Машинное обучение: оценка методов обучения с учителем

И. Куралёнок, Н. Поваров

Яндекс

СПб, 2013

Задача на сегодня

Задача: Есть метод обучения и данные, на которых обучаемся. 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

Источник проблемы

$$F_0 = \arg\max_{F(D)} \mu_{\xi \sim U(\Gamma)} T(y_\xi, F(x_\xi))$$

Мы обучаемся на одном множестве, а работаем на другом. А что, если эти множества отличаются?

Свойства выборки

“Иными словами, репрезентативная выборка представляет собой микрокосм, меньшую по размеру, но точную модель генеральной совокупности, которую она должна отражать.”

— Дж. Б. Мангейм, Р. К. Рич

Такого сложно достичь, поэтому хотим лишь “несмещенности” по параметрам обучения:

$$\begin{aligned} F_0 = & \quad \arg\max_F \mu_{\xi \sim U(D)} T(y_\xi, F(x_\xi)) \\ = & \quad \arg\max_F \mu_{\xi \sim U(\Gamma)} T(y_\xi, F(x_\xi)) \end{aligned}$$

Способы повлиять на несмещенность

Интересно получить выборку, несмещенную (смещенную не более чем ...) по результатам процедуры обучения:

- Найти «хороший» способ генерации выборки при условии процедуры подбора
- Наложить ограничения на процедуру подбора
- Ограничения на решающую функцию

$\Rightarrow$ **Надо научиться мерять смещенность выборки, и чем тоньше изменения, тем более точный инструмент измерения нужен.**

Известные способы оценки

Оценка по принципу “чёрного ящика”:

- Оценка в боевых условиях (на пользователях)
- Кросс-валидация
- Повторные выборки

Оценка по принципу “прозрачного ящика”:

- VC-оценки
- PAC-Bayes bounds
- Оценки по Воронцову

Оценка в боевых условиях

Как оценить, насколько пользователяю системы “хорошо” по его поведению? Конкретная реализация зависит от области, но можно выделить типы:

- blind testing;
- A/B тестирование (Abandonment Rate, MRR by long clicks, etc.);
- подстроенные результаты (TDI, BI, etc.).

Мы можем долго-долго рассказывать байки в этом месте :).

Cross-fold validation I

Разобьем множество X на два L и T так, чтобы $L \cup T = X, L \cap T = \emptyset$ случайным образом (Jackknife). Будем обучаться на одной половине а проверять результат обучения на другой.

+ простой и надежный
+ позволяет оценить распределение на множестве решений
– последовательные эксперименты зависимы
– используем мало данных для обучения
– непонятно как подбирать соотношения $\frac{|L|}{|T|}$

Cross-fold validation II

Можно организовать разными способами:

- 2-fold обычно так и делаем
- k-fold когда очень боимся зависимости экспериментов, а данных много
- Leave-one-out (LOO) когда совсем мало данных

Повторные выборки

В статисике продолжением метода Jackknife стал Bootstrapping. Сформируем 2 множества L и T так, что:

1. $|L| = |T| = |X|$
2. $l_i \sim U(X), t_i \sim U(X)$

Будем считать, что эти 2 множества разные.

+ используем полный объем выборки
+ на больших объемах вариативность выбора огромна
– теперь еще и T зависит от L и как это учесть – не ясно

$\Rightarrow$ можно применять только на больших объемах (>10k точек)

Как принять решение по результатам CV/ПВ

Не стоит забывать, что результаты CV/ПВ экспериментов всегда *зависимы*, и не спасают нас от смещенности исходной выборки.

1. провести серию последовательных эксприменто;
2. закрыв глаза на зависимость, построить доверительные интервалы на $T(F_0, T)$;
3. чтобы сравнить два альтернативных метода можно проверить

$$H_0 : \mu\left(T(F_{0_A}, T)\right) = \mu\left(T(F_{0_B}, T)\right)$$

например с помощью парного WX-test'а.

Сложность модели

Чем больше в модели параметров, тем большую информацию они несут.

— Ваш К.О.

Какая бывает информация в параметрах:

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

Если мы будем усложнять модель, соотношения информации будут двигаться.
$\Rightarrow$ **Хотим придумать рычажок, который контролирует сложность модели.**

Семейство полиномов p-й степени

Будем строить семейства решающих функций следующим образом:

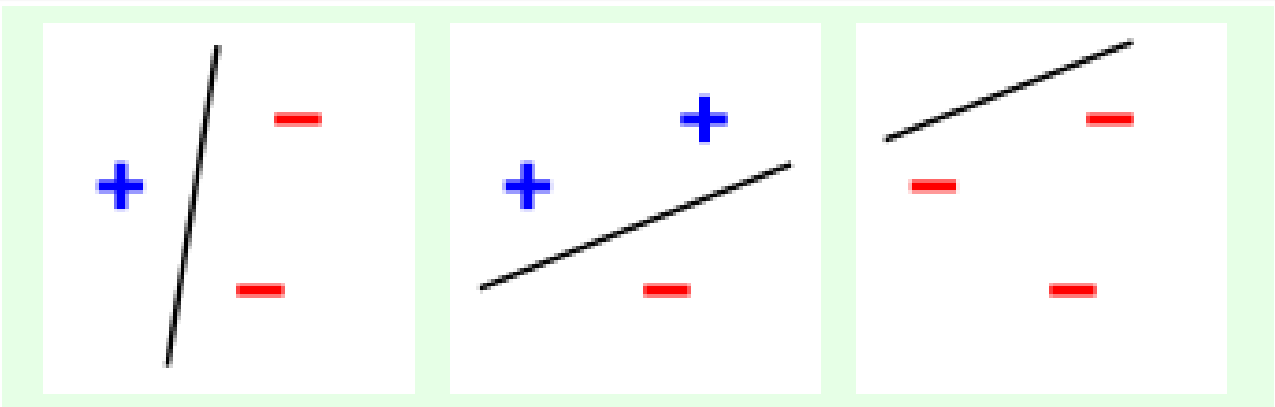
$$F(x, \lambda_1 = \{w\}) = w^T x$$
$$F(x, \lambda_2 = \{w, A\}) = w^T x + x^T A x$$
$$F(x, \lambda_3 = \{w, A, B\}) = w^T x + x^T A x + \sum_i \sum_j \sum_k b_{ijk} x_i x_j x_k$$
etc.

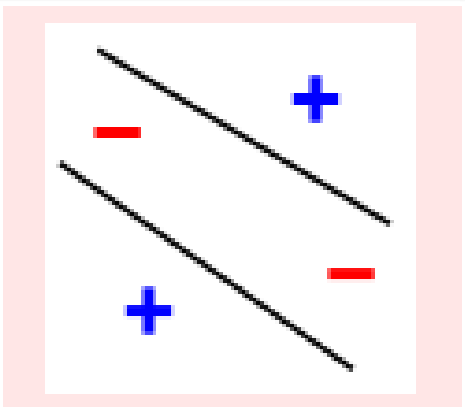
Понятно, что $|\lambda_i| < |\lambda_{i+1}|$.

Размерность Вапника-Червоненкиса

Definition (ru.wikipedia.org)

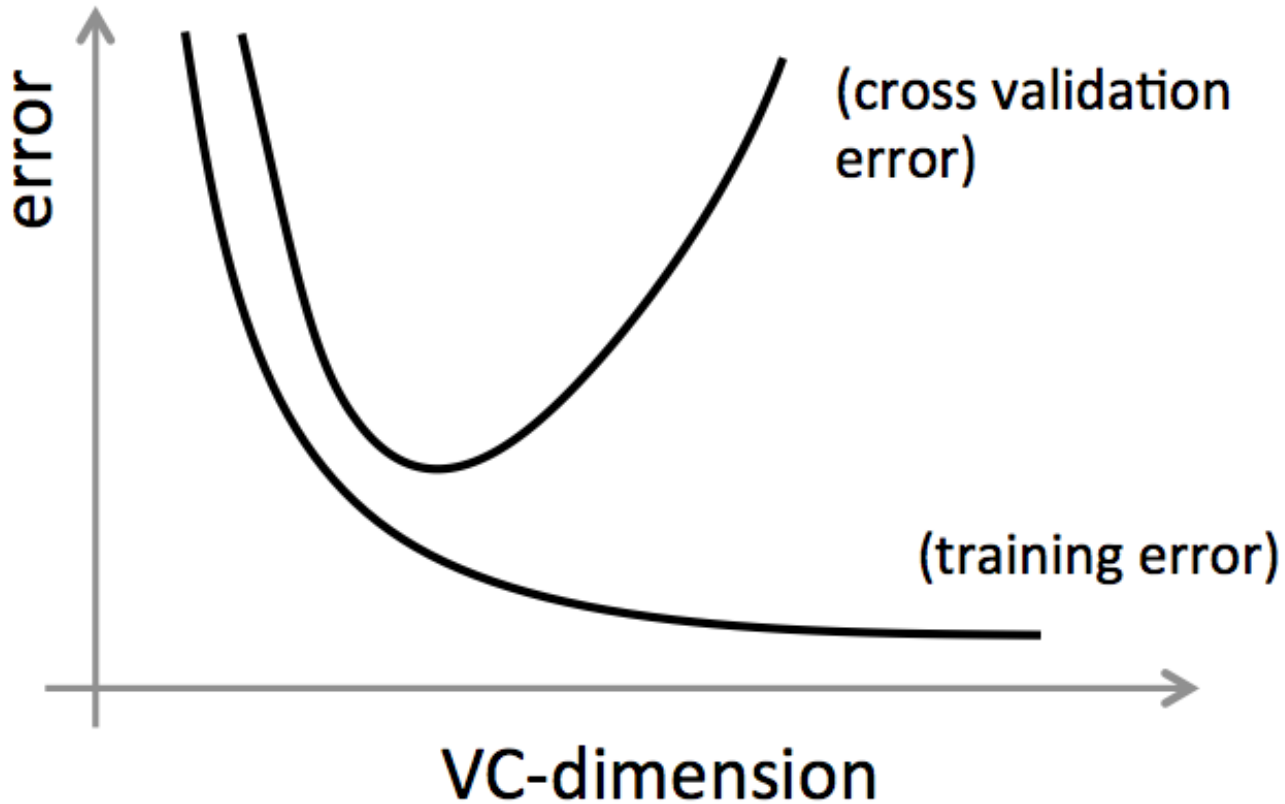
VC-размерность класса функций F — наибольшее количество точек, которое может быть разделено функциями семейства, вне зависимости от конфигурации множества точек.





картинки с en.wikipedia.org

Overfit vs. underfit



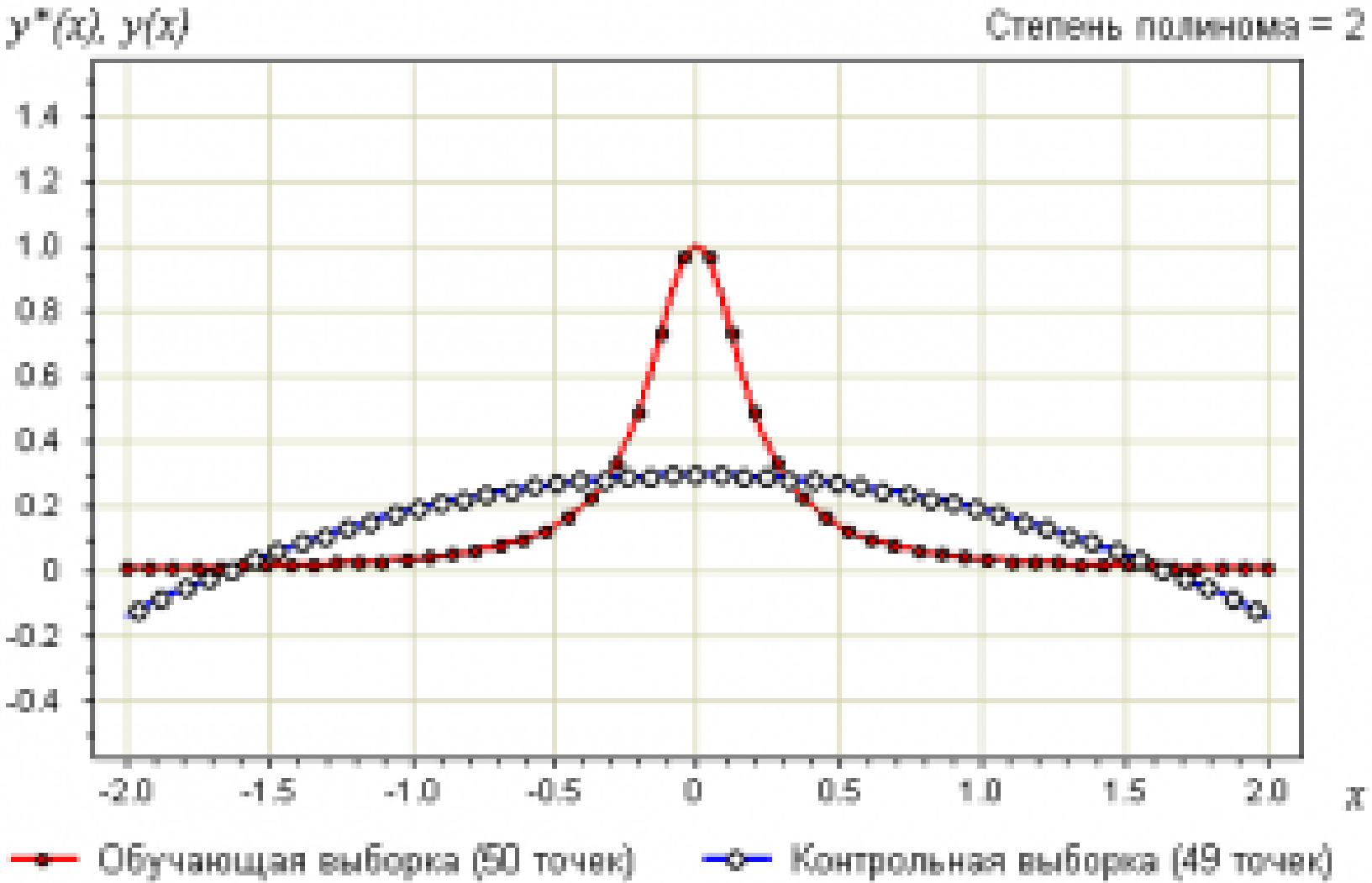
Overfit vs. underfit определения

Переобучение, переподгонка (overtraining, overfitting) — 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.

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

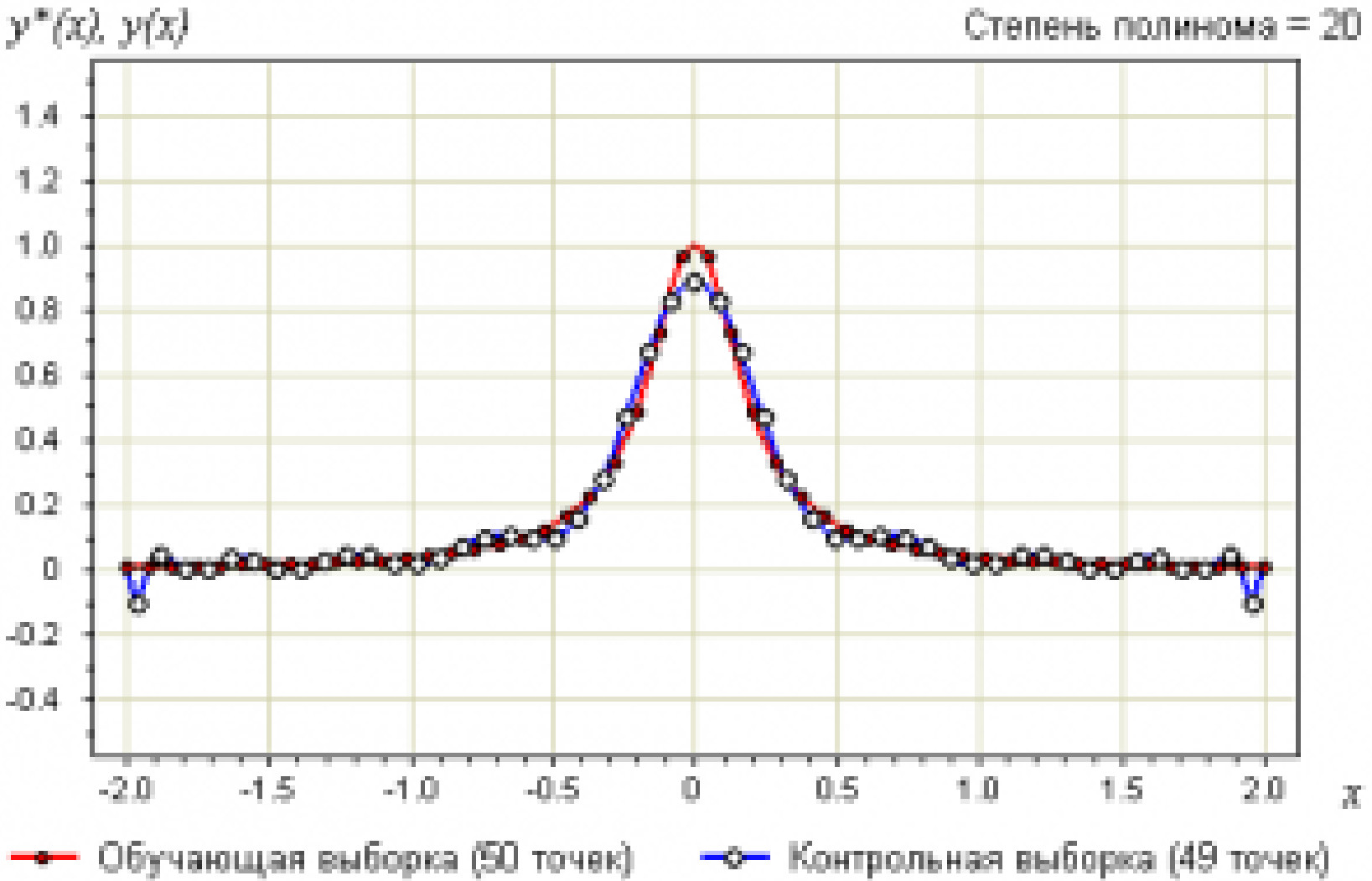
— **machinelearning.ru**

Как это выглядит в полиномах (underfit)



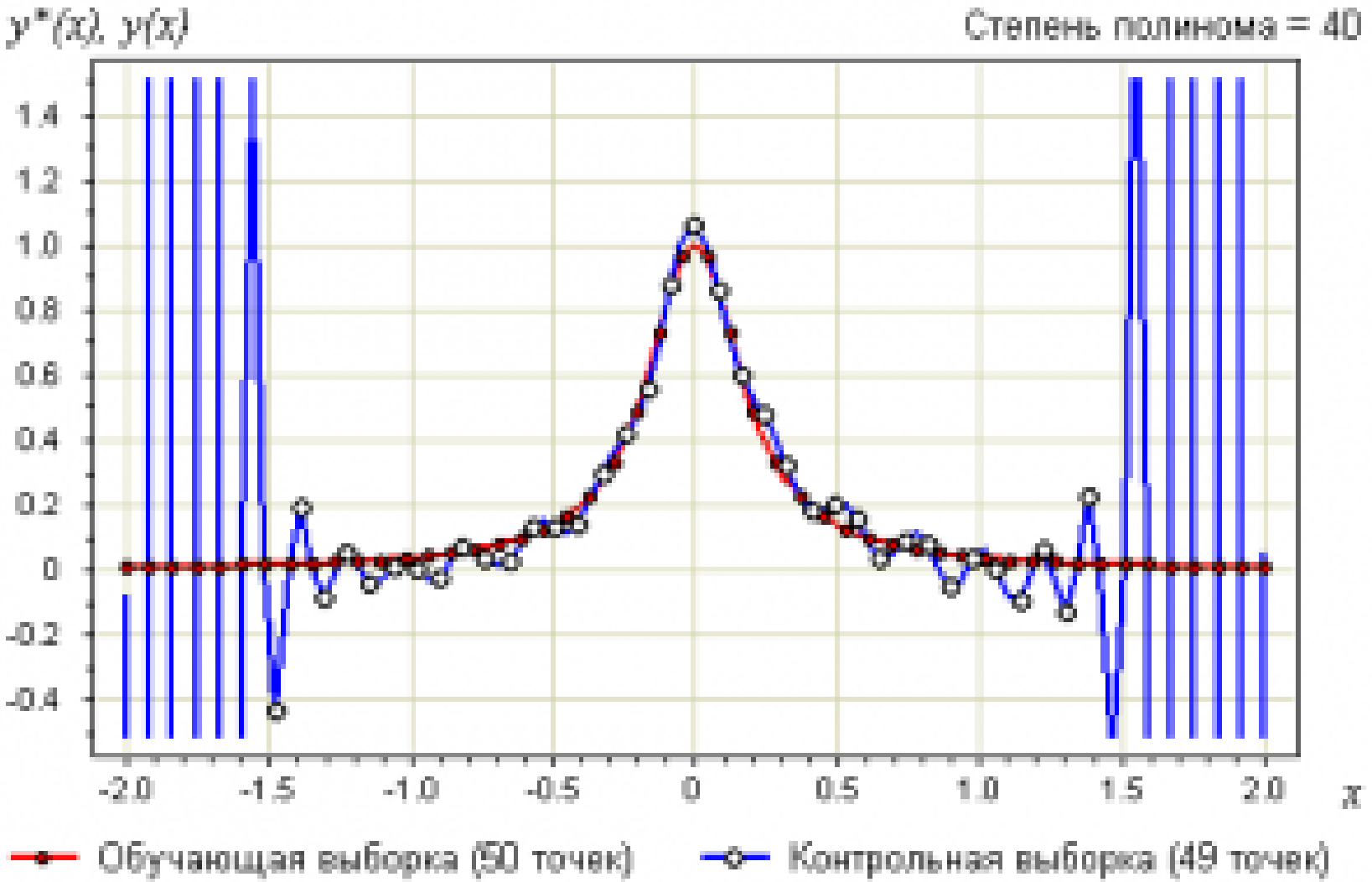
картинка с machinelearning.ru

Как это выглядит в полиномах (fit)



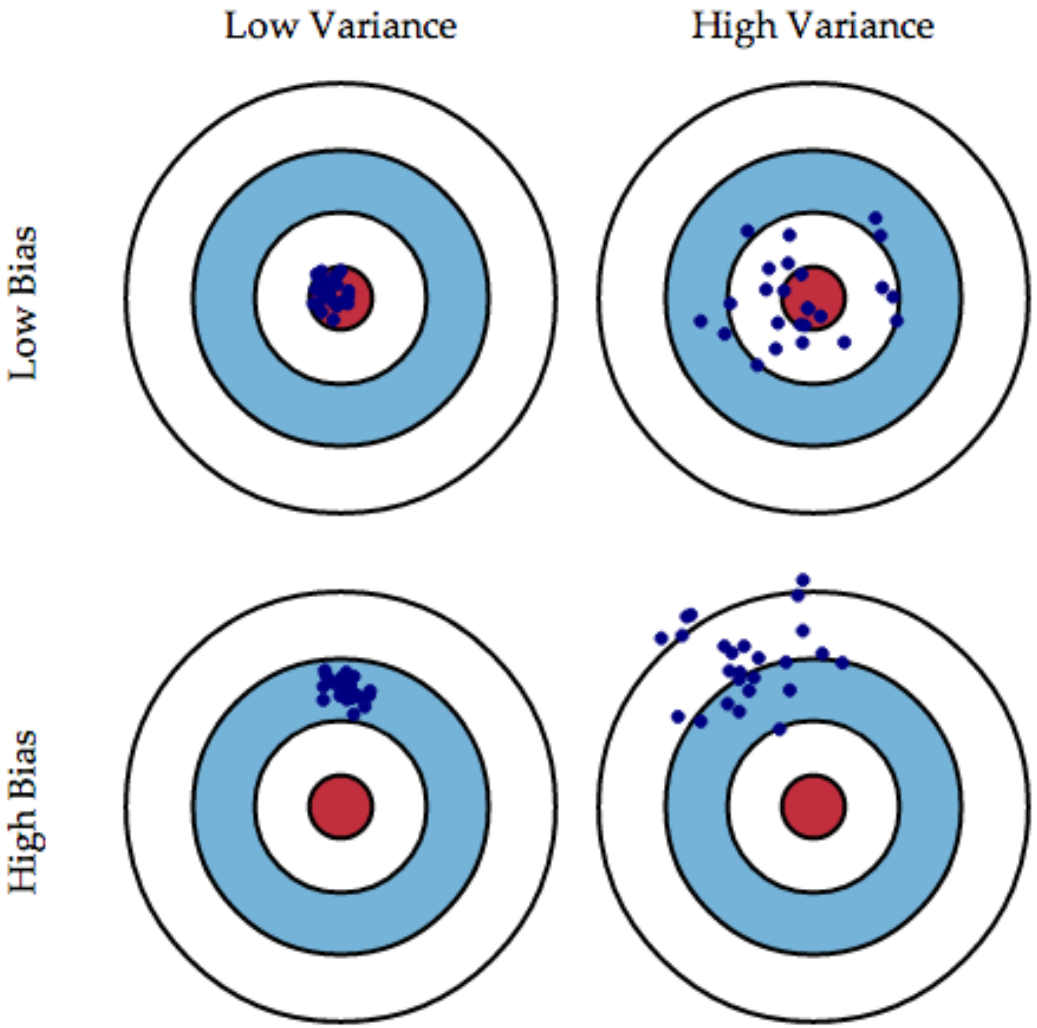
картинка с machinelearning.ru

Как это выглядит в полиномах (overfit)



картинка с machinelearning.ru

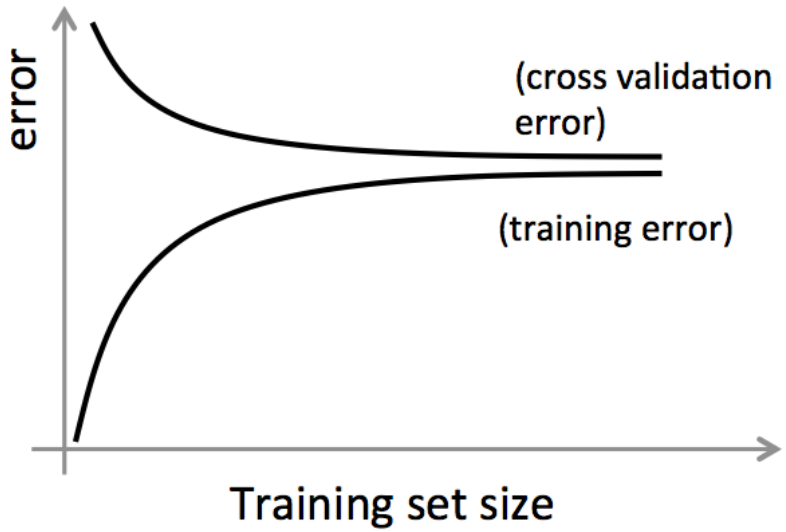
Как это выглядит в пространстве решений

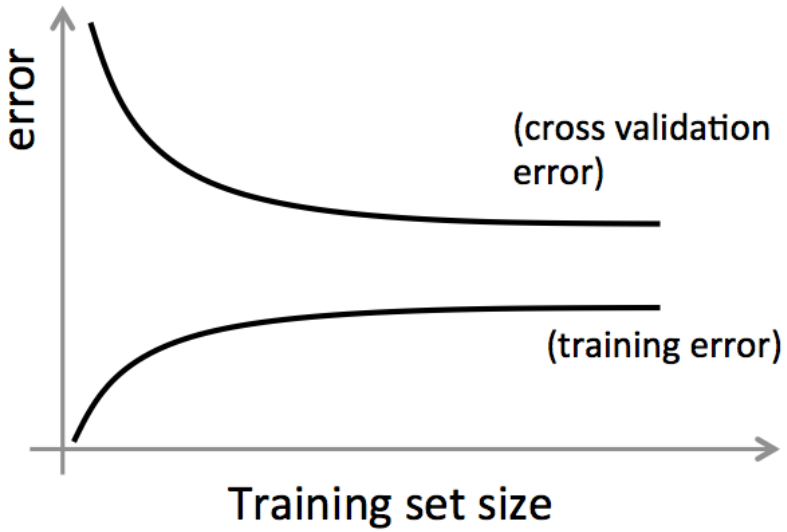


Что в какой ситуации делать

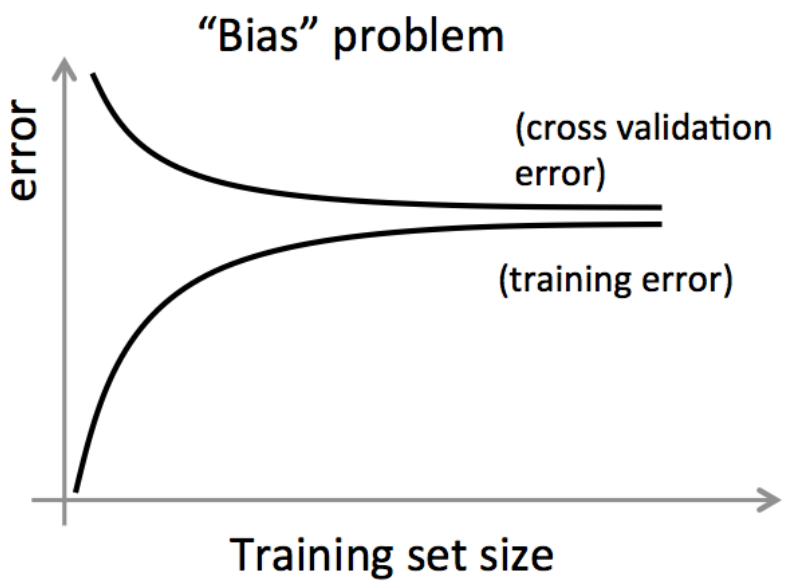
- Увеличение числа примеров для обучения фиксят high variance, но не high bias
- Меньшее число факторов фиксят high variance, но не high bias
- Увеличение числа факторов фиксят high bias, но не high variance
- Добавление степени к полиному и взаимодействующих факторов фиксят high bias, но не high variance

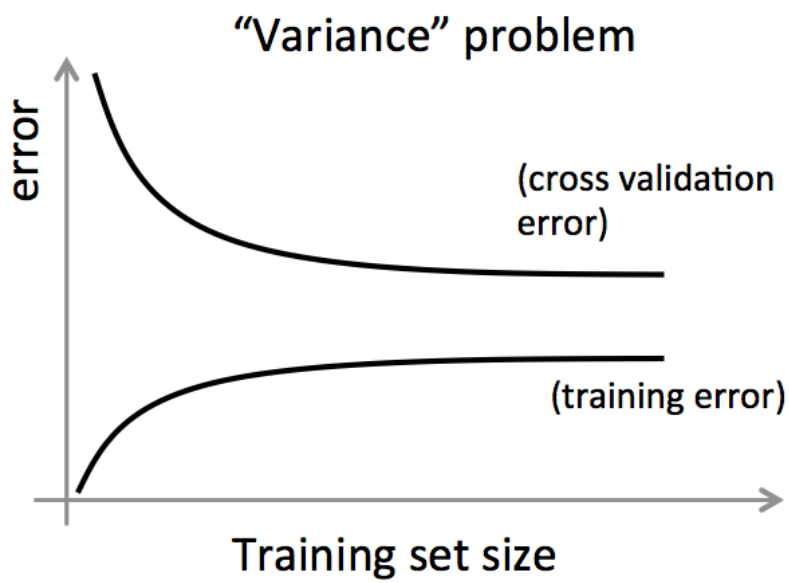
Понимаем в какой ситуации находимся





Понимаем в какой ситуации находимся





Теоретическая оценка

Цели оценки:

- Можно ли понять какой метод круче для заданного объема данных
- Предсказать сложность на которой достигается fit

Будем рассматривать задачу классификации и определять долю ошибок ($T = -\sum_i I(y_i \neq F(x_i))$).

Русские тоже что-то могут в ML :)

Владимир Наумович Вапник (NEC[1]),
Алексей Яковлевич Червоненкис (University of London).
Разработали первую теорию по оценке переобучения в зависимости от класса решающих функций в 60-70е годы.

[1] Здесь и далее по данным ru.wikipedia.org

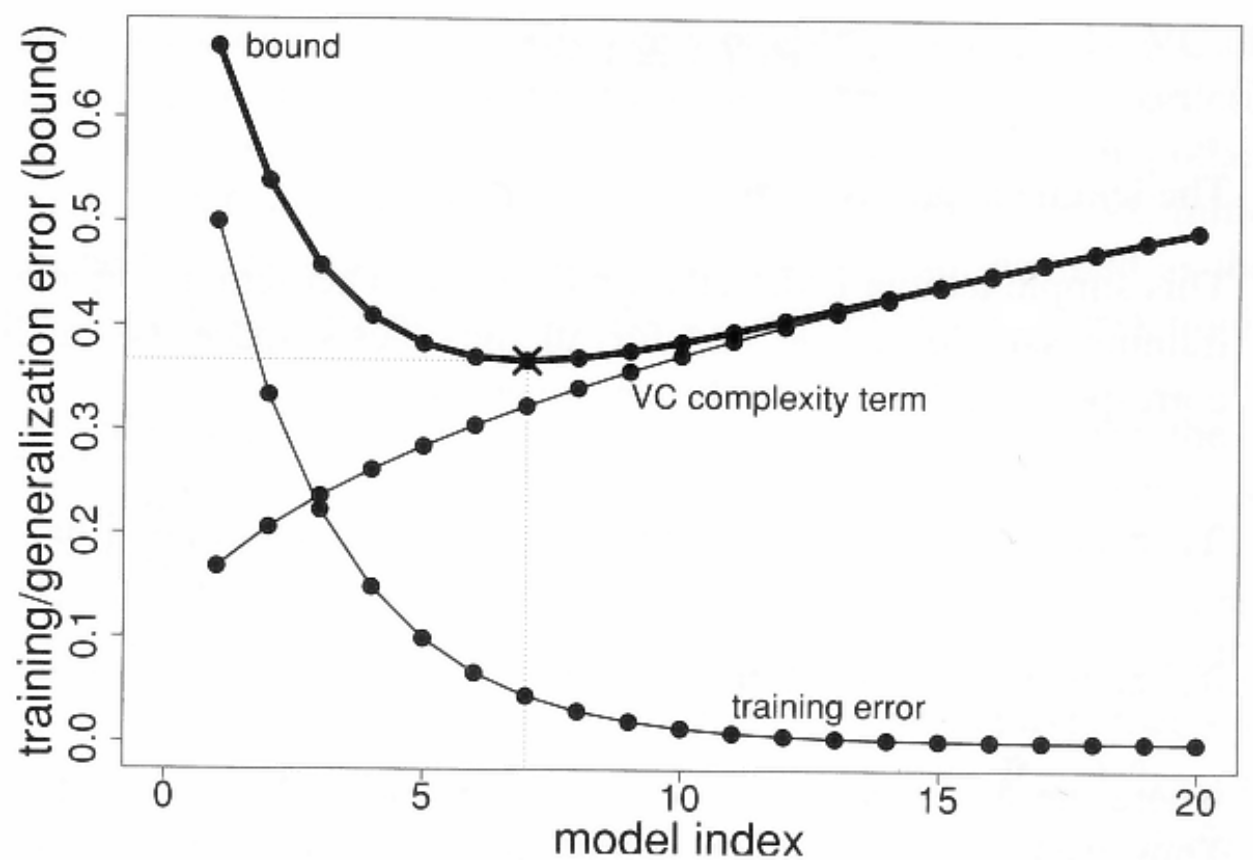
Оценка по методу Вапника-Червоненкиса I

Хотим оценить $T(T)$ сверху по $T(L)$ и свойствам семейства решающих функций.

$$T(T) \geq T(L) - \sqrt{\frac{h(\log(\frac{2n}{h}) + 1) - \log(\frac{\eta}{4})}{n}}$$

с вероятностью $1 - \eta$, где h – VC-оценка семейства, $n = |L|$.

Оценка по методу Вапника-Червоненкиса II



картинка с www.svms.org

PAC-Bayes bounds I

Кажется, что не все определяет только VC-размерность. Для одних данных работают одни решения, для других – другие.

Definition (Оценка McAllester'a (1998))

Введем априорное распределение над семейством решающих функций F. Тогда:

$$T(F_0, T) \geq T(F_0, L) - \sqrt{\frac{log\frac{1}{p(F_0)} + log(\frac{1}{\eta})}{2n}}$$

с вероятностью не меньше $1 - \eta$.

PAC-Bayes bounds II

Рассмотрим решение, как взвешивание между разными функциями семейства F.[2] Тогда получится еще краше (современный вид):

$$|\mu_\rho T(F_i, T) - \mu_\rho T(F_i, L)| \geq \sqrt{\frac{KL(\rho\|\pi) + log\frac{4n}{\eta}}{2n-1}}$$

с вероятностью не меньше $1 - \eta$, где ρ — априорное распределение в семействе F, π — апостериорное.

[2] Это такое байесовское решение, о котором мы поговорим в следующий раз

Оценка в слабой аксиоматике Воронцова

Воронцов Константин Вячеславович (ШАД в Москве, machinelearning.ru) написал докторскую диссертацию на тему

“Комбинаторная теория надёжности обучения по прецедентам”

в которой предложил интересную альтернативу VC-оценкам и PAC-Bayes.

Соображения об точности решения

До этого мы говорили о точных решениях. Но точность определения параметров обучения определяет количество информации в решении.
$\Rightarrow$ Не надо вычислять параметры решения до 100-го знака: sizeof(long double) $\gg$ sizeof(float)

В задачах итеративной оптимизации мы привыкли устремлять шаг к 0, что в случае ML приводит к большему variance решений.

Как еще можно переобучиться?

Будем долго и упорно подбирать метод обучения на фиксированном делении L, T:

$$\max_i \left(\arg\max_{f \in F_i} T(f, L) \right) (T)$$

Это же максимизация на всем множестве $L \cup T = X$!
Называется такое overfit on validate. Что с этим делать? Исследовать

$$\cancel{\max} \left(\arg\max_i \left(\arg\max_{f \in F_i} T(f, L) \right) (V) \right) (T)$$

Где взять данные для экспериментов

- Реальные данные
  - Поиск: РОМИП, TREC, Яндекс.ИМАТ, Yahoo LTRCh
  - Pascal Challenge
  - InnoCentive
  - Kaggle
- Синтетические данные (многомерный XOR)
- “Загадки”: задумаем «хитрое» распределение и попробуем его отгадать

Задача

Дано:

- $L = 1000$ точек, полученных по правилу:

$$x \sim U(0, 10]$$
$$y = ln(x)$$

- $T = 10000$ реализаций x для которых надо найти y

Задача: найти решение в классе полиномов оптимальной степени p, наилучшим образом приближающий y на T.

Где брать домашние задания

- svn checkout http://ml-lections.googlecode.com/svn/trunk/ml-lections-read-only
- Комитить не получится ;)
- Бонусом - лекции в tex.
- Лекции находятся в разных папках. В папке лекции есть папка homework.
- Помимо датасетов содержат файл howto.txt
- Вопросы: saintnik@yandex-team.ru