Машинное обучение: оценка методов обучения с учителем

И. Куралёнок, Н. Поваров

Яндекс

СПб, 2014

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

Вспомним задачу МО

$$\arg \max_{F,B:\hat{F}=B(F)} A(\Gamma, \hat{F})$$

- A — цели эксплуатации (например деньги) на всей области работы Γ
- B — способ оптимизации, который реализуем

Известные способы оценки

Оценка эксплуатации (по принципу “чёрного ящика”):

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

Оценка обучения (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 :).

Есть ли альтернативы

Можно ли как-то обойтись без членовредительства?

Не всегда “боевые условия” доступны:

- Люди не любят быть объектом экспериментирования
- Качество эксперимента может быть сильно хуже продакшена
- Количество одновременных экспериментов ограничено

$\Rightarrow$ Как бы нам эмулировать эксплуатацию?

Cross-fold validation I

Определение

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

Cross-fold validation II

Свойства

+ простой и надежный
+ позволяет оценить распределение на множестве решений
– последовательные эксперименты зависимы
– используем мало данных для обучения
– непонятно как подбирать соотношения $\frac{|L|}{|T|}$
– результаты рассказывают про эксплуатацию с точностью до репрезентативности множества X

Cross-fold validation III

Способы реализации

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

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*. С точностью до этого алгоритм такой:

1. провести серию последовательных экспериментов;
2. проверить

$$H_0 : \mathbb{E}\left(\mu(\hat{h}_A, T)\right) = \mathbb{E}\left(\mu(\hat{h}_B, T)\right)$$

   например с помощью парного WX-test'а;
3. исследовать средний знак разницы $\mu(\hat{h}_A, T) - \mu(\hat{h}_B, T)$.

Ввели понятия “хорошо” и “плохо”

$\Rightarrow$ **Попробуем понять почему один метод работает лучше чем другой.**

Источник проблемы

$$\hat{h} = \arg\max_h \mathbb{E}_{\xi \sim \Gamma} \mu(y_\xi, h(x_\xi))$$

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

Свойства выборки

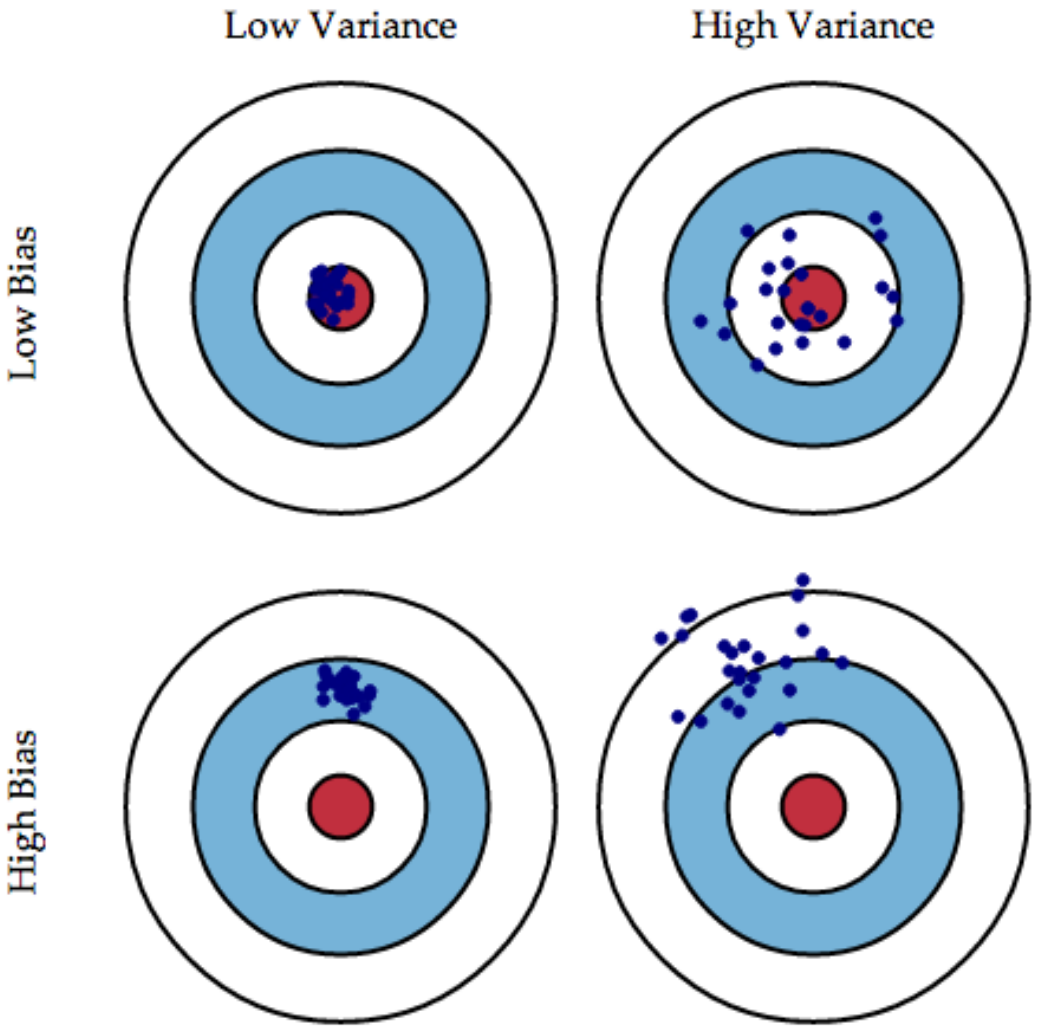
“Иными словами, репрезентативная выборка представляет собой микрокосм, меньшую по размеру, но точную модель генеральной совокупности, которую она должна отражать.”

— Дж. Б. Мангейм, Р. К. Рич

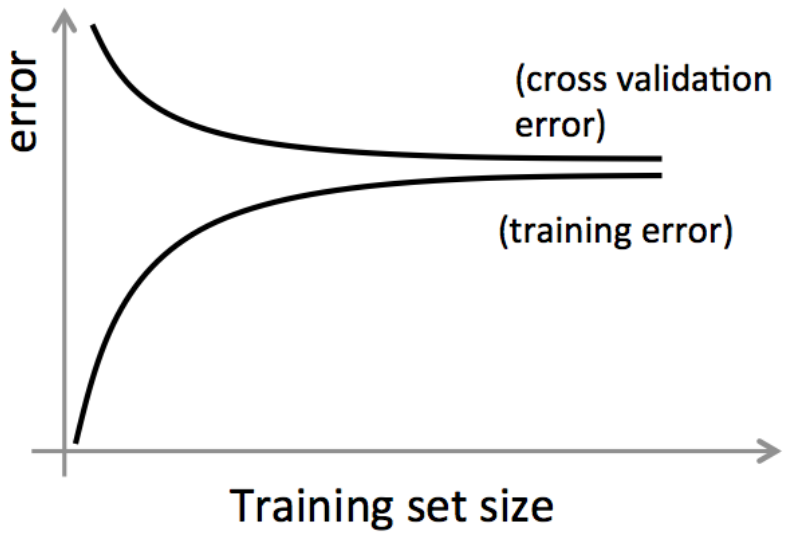
Такого сложно достичь, поэтому хотим лишь “несмещенности” по параметрам обучения:

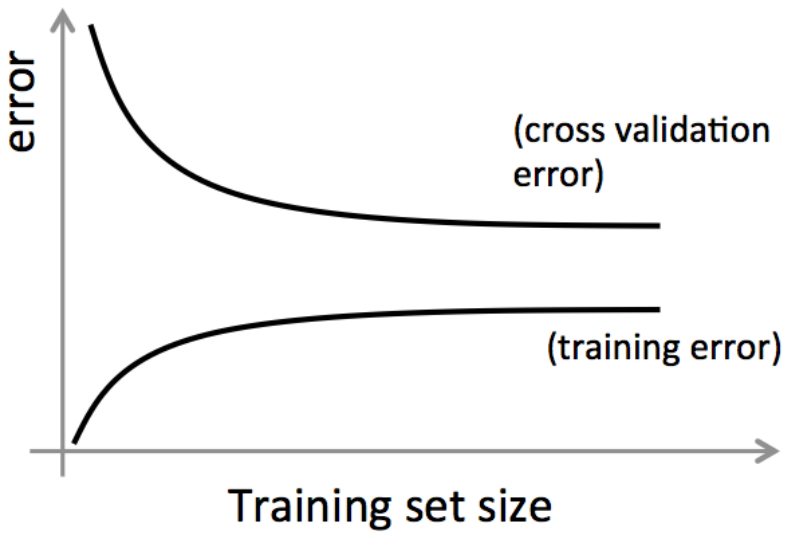
$$\begin{aligned}\hat{h} = &\ \arg\max_h \mathbb{E}_{\xi\sim D}\mu(y_\xi, h(x_\xi)) \\ = &\ \arg\max_h \mathbb{E}_{\xi\sim \Gamma}\mu(y_\xi, h(x_\xi))\end{aligned}$$

Как это выглядит в пространстве решений

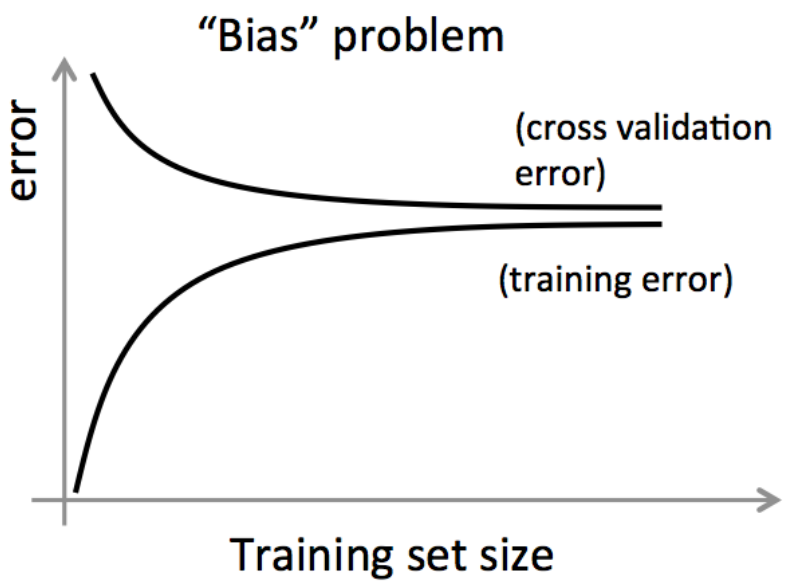


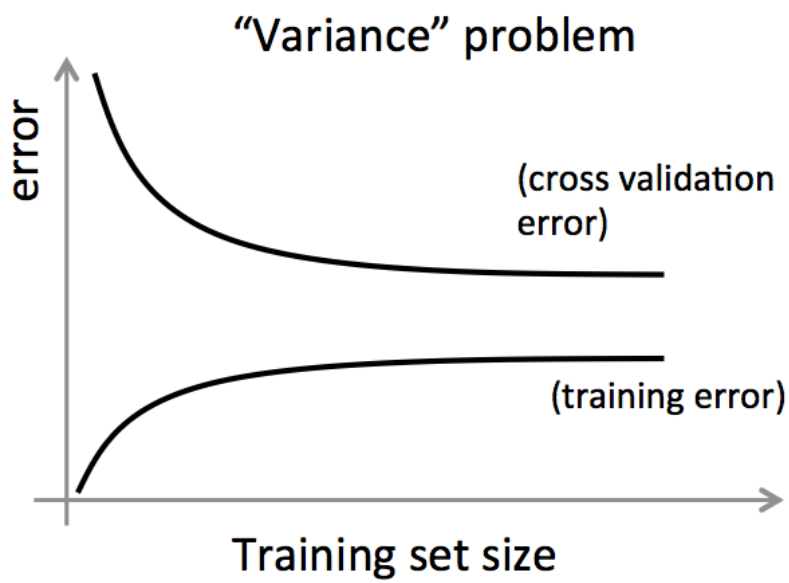
Понимаем в какой ситуации находимся





Понимаем в какой ситуации находимся





Что в какой ситуации делать

- В решающей функции:
  - усложнение может уменьшить bias
  - упрощение может уменьшить variance
- В пространстве задачи:
  - Меньшее число факторов могут уменьшить variance
  - Увеличение числа факторов может уменьшить bias
- Изменение целевой функции специальным образом может уменьшить variance
- Увеличение числа наблюдений может уменьшить variance

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:

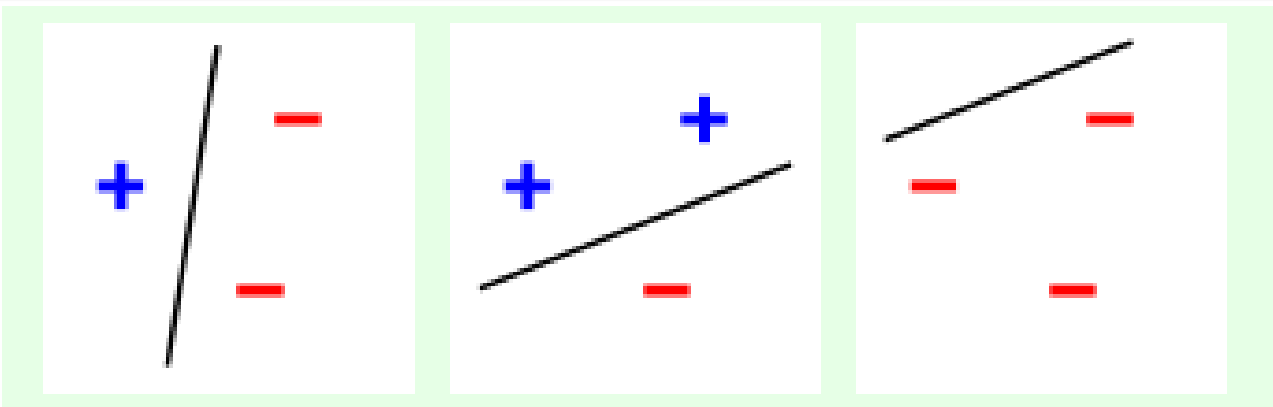
$$
\begin{aligned}
&F(x, \beta^1) = {\beta_1^1}^T x \\
&F(x, \beta^2) = {\beta_1^2}^T x + x^T \beta_2^2 x \\
&F(x, \beta^3) = {\beta_1^3}^T x + x^T \beta_2^3 x + \sum_i \sum_j \sum_k \beta_{3\,ijk}^3 x_i x_j x_k \\
&etc.
\end{aligned}
$$

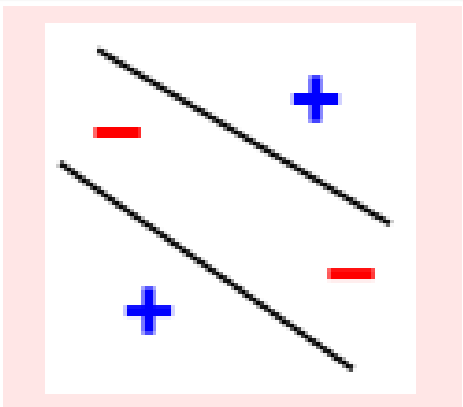
Понятно, что $dim(\beta_i) < dim(\beta_{i+1})$.

Размерность Вапника-Червоненкиса

Definition (ru.wikipedia.org)

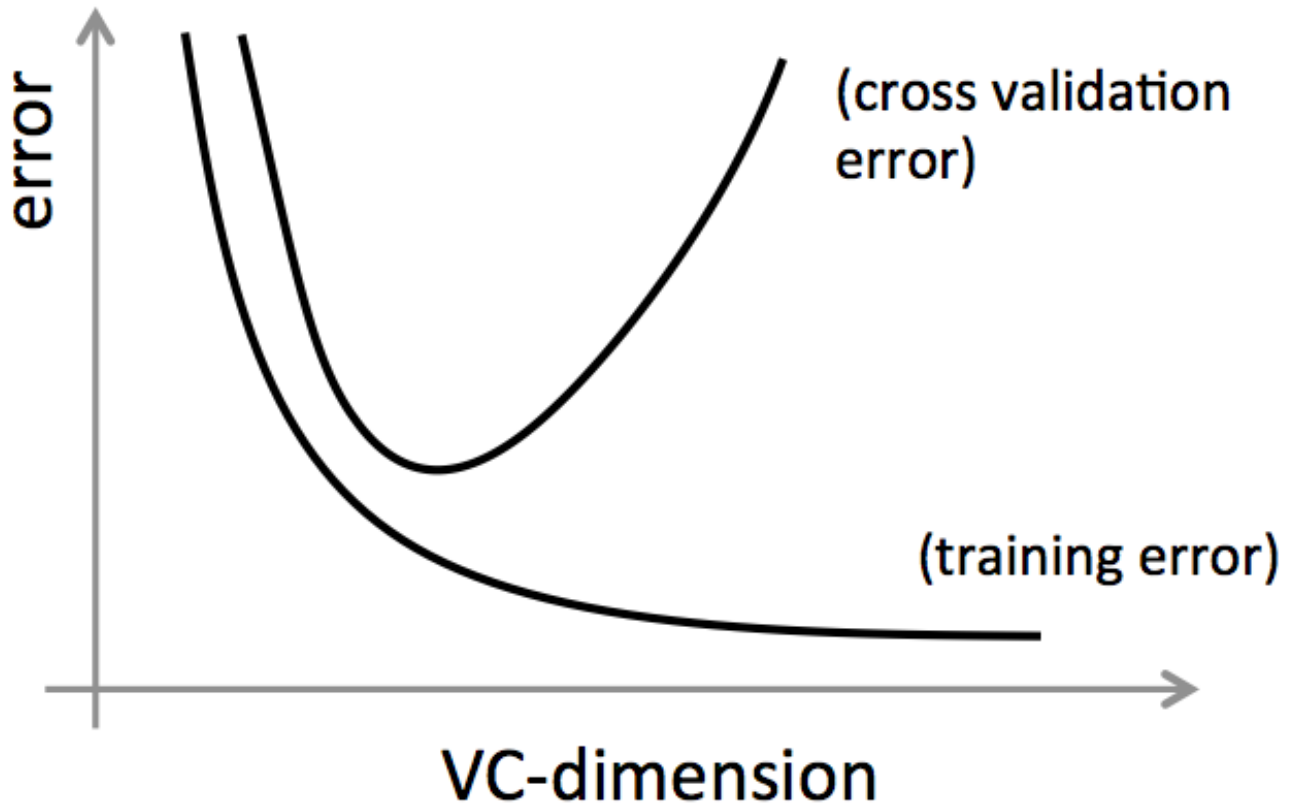
VC-размерность класса функций F — наибольшее количество точек, которое может быть разделено функциями семейства, вне зависимости от конфигурации множества точек.





картинки с en.wikipedia.org

Overfit vs. underfit



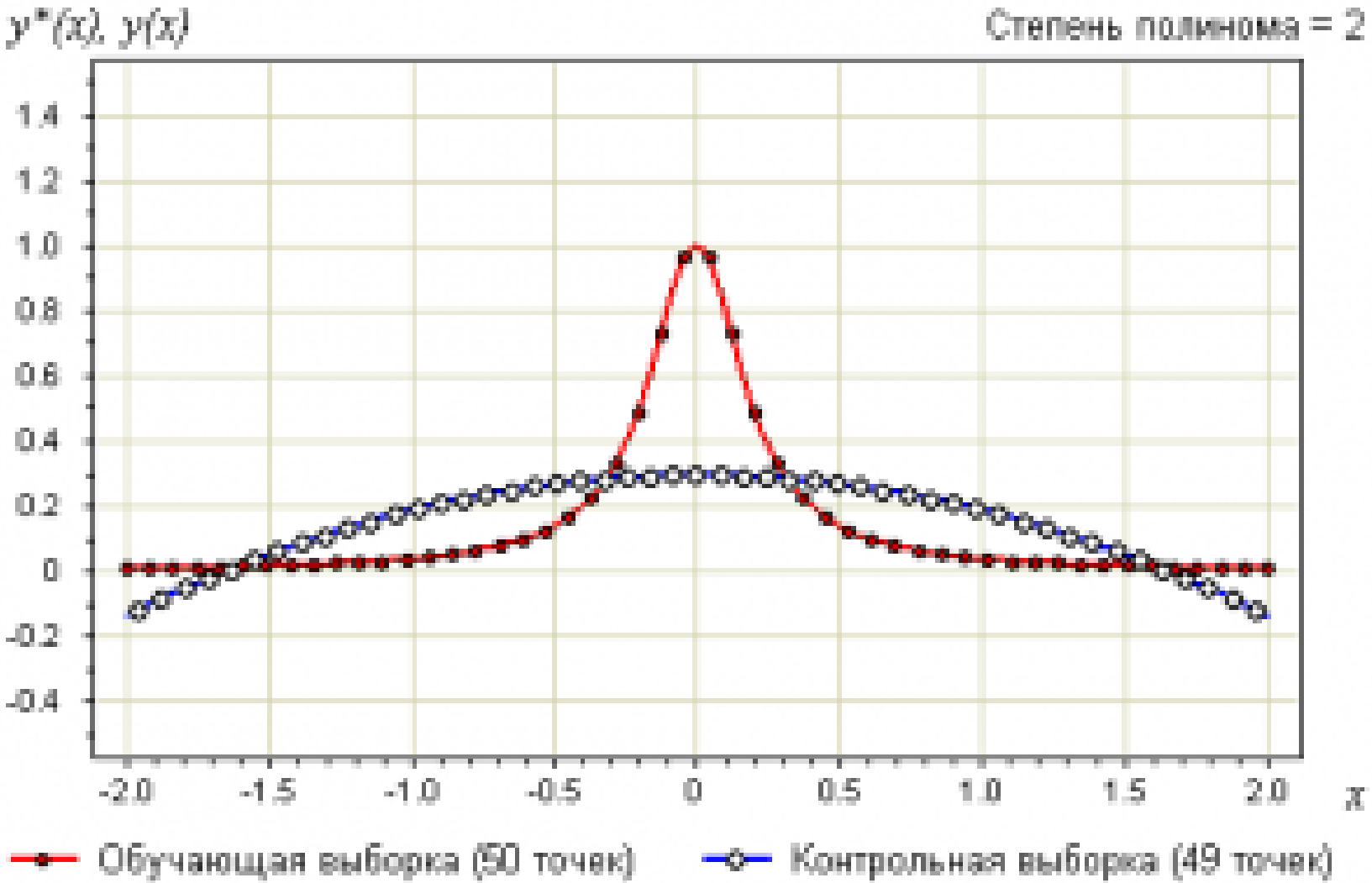
Overfit vs. underfit определения

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.

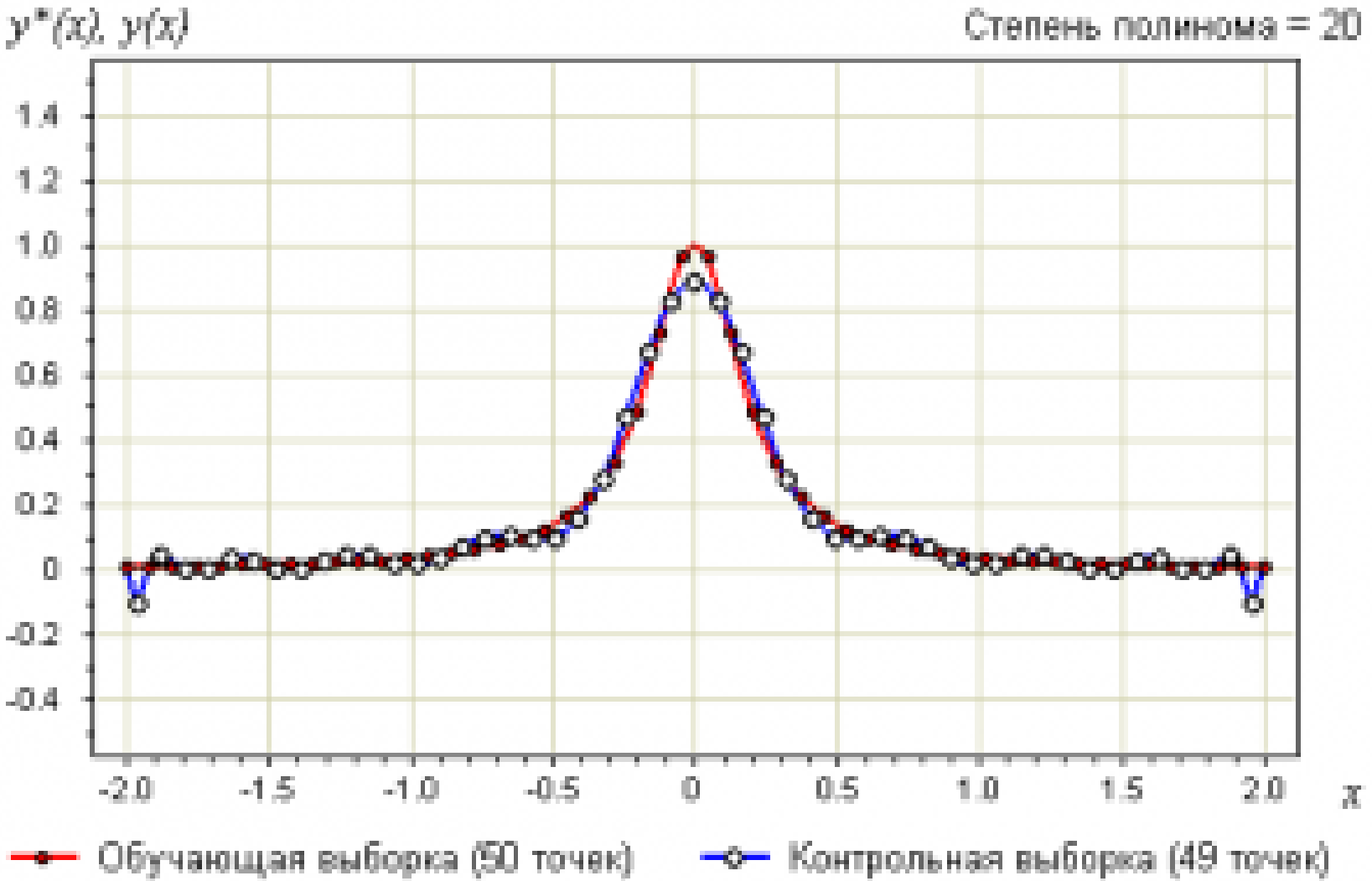
— **machinelearning.ru**

Как это выглядит в полиномах (underfit)



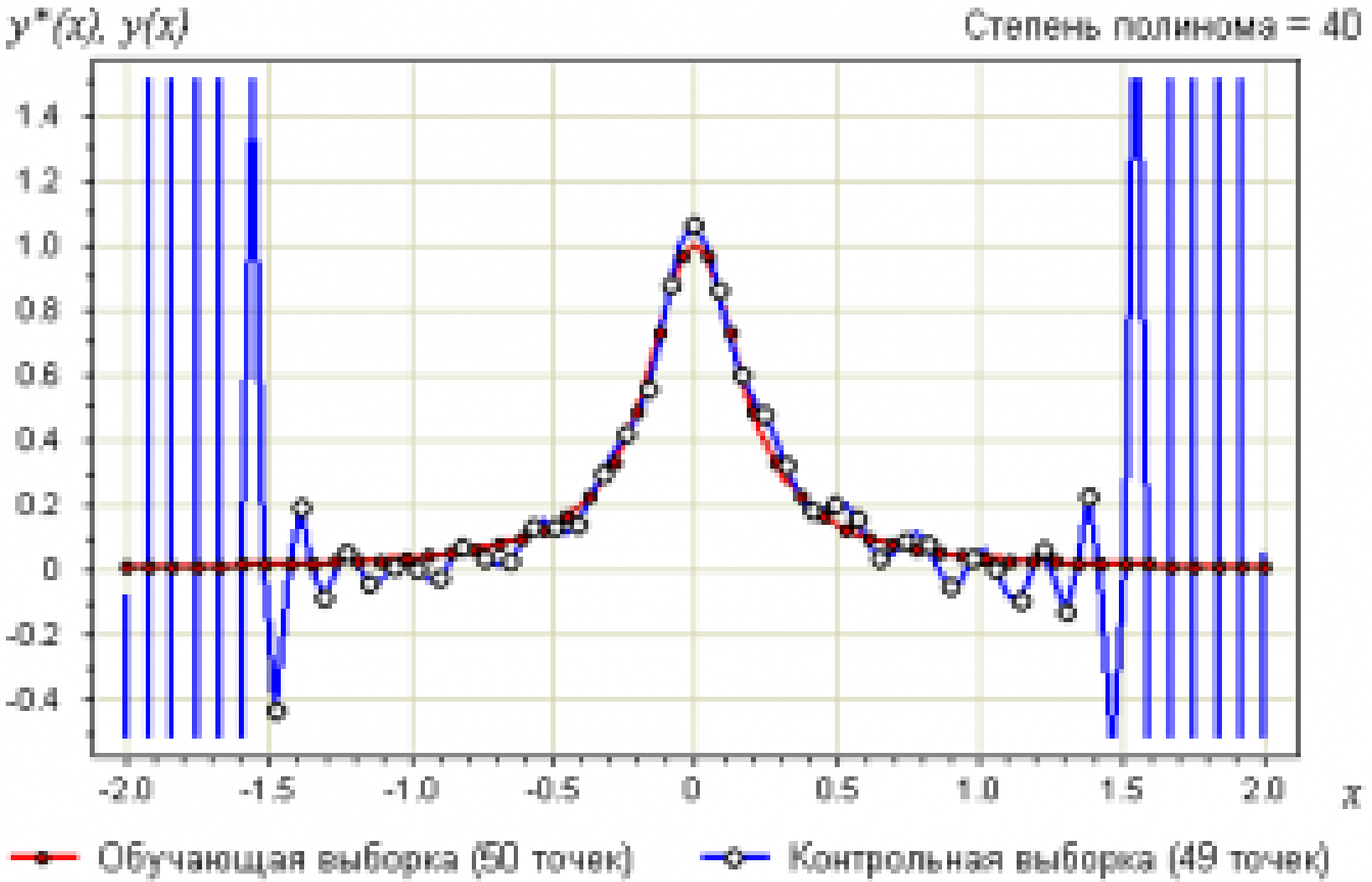
картинка с machinelearning.ru

Как это выглядит в полиномах (fit)



картинка с machinelearning.ru

Как это выглядит в полиномах (overfit)



картинка с machinelearning.ru

Принципы теоретической оценки

Подход к оценке:

- Загадаем какую-то функцию из одного класса
- По результатам работы загаданной функции, попробуем отгадать ее в том же классе, или в каком-то другом
- Сравним зависимость ошибки отгадывания от количества необходимых примеров
- Построим оценки на ошибку при заданном алгоритме отгадывания

Механизм теоретической оценки

Самый простой случай следующий:

- Будем рассматривать пространство $\{0,1\}^n$ и распределение D на этом пр-ве
- В рамках этого пространства зададим искомый сигнал (концепцию) $c \subset \{0,1\}^n$
- Рассмотрим гипотезу $h \subset \{0,1\}^n$ и введем ее ошибку:

$$error(h) = \sum_{x \in h \Delta c} D(x)$$

$$\Rightarrow$$

будем исследовать какие ошибки достижимы при разных формах c и h

Критика теоретического подхода

- Не рассматриваем ошибки в данных
- Проводится оценка наихудшего сценария, что редко встречается на практике

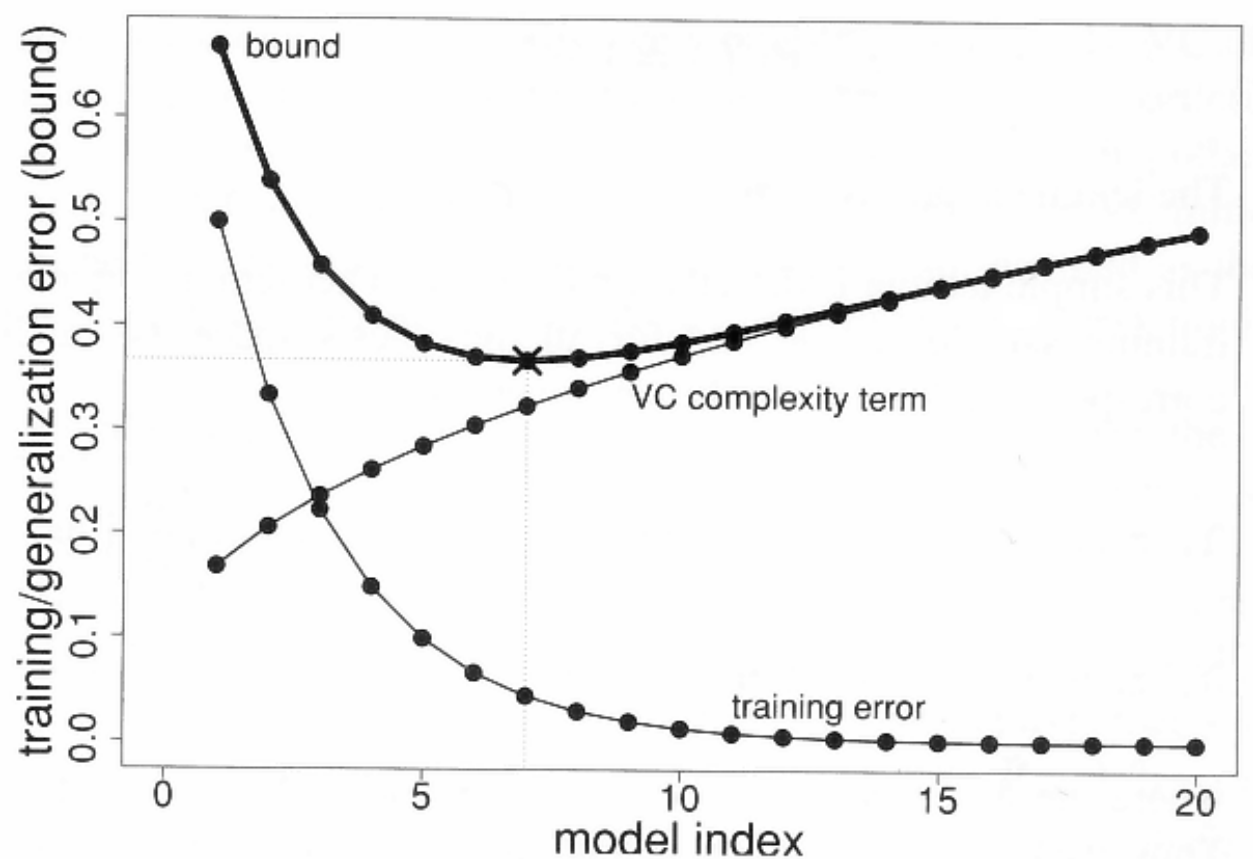
Оценка по методу Вапника-Червоненкиса I

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

$$T(T) \geq T(L) - \sqrt{\frac{dim_{VC}(H)(\log(\frac{2m}{dim_{VC}(H)}) + 1) - \log(\frac{\delta}{4})}{m}}$$

с вероятностью $1 - \delta$, $m = |L|$.

Оценка по методу Вапника-Червоненкиса II



картинка с www.svms.org

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.

Пару слов про KL-divergence

$$
\begin{aligned}
\log(p(y)) &= \log\left(\frac{p(y,\theta)}{p(\theta|y)}\right) \\
&= \int q(\theta) \log \frac{p(y,\theta)}{p(\theta|y)} d\theta \\
&= \int q(\theta) \log \frac{p(y,\theta)}{p(\theta|y)} \frac{q(\theta)}{q(\theta)} d\theta \\
&= \int q(\theta) \left(\log \frac{q(\theta)}{p(\theta|y)} + \log \frac{p(y,\theta)}{q(\theta)} \right) d\theta \\
&= \left(\int q(\theta) log \frac{q(\theta)}{p(\theta|y)} d\theta \right) + \left(\int q(\theta) log \frac{p(y,\theta)}{q(\theta)} d\theta \right)
\end{aligned}
$$

Красная часть называется Kullback–Leibler divergence ($KL(p\|q) = D_{KL}(p\|q)$). Синяя—ELBO(Evidence lower bound).

PAC обучаемость

Таки Probably Approximately Correct :)

Будем загадывать c не просто так, а в каком-то классе C, и отгадывать h в классе H

Definition

Если мы можем найти гипотезу в классе H, которая дает ошибку не более ϵ с вероятностью не более $1 - \delta$ за полиномиальное время, то назовем класс C PAC обучаемым над классом H

Пара интересных результатов

Если есть консистентный алгоритм, который обучает гипотезы из H по сигналу из C, то для достижения ошибки ϵ с вероятностью $1 - \delta$, то ему понадобится не более:

$$\frac{1}{\epsilon(1-\sqrt{\epsilon})}\left(2dim_{VC}(H)\log\frac{6}{\epsilon}+\log\frac{2}{\delta}\right)$$

точек. Или если $|H| < \infty$, то:

$$\frac{1}{\epsilon}\left(\log|H|+\log\frac{1}{\delta}\right)$$

PAC-Bayes bounds I

Если подходить к вопросу не просто нахождением $\hat{h}$, а искать распределение над H, то к этому вопросу можно подходить последовательно (PAC-Bayes):

Definition (Оценка McAllester'a (1998))

Введем априорное распределение над семейством решающих функций h. Тогда:

$$T(\hat{h}, T) \geq T(\hat{h}, L) - \sqrt{\frac{log\frac{1}{p(\hat{h})} + log(\frac{1}{\delta})}{2m}}$$

с вероятностью не меньше $1 - \delta$.

PAC-Bayes bounds II

Рассмотрим решение, как взвешивание между разными функциями семейства F.[1] Тогда получится еще краше (современный вид):

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

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

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

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