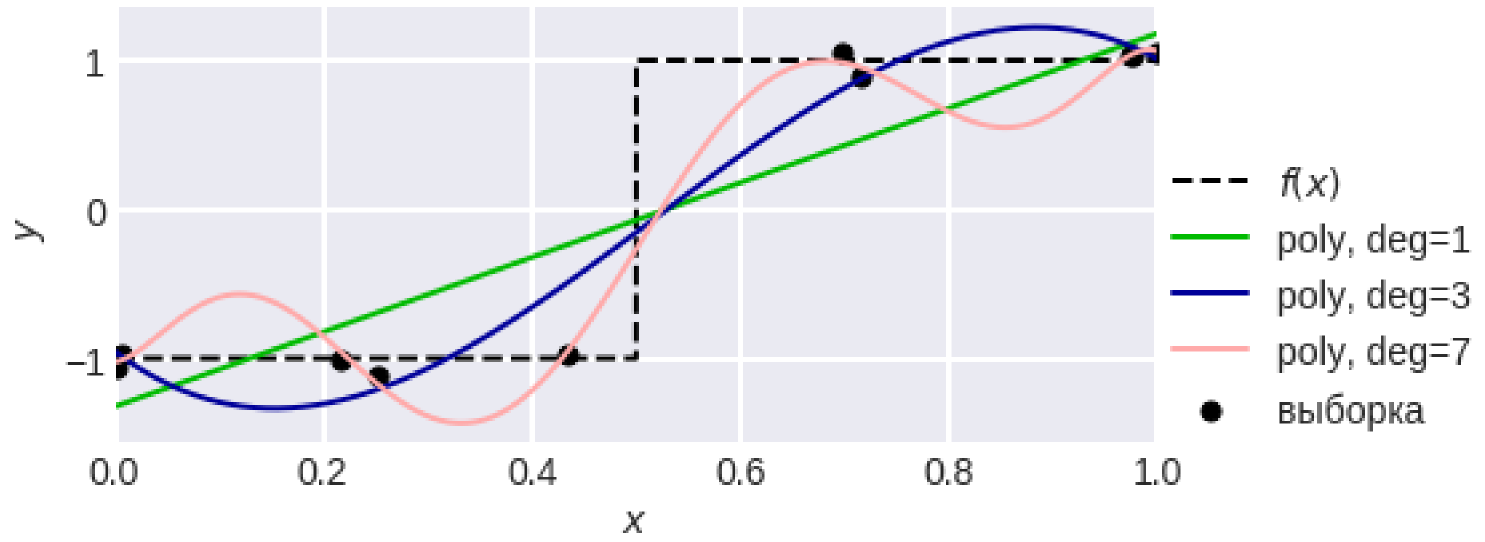
Повышение сложности решает проблему недообучения и вызывает переобучение

дальше это увидим

Проблема: постановка конкретной задачи

целевая зависимость известна с точностью до шума, ищем решение в классе полиномов

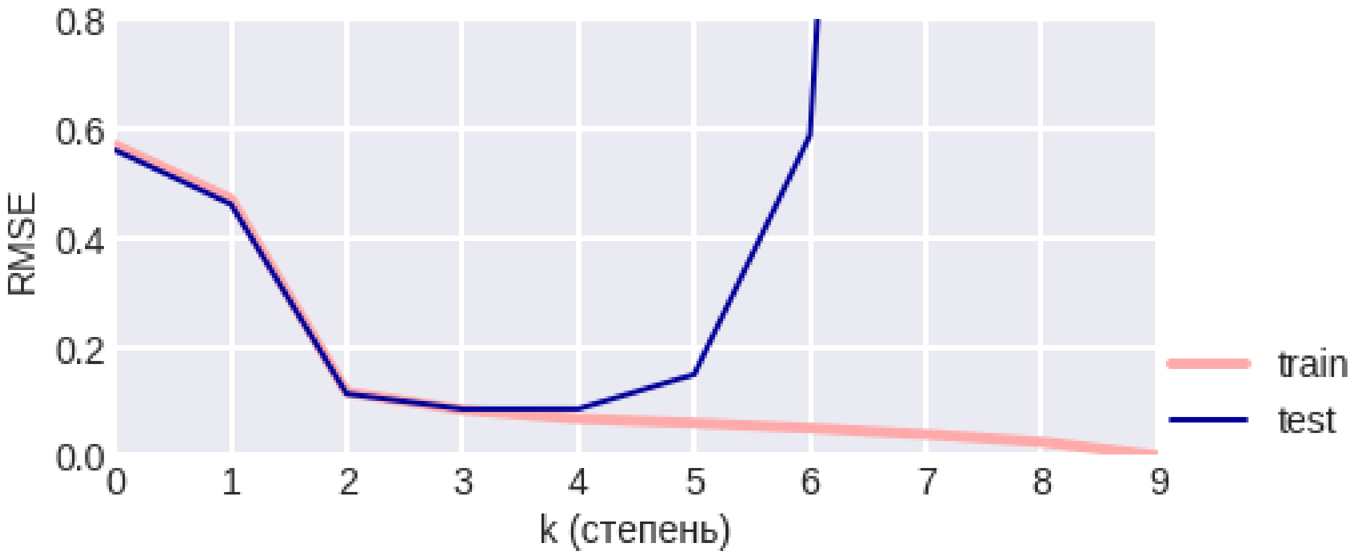




Проблема: сравнение разных по сложности алгоритмов

Полиномы малой степени – недостаточно хорошо описывают данные

Полиномы большой степени – проходят через точки обучения, но явно не похожи на «естественные функции»



При увеличении степени

- **ошибка на обучении падает**
- **ошибка на контроле сначала падает, потом растёт**

Задача регрессии (есть обобщения для классификации)

Пусть $y \equiv y(x) = f(x) + \varepsilon, \varepsilon \sim \mathrm{random}(0, \sigma^2)$

наш ответ в конкретной новой (нет в обучении) точке $a \equiv a(x)$, тогда

$$\mathbf{E}(y-a)^2 = \mathbf{E}(y^2 + a^2 - 2ya) =$$
$$= \mathbf{E}y^2 - (\mathbf{E}y)^2 + (\mathbf{E}y)^2 + \mathbf{E}a^2 - (\mathbf{E}a)^2 + (\mathbf{E}a)^2 - 2f\,\mathbf{E}a =$$
$$= \mathbf{D}y + \mathbf{D}a + (\mathbf{E}y)^2 + (\mathbf{E}a)^2 - 2\mathbf{E}ya =$$
$$= \mathbf{D}y + \mathbf{D}a + f^2 + (\mathbf{E}a)^2 - 2f\,\mathbf{E}a =$$
$$= \mathbf{D}y + \mathbf{D}a + (\mathbf{E}(f-a))^2 =$$
$$= \sigma^2 + \mathrm{variance}(a) + \mathrm{bias}^2(f, a)$$

- **Разброс (Variance)** – $\mathbf{D}a$
- **Смещение (Bias)** – $\mathbf{E}(f-a)$
- **Шум** – σ^2

Тонкий момент про независимость:

$$\mathbf{E}(ya) = \mathbf{E}((f(x)+\varepsilon)\cdot a(x)) =$$

const

$$= f(x)\mathbf{E}(a(x))$$

с.в.

ε – шум в новой точке x

a – обучен на таком же, но независимом шуме

Задача регрессии

Важно: по чему берётся матожидание

$$\mathbf{E}(y-a)^2 \equiv \mathbf{E}_{(x_i, f(x_i)+\varepsilon_i)_{i=1}^m}(y-a)^2$$

по данным (обучающей выборке)!

Выборки (случайные!) выбираются согласно некоторому распределению

$\Rightarrow$ алгоритм a, полученный с помощью обучения на выборке, случаен

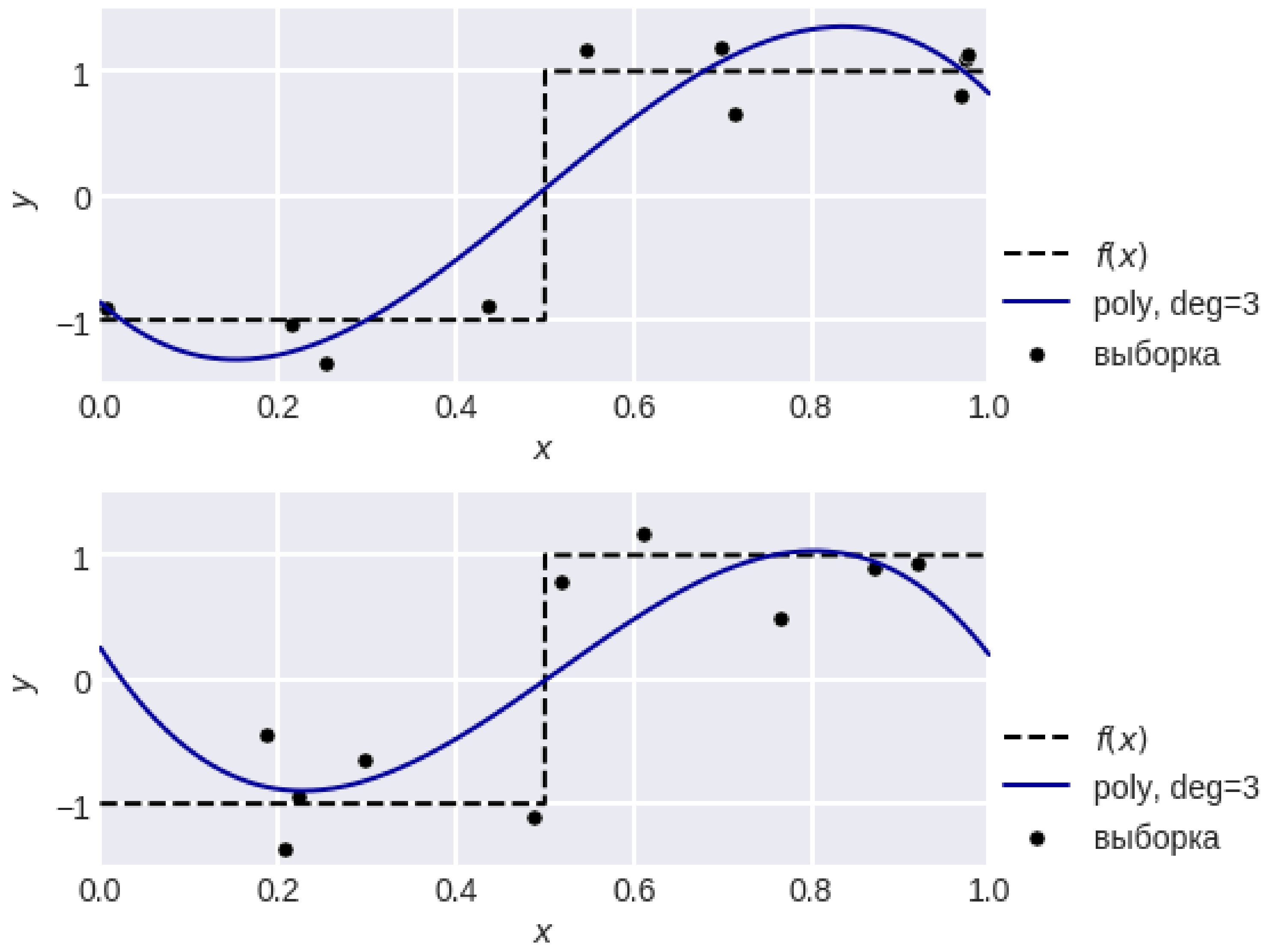
Формулу мы получили на конкретном объекте

$$\mathbf{E}(y-a)^2 \equiv \mathbf{E}(y(x)-a(x))^2$$

При желании можно проинтегрировать по всем объектам!

$$\mathbf{E}_D\mathbf{E}_X(y(x)-a_D(x))^2 = \mathbf{E}_X\mathbf{E}_D(y(x)-a_D(x))^2$$

Случайные выборки



для разных выборок будут разные решения – в рамках одной модели

Разброс и смещение

Разброс (Variance) $\mathrm{D}a$

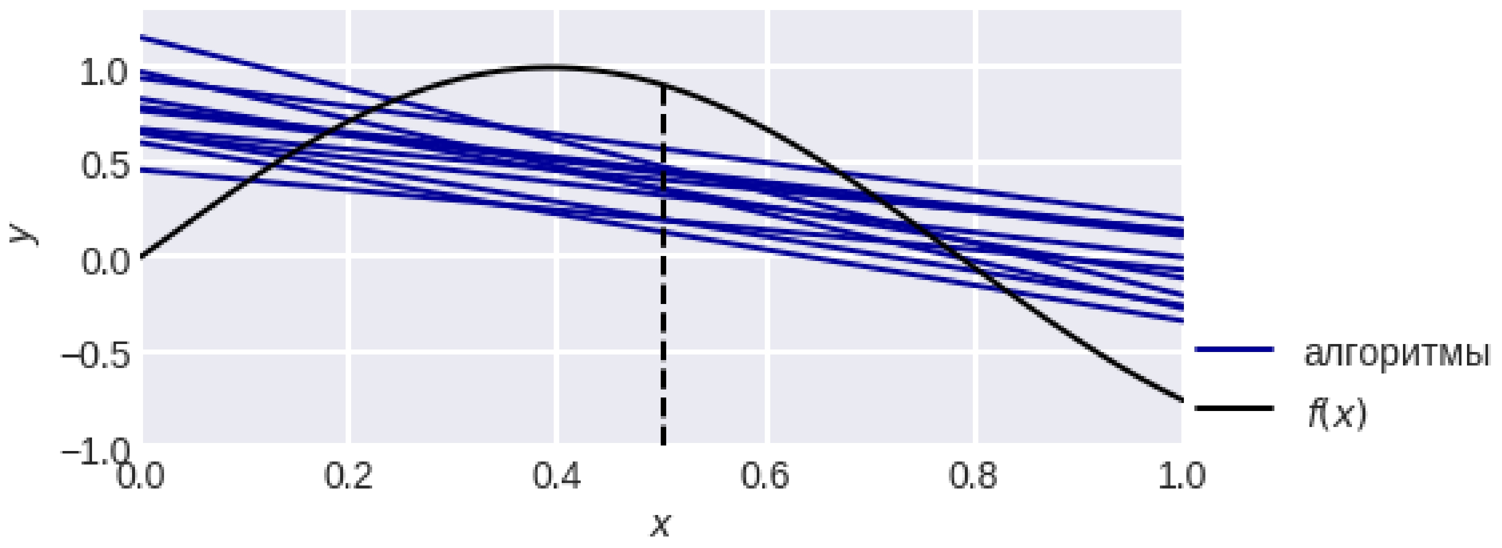
– разнообразие алгоритмов

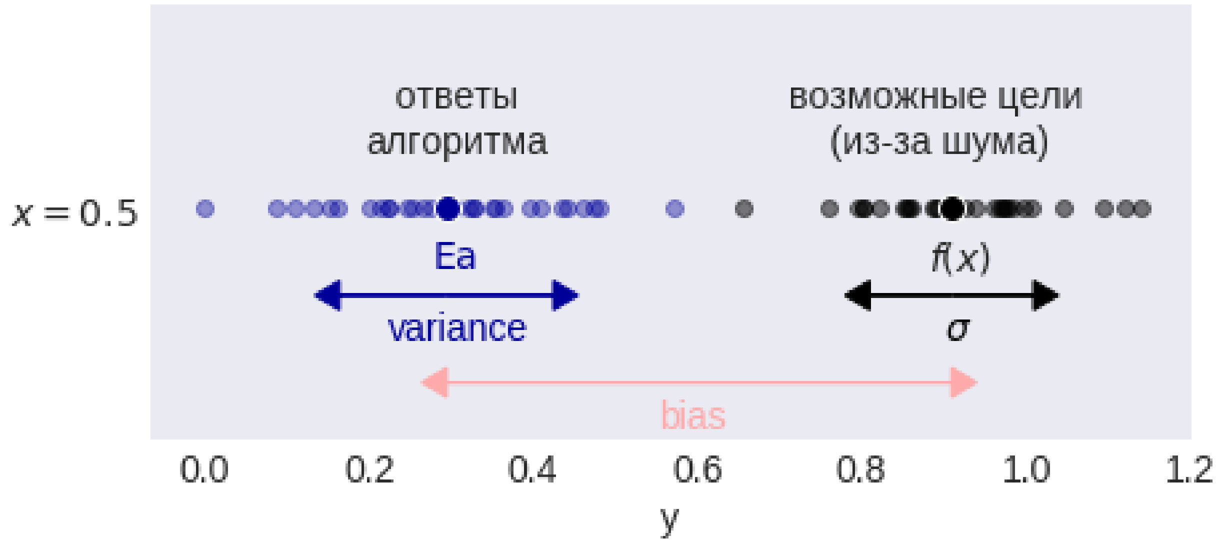
(из-за стохастической природы настройки и/или случайности обучающей выборки, в том числе, шума)

Смещение (Bias) $\mathrm{E}(f-a)$

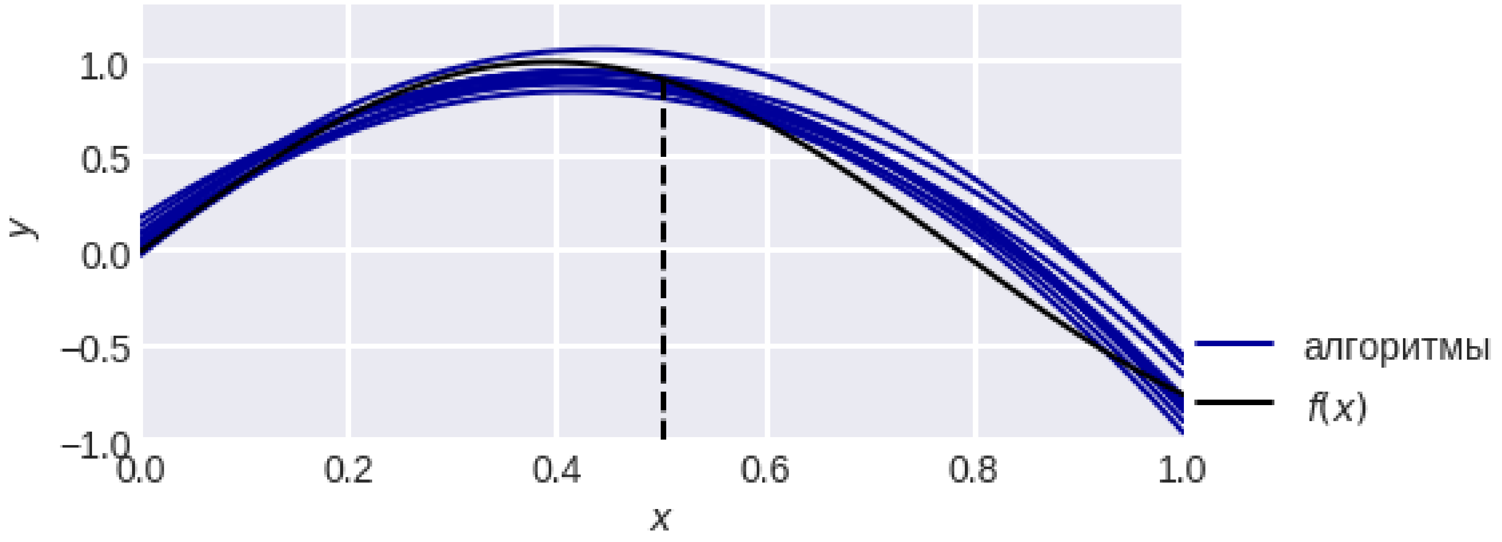
– способность модели алгоритмов настраиваться на целевую зависимость

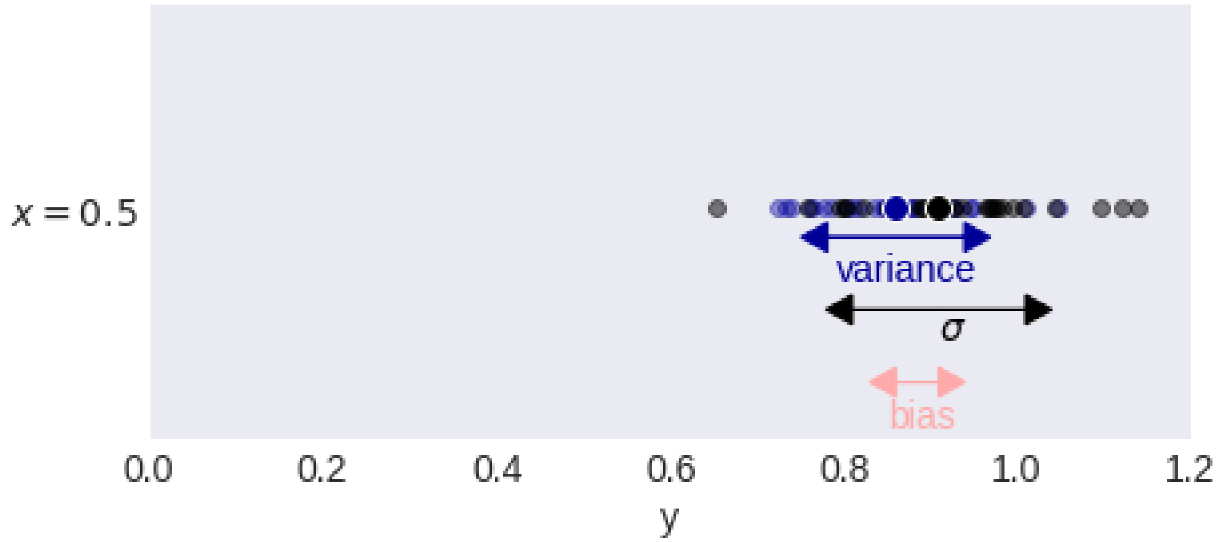
Эксперимент: генерируем разные обучающие выборки…



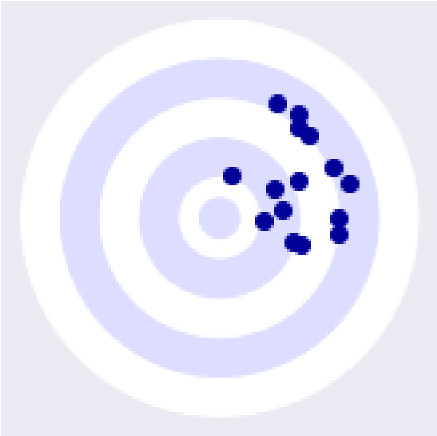


Эксперимент: генерируем разные обучающие выборки…

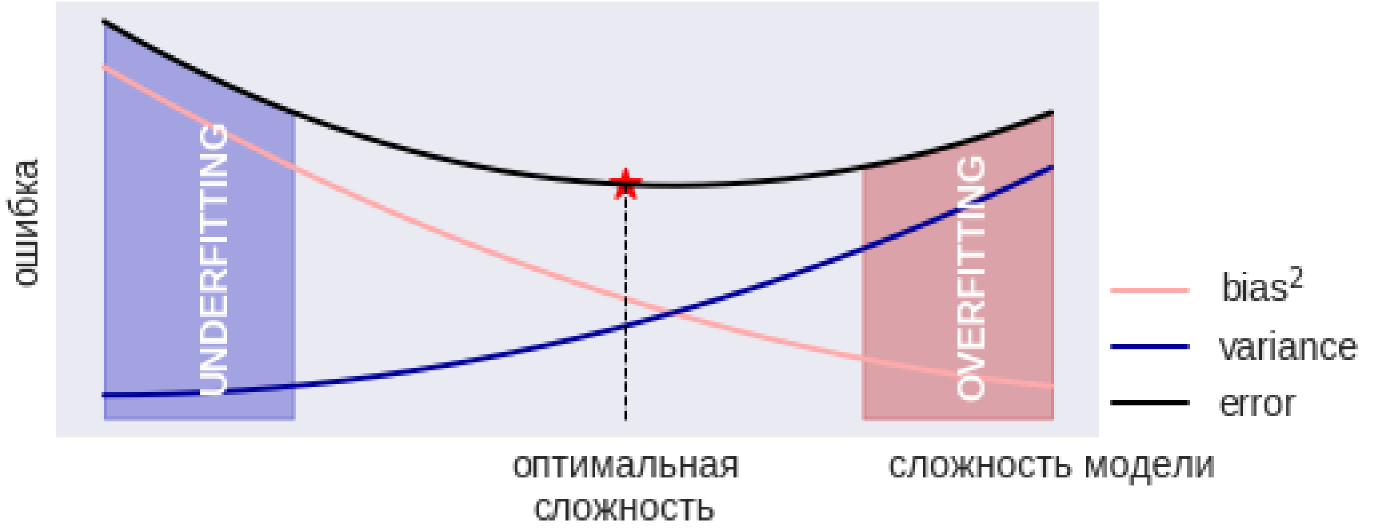




Разброс и смещение

	Малое смещение **Хорошо: настраиваемся на целевую зависимость**	**Большое смещение** **Плохо: модель не соответствует данным**
Малый разброс **Хорошо:** **Модель устойчива** **(не зависит от шума в данных)**		
Большой разброс **Плохо: слишком сложная модель (много алгоритмов в ней), настраиваемся на шум**		

Частая картинка



Примеры

MLE обычно несмещённая оценка, но большой разброс

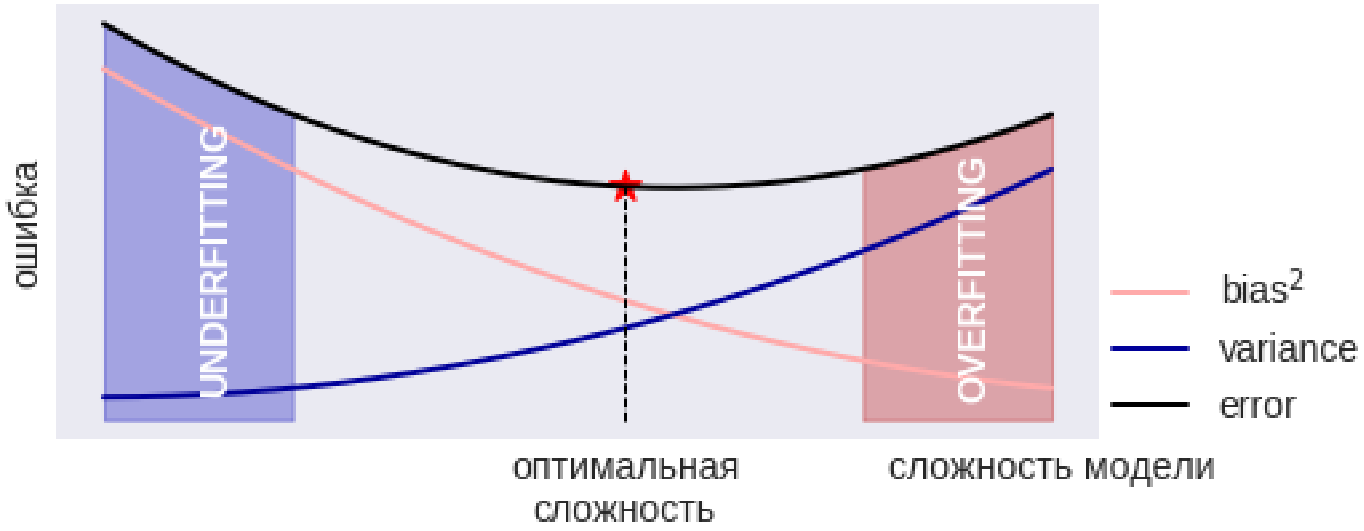
MAP – обычно смещённая, но малый разброс

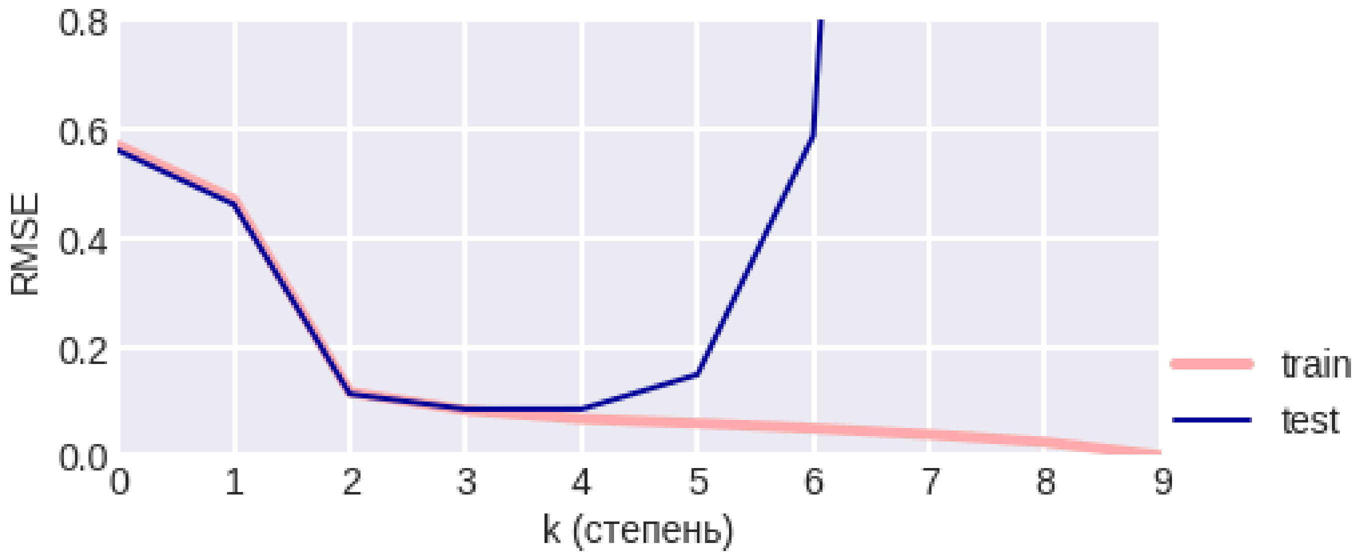
«Бедная» модель – не может настроиться на целевую зависимость

«Сложная» модель – может, но не настраивается
(т.к. подвержена переобучению, настраивается на шум)

Есть такие определения: overfitting = «too much variance», complexity = (1 / variance)
Часто: «ёмкость» (capacity), «способность к обобщению» (representation power)…

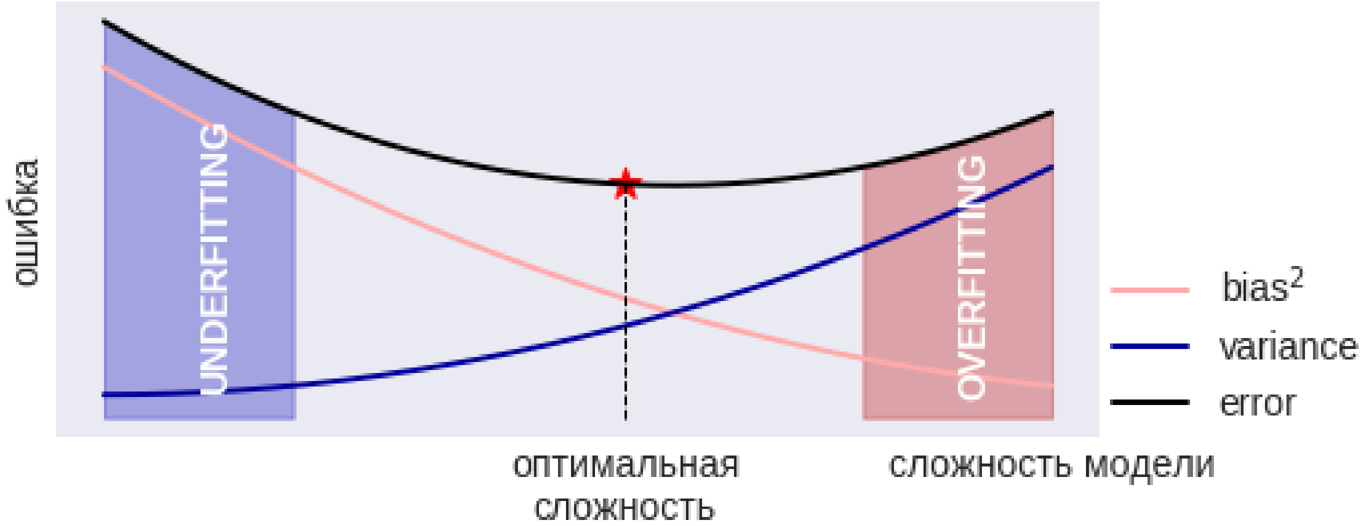
Частая картинка (шума на графиках нет по понятным причинам…)

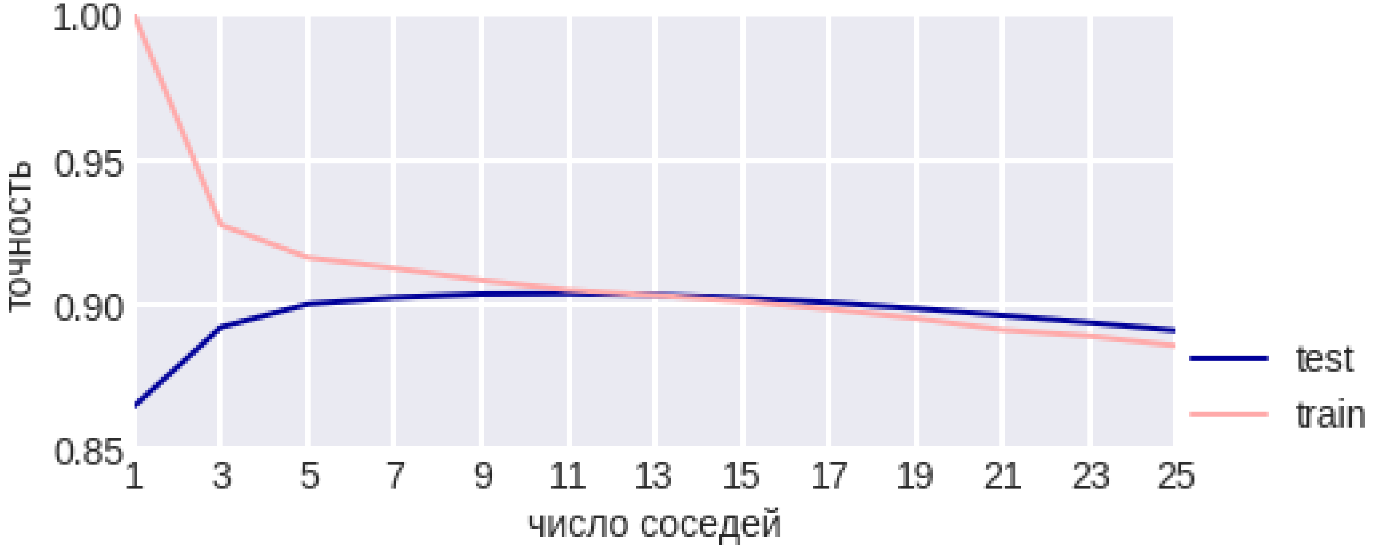




видим, что на тесте такой же график ошибки, что и на «частой картинке»

Частая картинка





тут точность (не ошибка) и сложность ~ 1/(число соседей)

Теория: bias-variance

Для k ближайших соседей есть формула:

$$\mathbf{E}(y-a)^2 = \left(f(x) - \frac{1}{k}\sum_{t=1}^{k} y(x_t) \right)^2 + \frac{\sigma^2}{k} + \sigma^2$$

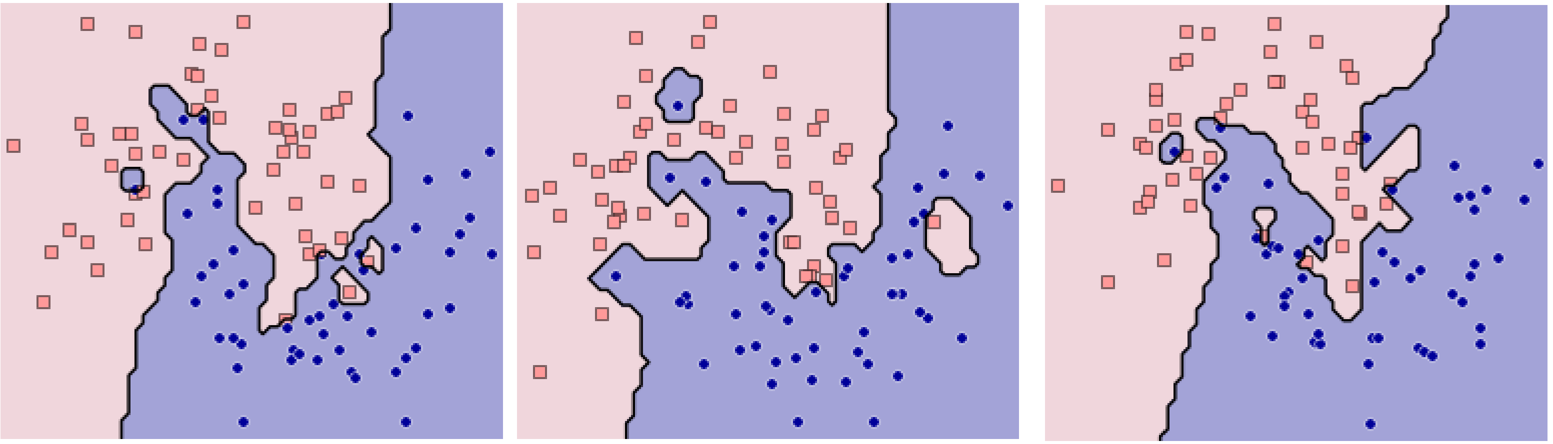
Hastie T., Tibshirani R., Friedman J. «The Elements of Statistical Learning» – 2009.

Мы вывели для задач регрессии с MSE

Есть вывод и для задач классификации

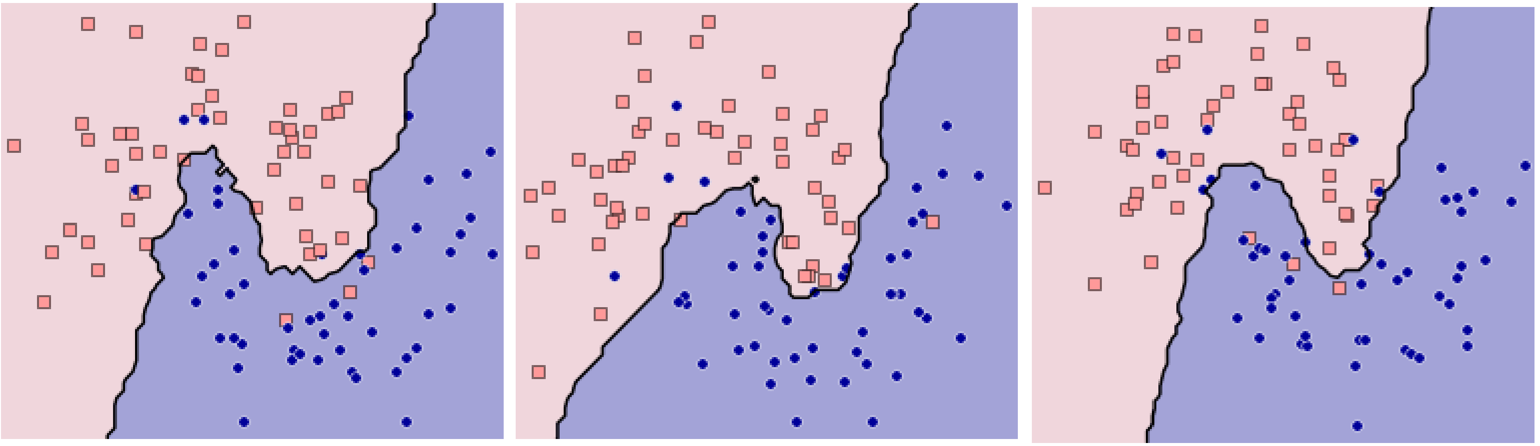
Domingos P. «A unified bias-variance decomposition» // ICML – 2000.

Почему 1NN сложнее 9NN



Разделяющие поверхности 1NN для разных выборок

(одинаково распределённых)



Разделяющие поверхности 9NN для тех же выборок

Результат стабилен!

Почему 1NN сложнее 9NN

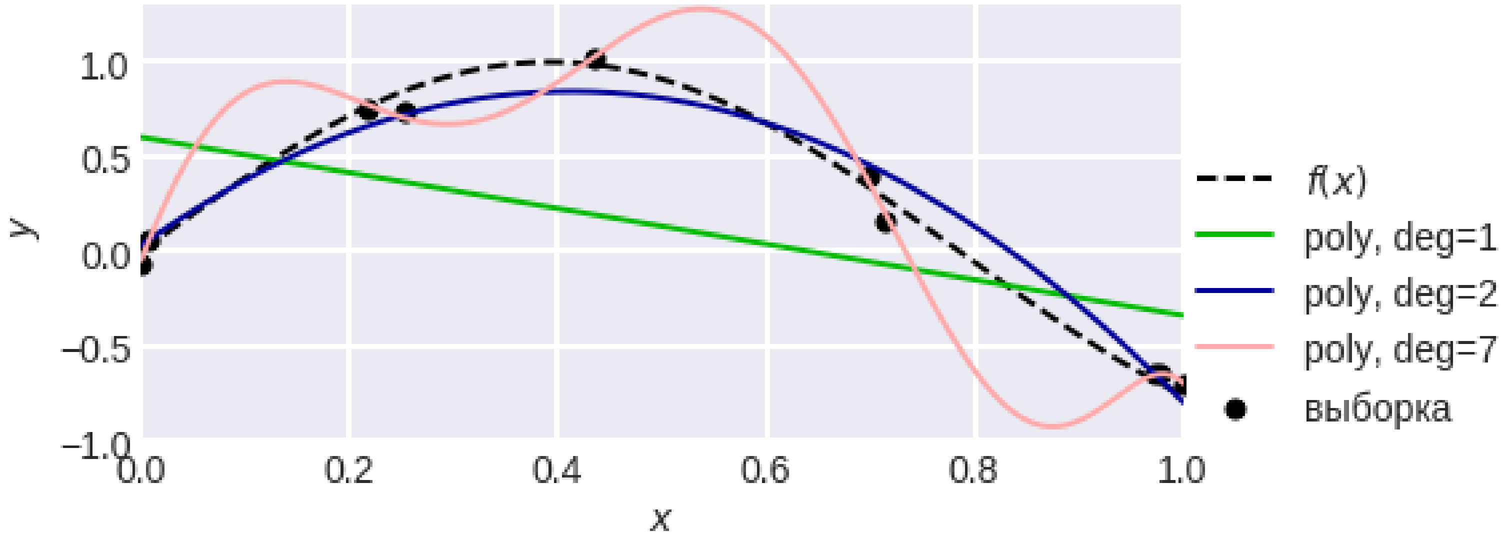
Эти алгоритмы имеют

- **одинаковые параметры** (что бы не понималось под этим…)
- **требуют хранения всей обучающей выборки (lazy algorithms)**
- **9NN даже «чуть сложнее в реализации»**

но разброс у 9NN меньше…

смещения не отличаются???

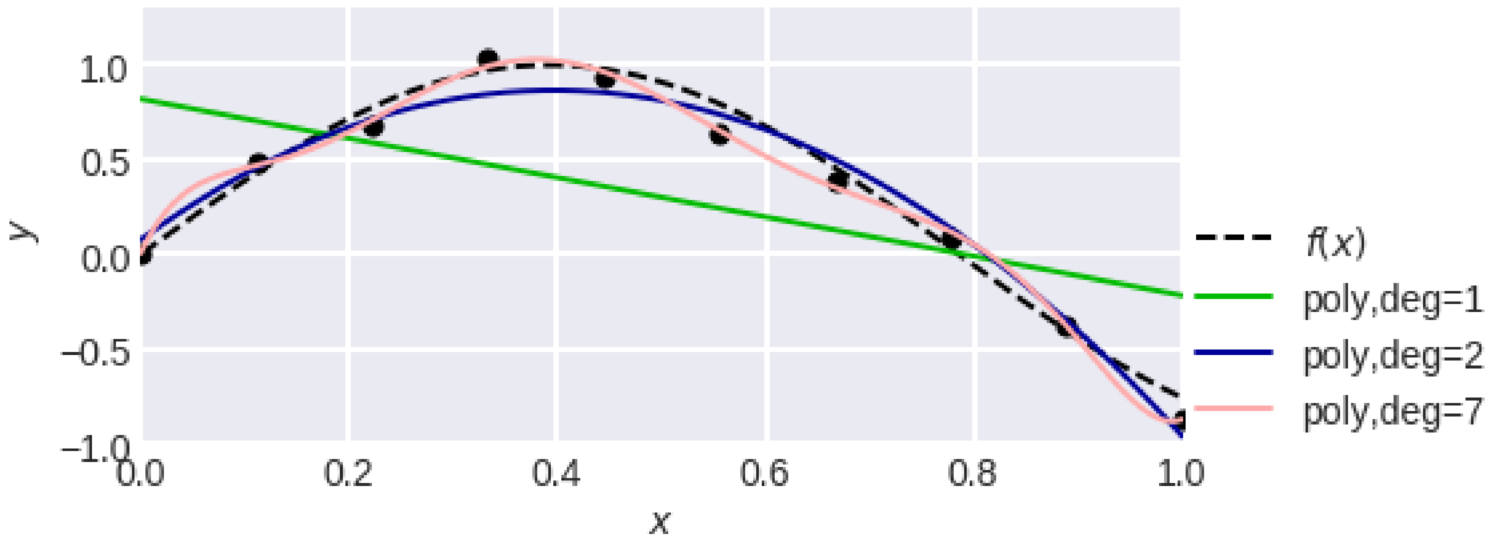
Способы борьбы с переобучением



Будем иллюстрировать на такой модельной задаче…

Способы борьбы с переобучением

1. Выборка специальной структуры

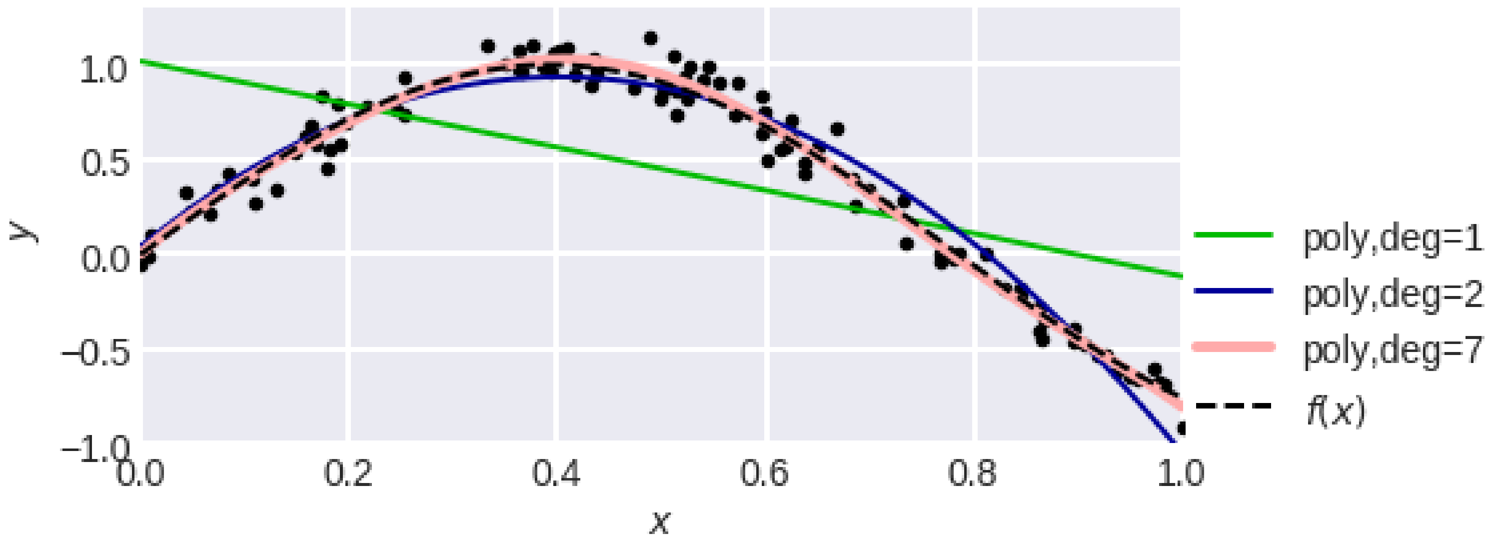


Даже при наличии шума, если есть возможность «формировать выборку», это можно сделать так, чтобы уменьшить переобучение

Выбор специальных данных (ex: которые обманывают алгоритм)

Способы борьбы с переобучением

2. Увеличение объёма данных

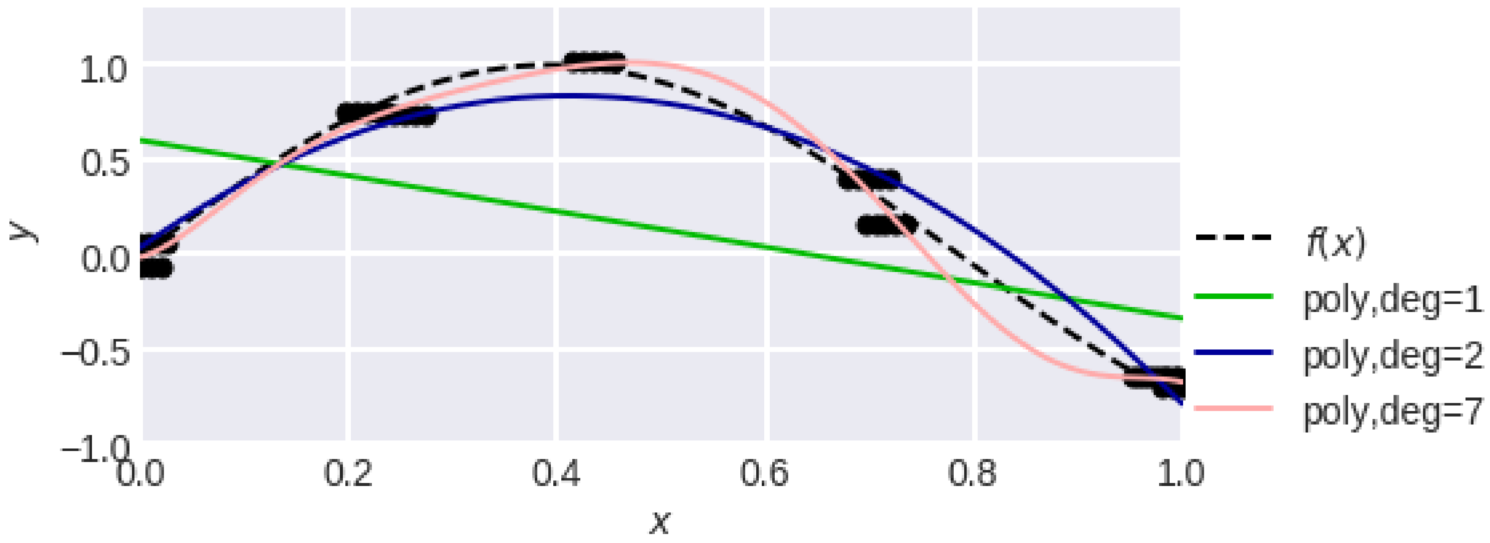


Данные первичны, алгоритмы вторичны!

Но чтобы сложные алгоритмы не переобучались нужны действительно большие объёмы.

Способы борьбы с переобучением

1+2=3. «Аугментация»



Искусственное увеличение выборки так, чтобы алгоритм удовлетворял требуемым свойствам

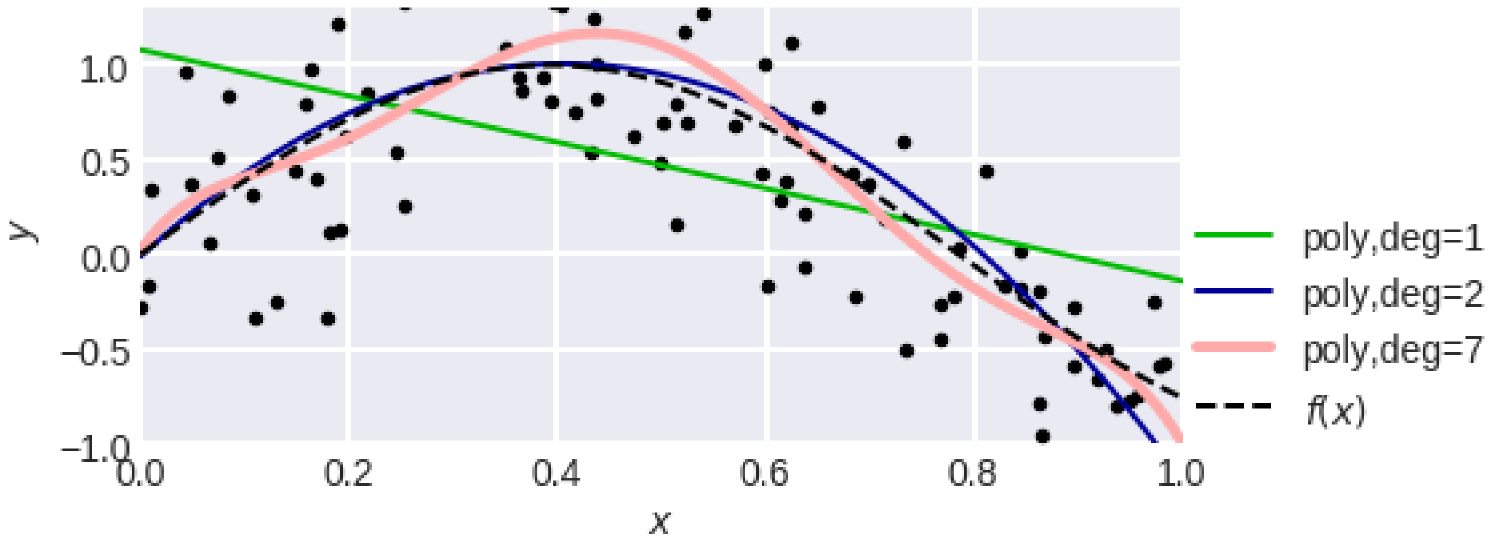
Частый приём: внесение шума в данные

В нейросетях м.б. добавления шума в промежуточные слои!

Иногда: в целевой признак.

Способы борьбы с переобучением

4. Улучшение качества данных



шум / выбросы / аномальные дубликаты и пропуски

Хотя это всё-таки, как правило, не способ борьбы с переобучением.
Это больше влияет на ошибку ε в $y(x) = f(x) + \varepsilon$!

Способы борьбы с переобучением

5. Использование других данных / задач / готовых моделей

Как правило в DL, где модели сложные…

1) нейросеть можно обучить на аналогичной задаче

ex.: другая задача классификации

ex.: такая же задача, но данные на другом оборудовании

ex.: синтетические данные (в сегментации)

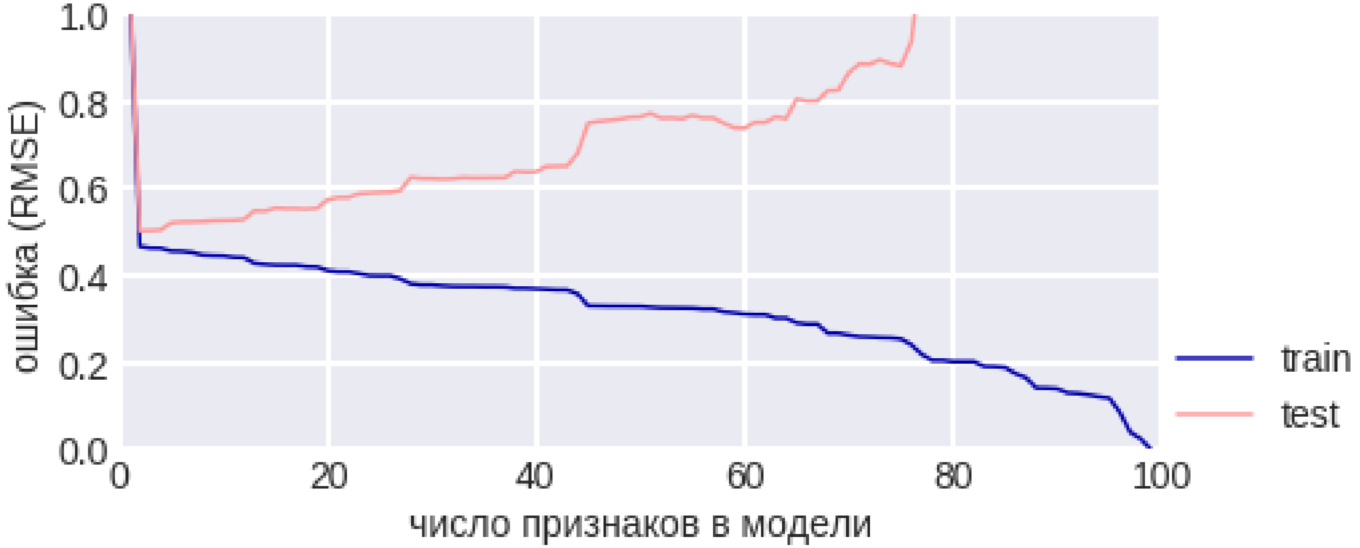
2) можно взять уже обученную (на другой задаче) нейросеть и дообучить её

6. Сокращение размерности, отбор признаков

(тоже формально про данные)

Почему много признаков – плохо

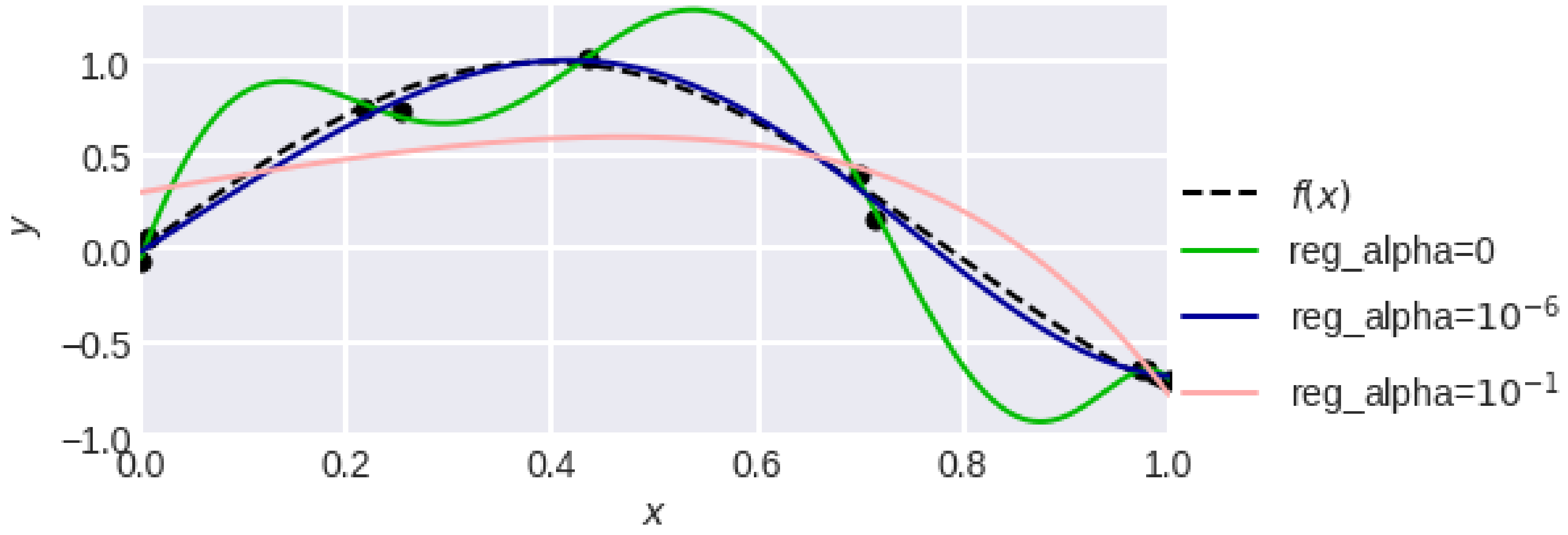
$$m = n = 100,\ y = X_1 - X_2 + \mathrm{norm}(0, 0.5),\ X_i = \mathrm{norm}(0, 1)$$



Способы борьбы с переобучением

7. Регуляризация

До этого говорили про данные, теперь про алгоритмы…



Уменьшение сложности модели!

Изменение настройки модели

здесь: добавление штрафующего слагаемого в опт. функционал

Регуляризация

- **Добавление штрафующего слагаемого к минимизируемому функционалу**

$$(y(x)-f(x\,|\,w))^2+\lambda\,||\,w\,||^p\to\min$$

обоснование в MAP

- **Разреженные представления**

(зануления весов, выходов нейронов)

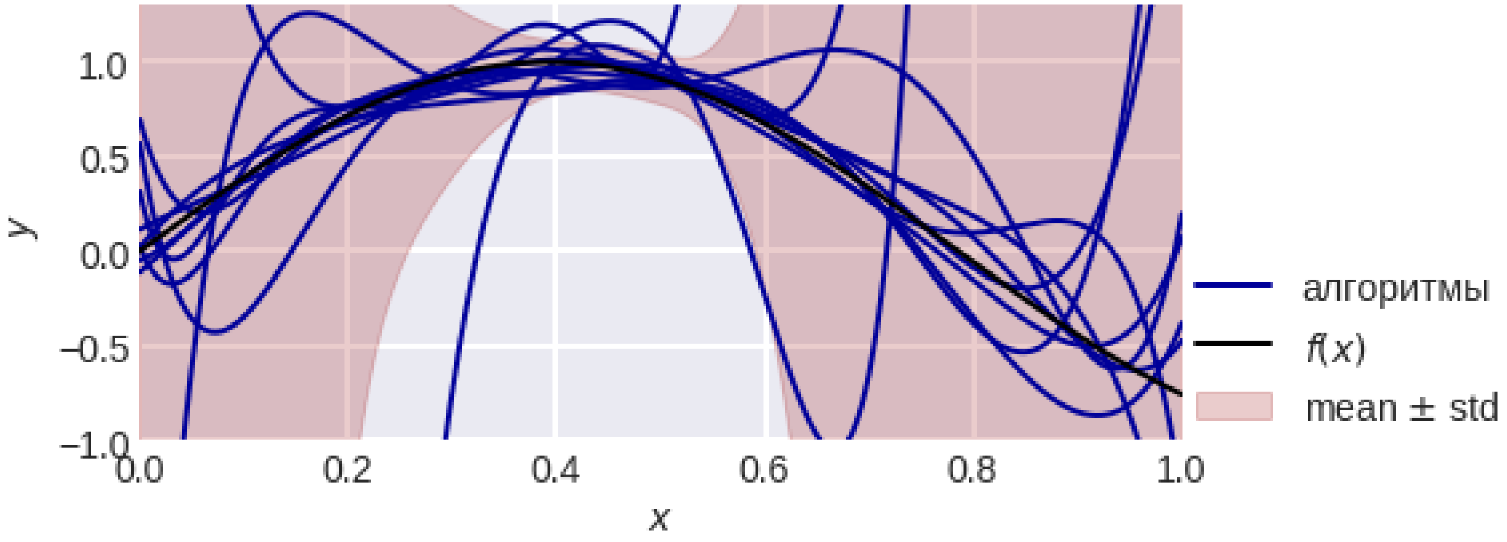
- **Прореживание (Drop Out)**

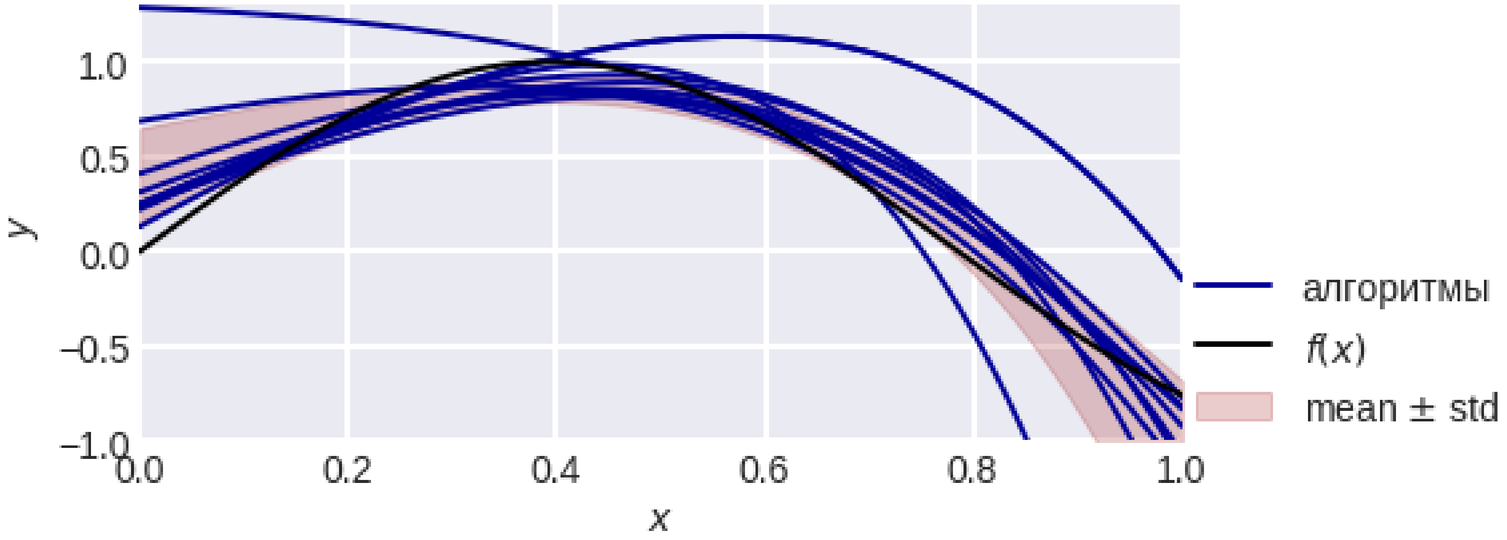
- **Подрезка деревьев (Pruning)**

- **Разделение параметров (Parameter Sharing)**

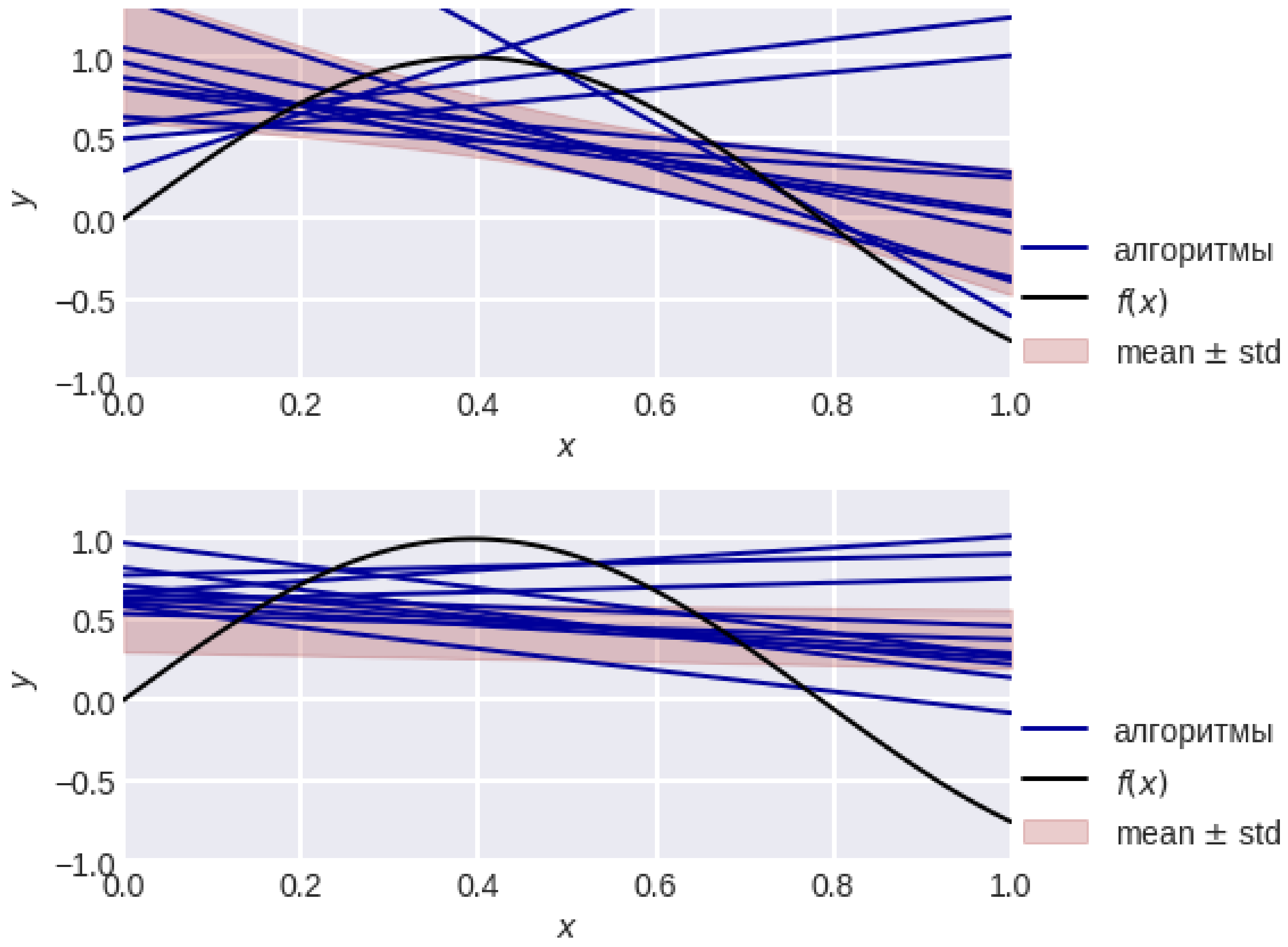
Тренируем НС требуя, чтобы значения её параметров были также близки к параметрам другой НС, обученной без учителя

Пример регуляризации: до и после

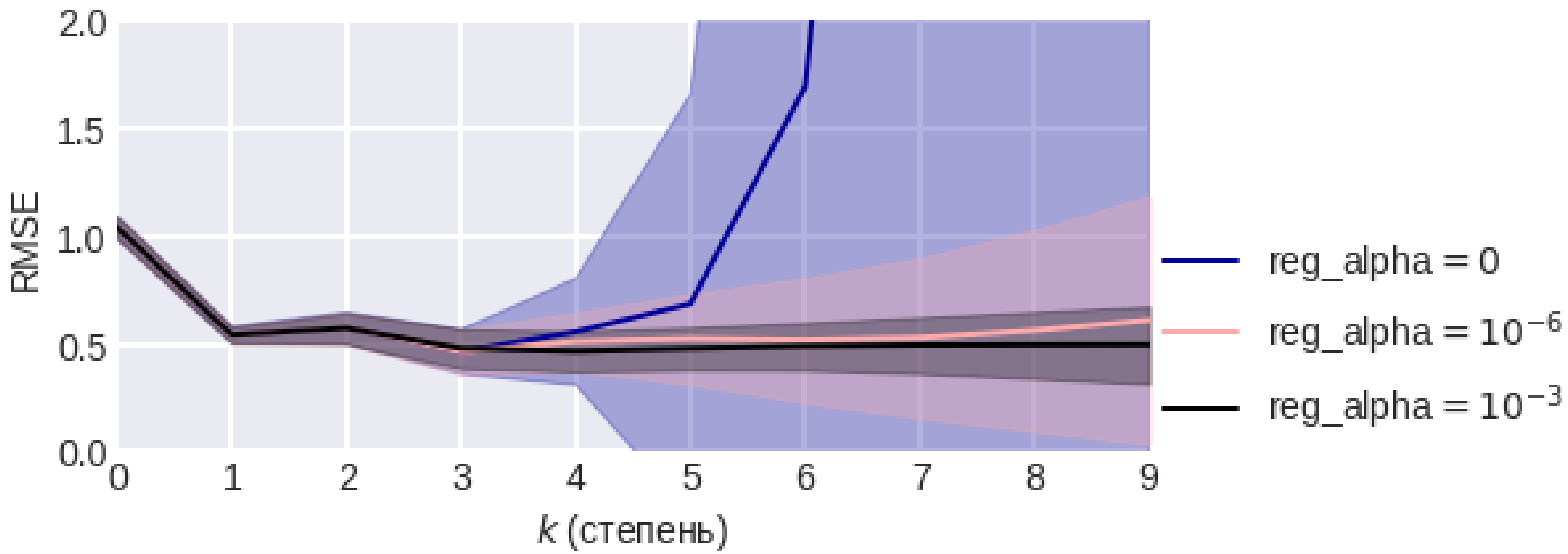




Пример регуляризации: до и после



Пример регуляризации: до и после



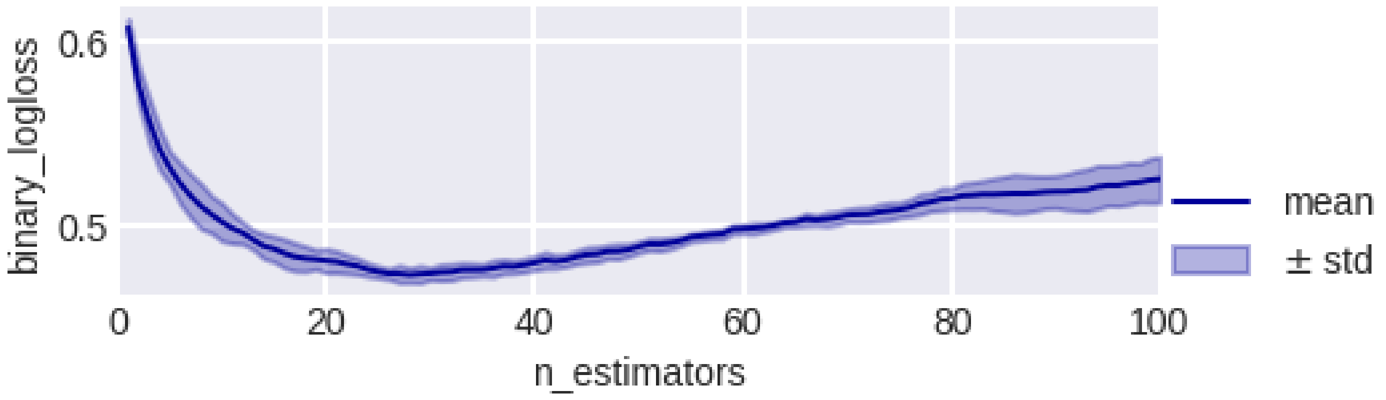
уже видели, что параметры линейной регрессии $\to$ 0

Способы борьбы с переобучением

8. Организация контроля
– самое важное!
hold out, CV, и т.п.

- **ранняя остановка (early stopping)**

обучение НС, бустинг – где есть итерации
используем отложенный контроль



[DLbook] В модельной ситуации ES эквивалентна L2-регуляризации

Способы борьбы с переобучением

9. Выбор архитектуры алгоритма

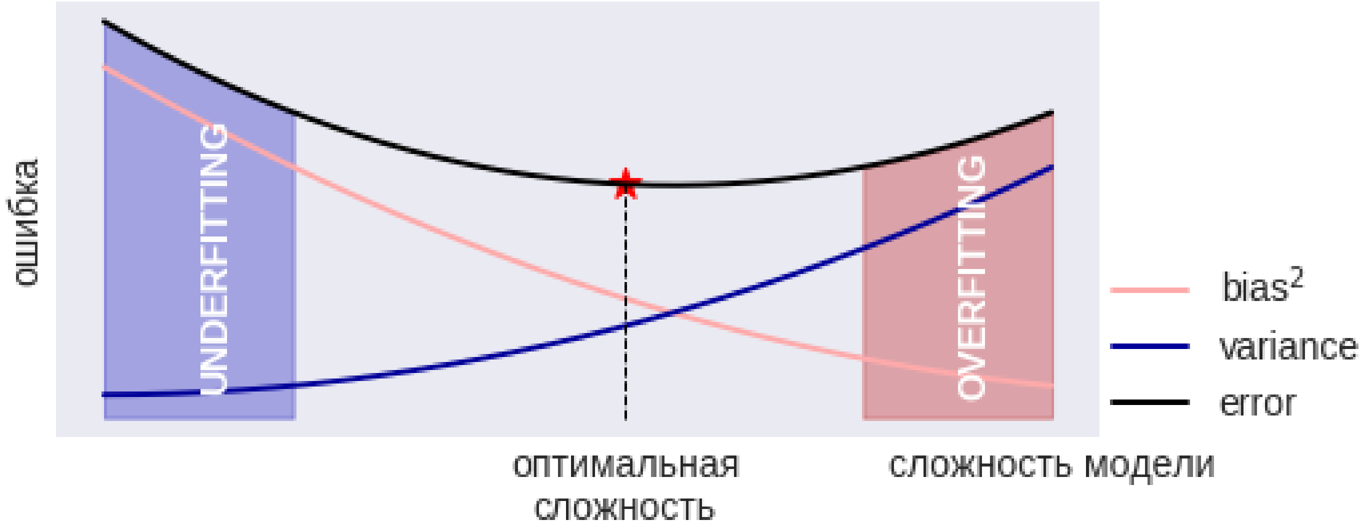
Пример: свёртки + пулинг, где есть инвариантность
(+ сокращает число параметров)

Пример: усреднение, бэгинг

в отличие от уменьшения сложности (см. раньше) тут сразу выбираем простое/специально устроенное и т.п.

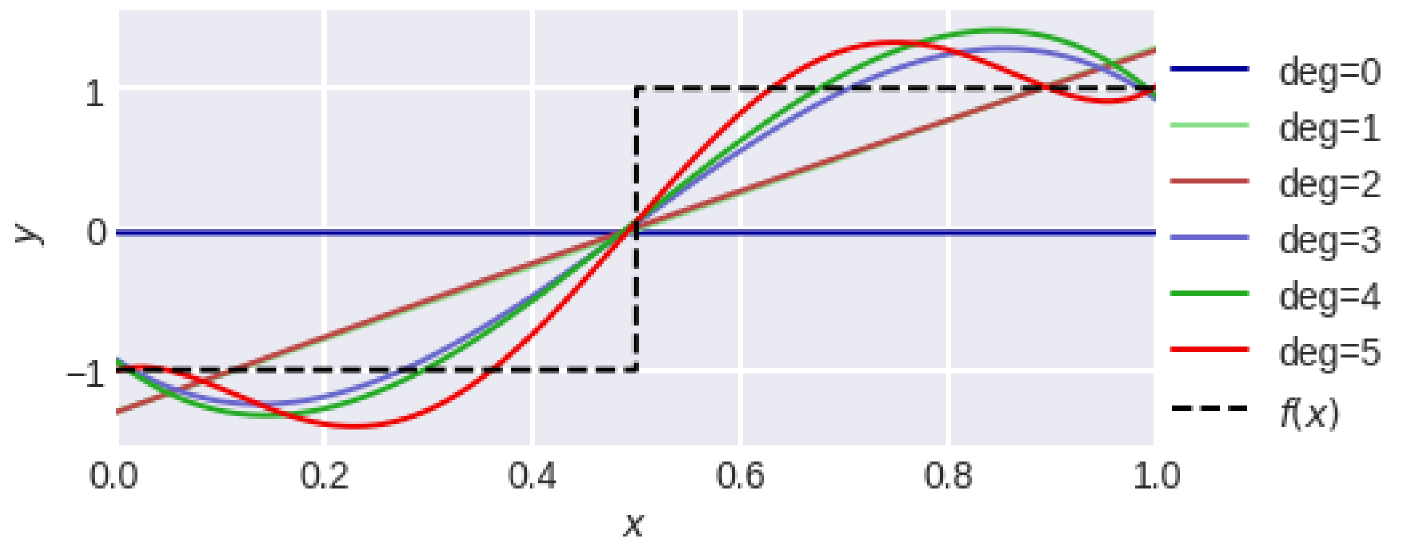
Пример: batch normalization

В чём нас обманывают…



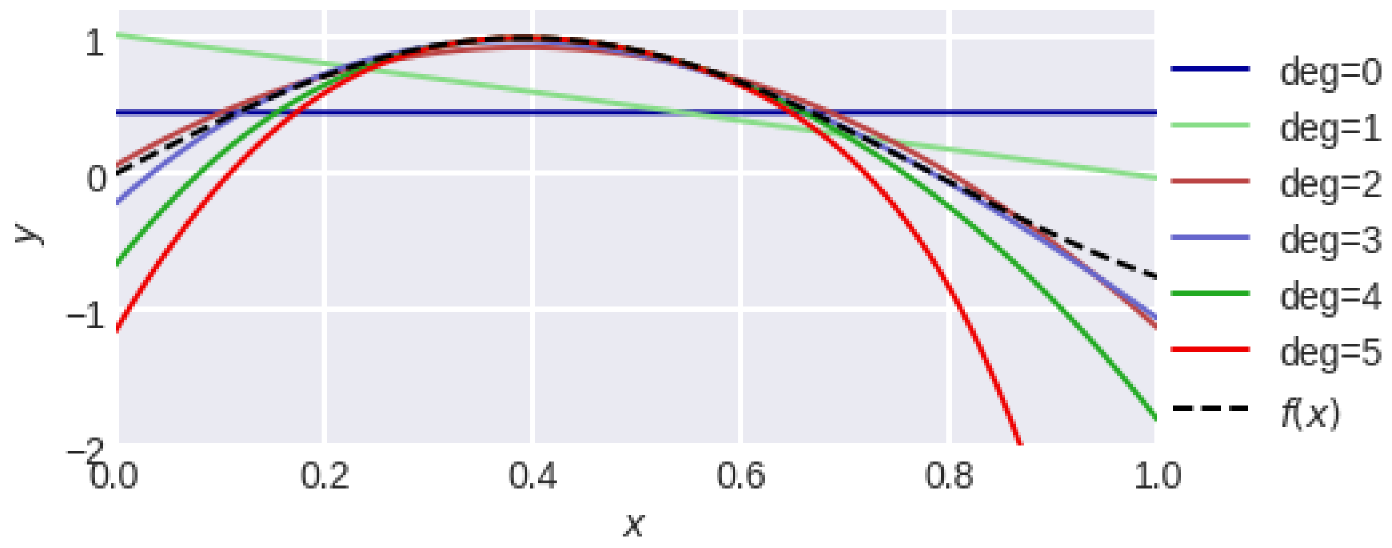
вспомним картинку – проведём эксперимент

В чём нас обманывают…

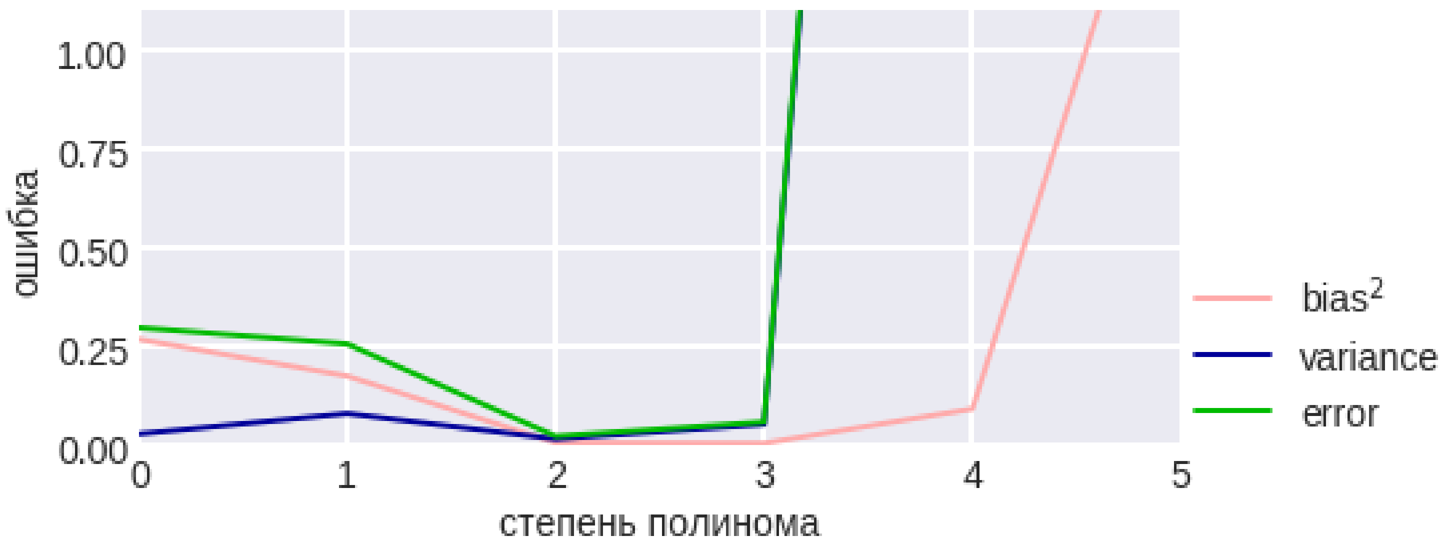


Это матожидания наших полиномиальных моделей!

В чём нас обманывают…



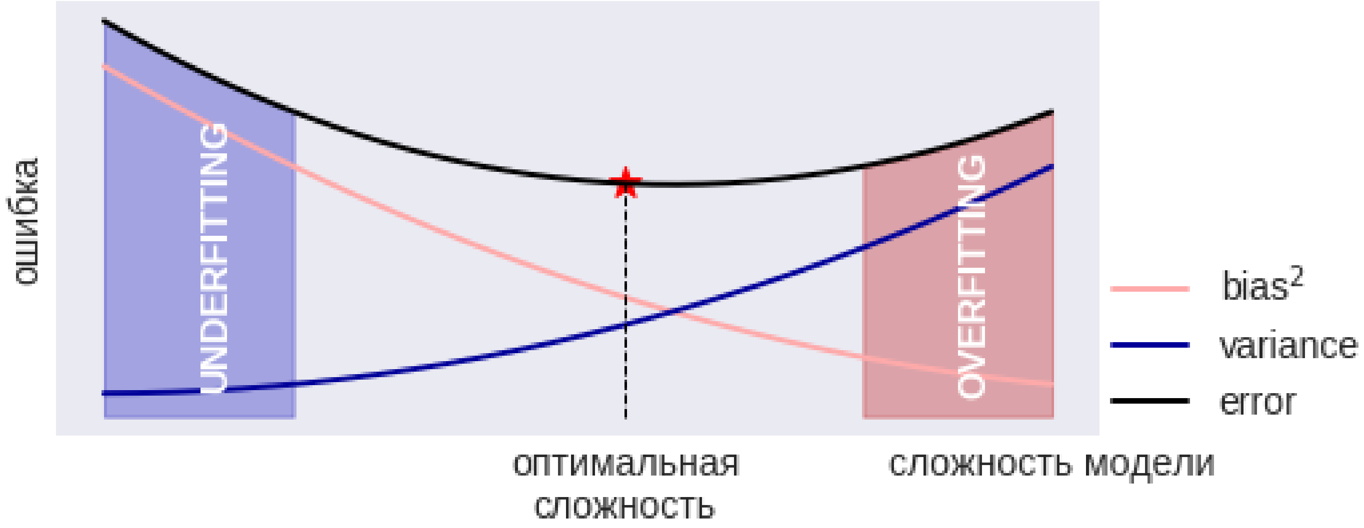
Полиномы 2й степени «самые лёгкие…»



В чём нас обманывают…

Степень полинома – «естественная» мера его сложности

Но тогда классической картинки мы не видим!



- **общая ошибка может не быть неунимодальной («слегка»)**
- **смещение и разброс могут не быть строго монотонными!**
- **смещение может возрастать при увеличении сложности!**

Может быть (и это нормально) – сложность модели относительно данных!

Вспомним… сложность реализации схемы в конкретном базисе

Итоги

«Перенастройка» – главная проблема ML, но есть и проблема недонастройки

Всё, казалось бы, регулируется сложностью…
но есть ещё много трюков с данными
(увеличение выборки, аугментация, специальная структура и т.п.)

Понятие «сложность» тоже зависит от данных

Ссылки

Классика ML

Hastie T., Tibshirani R., Friedman J. «The Elements of Statistical Learning» – 2009.

Почти упрощённый конспект лекции

https://dyakonov.org/2018/04/25/смещение-bias-и-разброс-variance-модели-алгорит/