СТАТИСТИЧЕСКАЯ ТЕОРИЯ ПРИНЯТИЯ РЕШЕНИЙ

Сергей Николенко

СПбГУ — Санкт-Петербург
29 сентября 2018 г.

Random facts:

- 29 сентября 1897 г. на премьере спектакля «Креольский суд» в театре «Олимпо» (Буэнос-Айрес) было впервые исполнено танго
- 29 сентября 1907 г. в Петербурге было торжественно открыто регулярное трамвайное движение
- 29 сентября 1922 г. из Петрограда отплыл «Обербургомистр Хакен», знаменитый «философский пароход»
- 29 сентября 1963 г. атомная подводная лодка «К-181» всплыла в полынье на Северном полюсе
- 29 сентября 1984 г. близ посёлка Куанда встретились и уложили последнее, «золотое» звено строители БАМа

Статистическая теория принятия решений

Функция потери

- Сейчас мы попытаемся понять, что же на самом деле происходит в этих методах.
- Начнём с обычной регрессии – непрерывный вещественный вход $x \in \mathbb{R}^p$, непрерывный вещественный выход $y \in \mathbb{R}$; у них есть некоторое совместное распределение $p(x, y)$.
- Мы хотим найти функцию $f(x)$, которая лучше всего предсказывает y.

Функция потери

- Введём *функцию потери* (loss function) $L(y, f(x))$, которая наказывает за ошибки; естественно взять квадратичную функцию потери

$$L\left(y, f(x)\right) = (y - f(x))^2.$$

- Тогда каждому f можно сопоставить *ожидаемую ошибку предсказания* (expected prediction error):

$$\mathrm{EPE}(f) = \mathrm{E}(y - f(x))^2 = \int\int (y - f(x))^2 p(x, y) dx dy.$$

- И теперь самая хорошая функция предсказания $\hat{f}$ – это та, которая минимизирует $\mathrm{EPE}(f)$.

Функция потери

- Это можно переписать как

$$\mathrm{EPE}(f) = \mathrm{E}_x \mathrm{E}_{y|x} \left[(y - f(x))^2 \mid x \right],$$

и, значит, можно теперь минимизировать EPE поточечно:

$$\hat{f}(x) = \arg\min_c \mathrm{E}_{y|x'} \left[(y - c)^2 \mid x' = x \right],$$

а это можно решить и получить

$$\hat{f}(x) = \mathrm{E}_{y|x'}(y \mid x' = x).$$

- Это решение называется *функцией регрессии* и является наилучшим предсказанием y в любой точке x.

k-NN

- Теперь мы можем понять, что такое k-NN.
- Давайте оценим это ожидание:

$$f(x) = \mathrm{E}_{y|x'}(y \mid x' = x).$$

- Оценка ожидания – это среднее всех y с данным x. Конечно, у нас таких нету, поэтому мы приближаем это среднее как

$$\hat{f}(x) = \mathrm{Average}\left[y_i \mid x_i \in N_k(x)\right].$$

- Это сразу два приближения: ожидание через среднее и среднее в точке через среднее в ближних точках.
- Иначе говоря, k-NN предполагает, что в окрестности x функция $y(x)$ не сильно меняется, а лучше всего – она кусочно-постоянна.

Линейная регрессия

- А линейная регрессия – это модельный подход, мы предполагаем, что функция регрессии линейна от своих аргументов:
$$f(x) \approx x^\top w.$$
- Теперь мы не берём условие по x, как в k-NN, а просто собираем много значений для разных x и обучаем модель.

Классификация

- То же самое можно и с задачей классификации сделать. Пусть у нас переменная g с K возможными значениями $g_1, \dots, g_k$ предсказывается.
- Введём функцию потери, равную 1 за каждый неверный ответ. Получим
$$\mathrm{EPE} = \mathrm{E}\left[L(g, \hat{g}(x))\right].$$
- Перепишем как раньше:
$$\mathrm{EPE} = \mathrm{E}_x \sum_{k=1}^{K} L(g_k, \hat{g}(x)) p(g_k \mid x).$$
- Опять достаточно оптимизировать поточечно:
$$\hat{g}(x) = \arg\min_g \sum_{k=1}^{K} L(g_k, \hat{g}(x)) p(g_k \mid x).$$

Классификация

- Опять достаточно оптимизировать поточечно:

$$\hat{g}(x) = \arg\min_g \sum_{k=1}^{K} L(g_k, \hat{g}(x)) p(g_k \mid x).$$

- Для 0-1 функции потери это упрощается до

$$\hat{g}(x) = \arg\min_g \left[1 - p(g \mid x)\right], \text{ т.е.}$$

$$\hat{g}(x) = g_k, \text{ если } p(g_k \mid x) = \max_g p(g \mid x).$$

- Это называется *оптимальным байесовским классификатором*; если модель известна, то его обычно можно построить.

Bias-variance decomposition

- Рассмотрим совместное распределение $p(y, x)$ и квадратичную функцию потерь $L(y, f(x)) = (y - f(x))^2$.
- Мы знаем, что тогда оптимальная оценка – это функция регрессии

$$\hat{f}(x) = \mathrm{E}\left[y \mid x\right] = \int y p(y \mid x) dx.$$

Bias-variance decomposition

- Давайте подсчитаем ожидаемую ошибку и перепишем её в другой форме:

$$\mathrm{E}[L] = \mathrm{E}[(y - f(x))^2] = \mathrm{E}[(y - \mathrm{E}\left[y \mid x\right] + \mathrm{E}\left[y \mid x\right] - f(x))^2] =$$
$$= \int \left(f(x) - \mathrm{E}\left[y \mid x\right]\right)^2 p(x)dx + \int \left(\mathrm{E}\left[y \mid x\right] - y\right)^2 p(x,y)dxdy,$$

потому что

$$\int \left(f(x) - \mathrm{E}\left[y \mid x\right]\right)\left(\mathrm{E}\left[y \mid x\right] - y\right) p(x,y)dxdy = 0.$$

Bias-variance decomposition

- Эта форма записи – разложение на bias-variance и noise:

$$\mathrm{E}[L] = \int \left(f(x) - \mathrm{E}\left[y \mid x\right]\right)^2 p(x)dx + \int \left(\mathrm{E}\left[y \mid x\right] - y\right)^2 p(x,y)dxdy,$$

- Отсюда, кстати, тоже сразу видно, что от $f(x)$ зависит только первый член, и он минимизируется, когда

$$f(x) = \hat{f}(x) = \mathrm{E}\left[y \mid x\right].$$

- А noise, $\int \left(\mathrm{E}\left[y \mid x\right] - y\right)^2 p(x,y)dxdy$, – это просто свойство данных, дисперсия шума.

Bias-variance decomposition

- Если бы у нас был всемогущий компьютер и неограниченный датасет, мы бы, конечно, на этом и закончили, посчитали бы $\hat{f}(x) = \mathrm{E}\left[y \mid x\right]$, и всё.
- Однако жизнь – борьба, и у нас есть только ограниченный датасет из N точек. Предположим, что этот датасет берётся по распределению $p(x, y)$ – т.е. фактически рассмотрим много-много экспериментов такого вида:
  - взяли датасет D из N точек по распределению $p(x, y)$;
  - подсчитали нашу чудо-регрессию;
  - получили новую функцию предсказания $f(x; D)$.
- Разные датасеты будут приводить к разным функциям предсказания…

Bias-variance decomposition

- ...а потому давайте усредним теперь по датасетам.
- Наш первый член в ожидаемой ошибке выглядел как $\left(f(x) - \hat{f}(x)\right)^2$, а теперь будет $\left(f(x; D) - \hat{f}(x)\right)^2$, и его можно усреднить по D, применив такой же трюк:

$$\left(f(x; D) - \hat{f}(x)\right)^2$$
$$= \left(f(x; D) - \mathrm{E}_D\left[f(x; D)\right] + \mathrm{E}_D\left[f(x; D)\right] - \hat{f}(x)\right)^2$$
$$= \left(f(x; D) - \mathrm{E}_D\left[f(x; D)\right]\right)^2 + \left(\mathrm{E}_D\left[f(x; D)\right] - \hat{f}(x)\right)^2 + 2(\ldots)(\ldots),$$

и в ожидании получится...

Bias-variance decomposition

- ...и в ожидании получится

$$\mathrm{E}_D\left[\left(f(x;D)-\hat{f}(x)\right)^2\right]=$$
$$=\mathrm{E}_D\left[\left(f(x;D)-\mathrm{E}_D\left[f(x;D)\right]\right)^2\right]+\left(\mathrm{E}_D\left[f(x;D)\right]-\hat{f}(x)\right)^2.$$

- Разложили на дисперсию $\mathrm{E}_D\left[\left(f(x;D)-\mathrm{E}_D\left[f(x;D)\right]\right)^2\right]$ и квадрат систематической ошибки $\left(\mathrm{E}_D\left[f(x;D)\right]-\hat{f}(x)\right)^2$; это и есть bias-variance decomposition.

Bias-variance-noise

$$\text{Expected loss} = (\text{bias})^2 + \text{variance} + \text{noise},$$

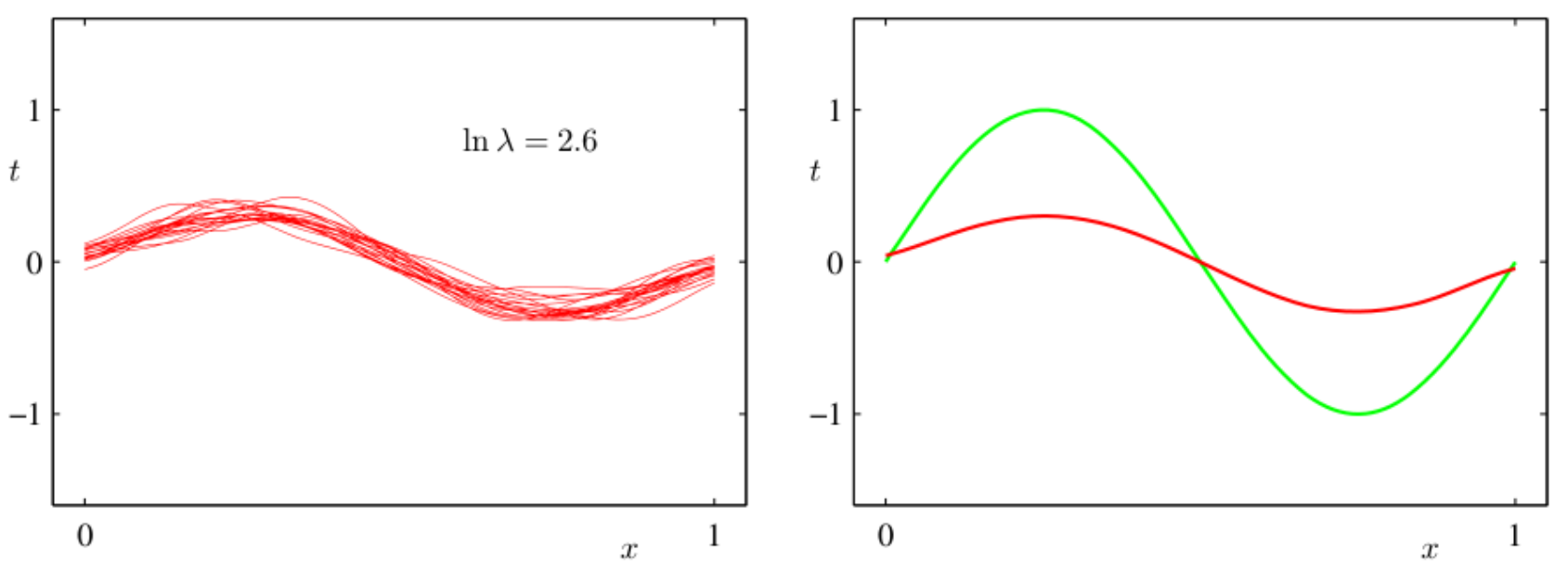
где

$$(\text{bias})^2 = \left(\mathrm{E}_D\left[f(x;D)\right] - \hat{f}(x)\right)^2,$$
$$\text{variance} = \mathrm{E}_D\left[\left(f(x;D) - \mathrm{E}_D\left[f(x;D)\right]\right)^2\right],$$
$$\text{noise} = \int \left(\mathrm{E}\left[y \mid x\right] - y\right)^2 p(x,y)dxdy.$$

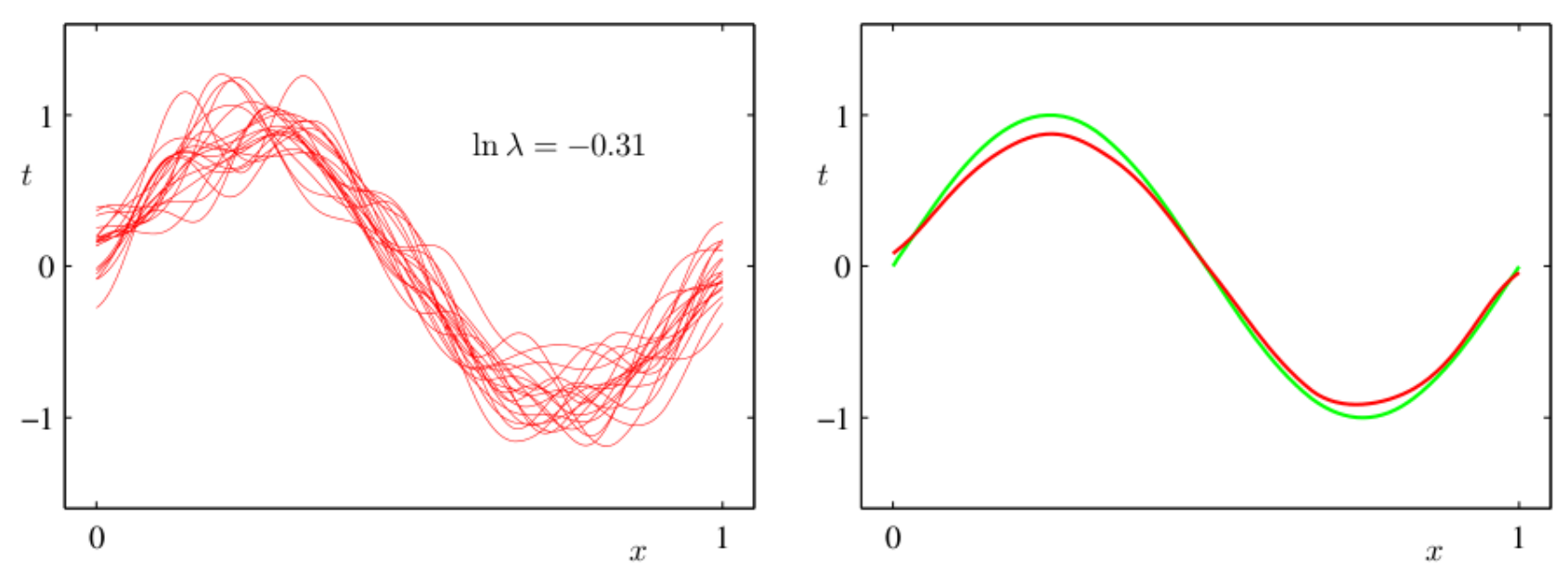
Пример

- Теперь давайте посмотрим на пример: опять та же синусоида, опять приближаем её линейной регрессией с полиномиальными признаками (максимальным их числом).
- И мы регуляризуем эту регрессию с параметром α.
- Будем набрасывать много датасетов и смотреть, что меняется при этом.

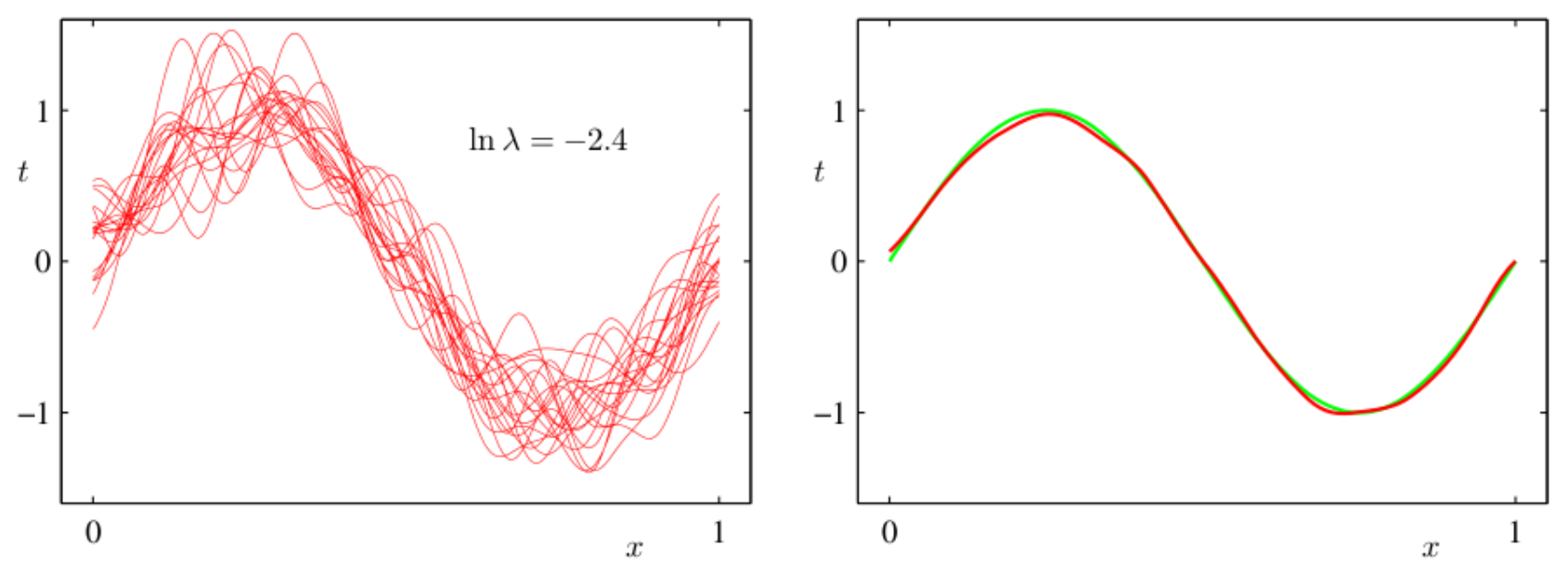
Регуляризатор и bias-variance



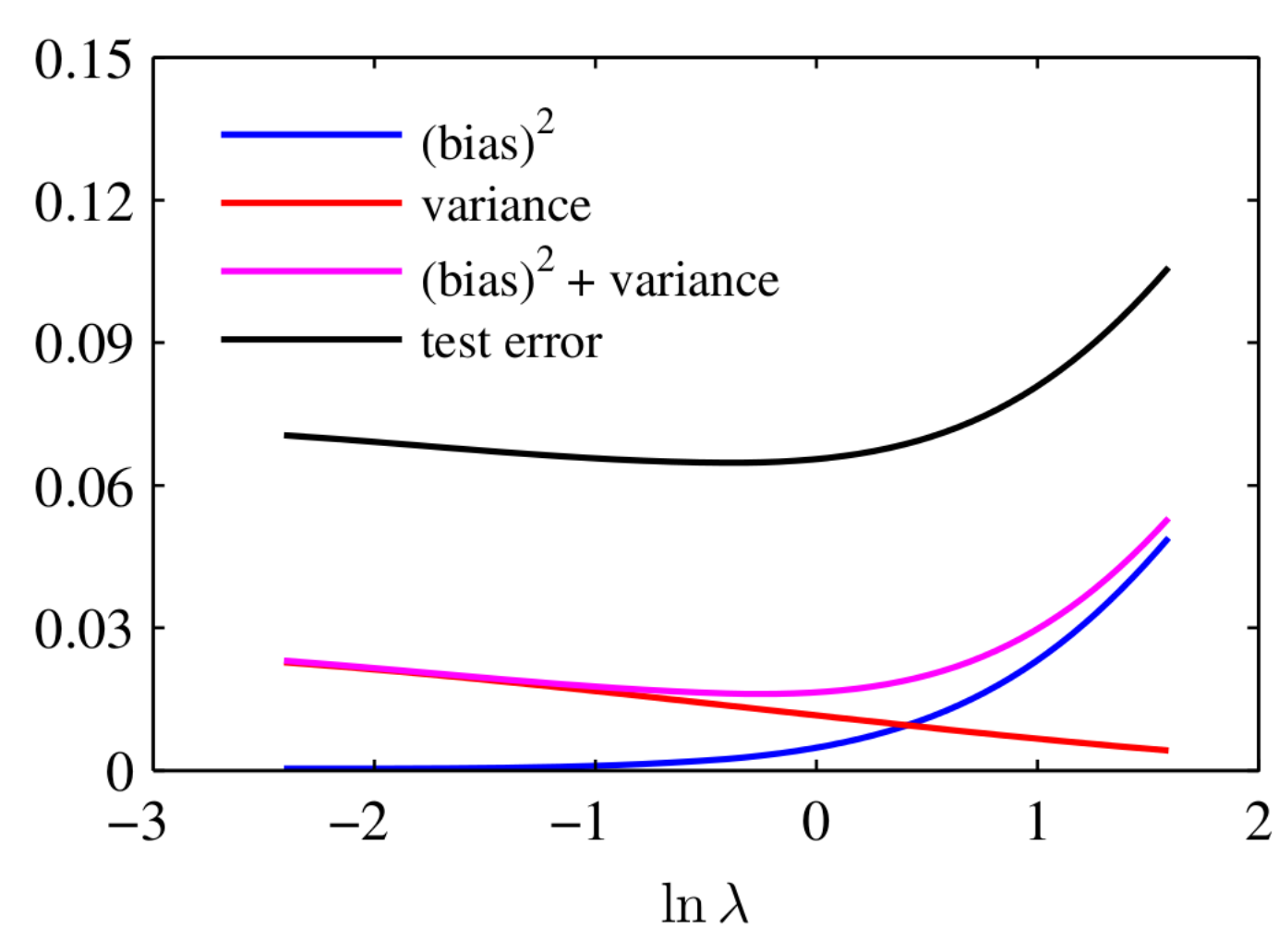
Регуляризатор и bias-variance



Регуляризатор и bias-variance



Регуляризатор и bias-variance



Эквивалентное ядро и сравнение моделей

Эквивалентное ядро

- Вспомним наши байесовские предсказания:

$$p(t \mid t, \alpha, \beta) = N(t \mid \mu_N^\top \phi(x), \sigma_N^2),$$
$$\text{где } \sigma_N^2 = \frac{1}{\beta} + \phi(x)^\top \Sigma_N \phi(x).$$

- Давайте перепишем среднее апостериорного распределения в другой форме (вспомним, что $\mu_N = \beta \Sigma_N \Phi^\top t$):

$$y(x, \mu_N) = \mu_N^\top \phi(x) = \beta \phi(x)^\top \Sigma_N \Phi^\top t =$$
$$= \sum_{n=1}^{N} \beta \phi(x)^\top \Sigma_N \phi(x_n) t_n.$$

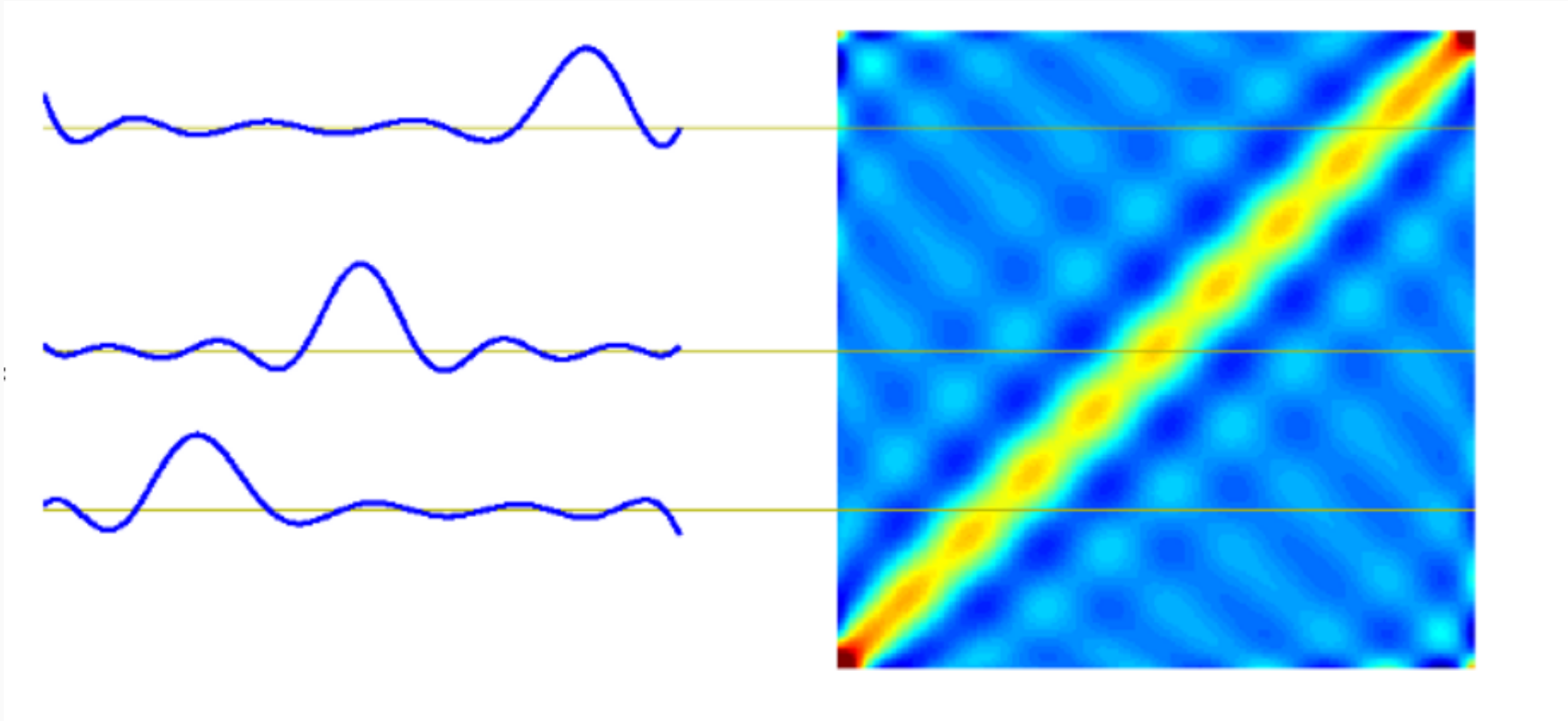
Эквивалентное ядро

- $y(x, \mu_N) = \sum_{n=1}^{N} \beta\phi(x)^\top \Sigma_N \phi(x_n) t_n$.
- Это значит, что предсказание можно переписать как

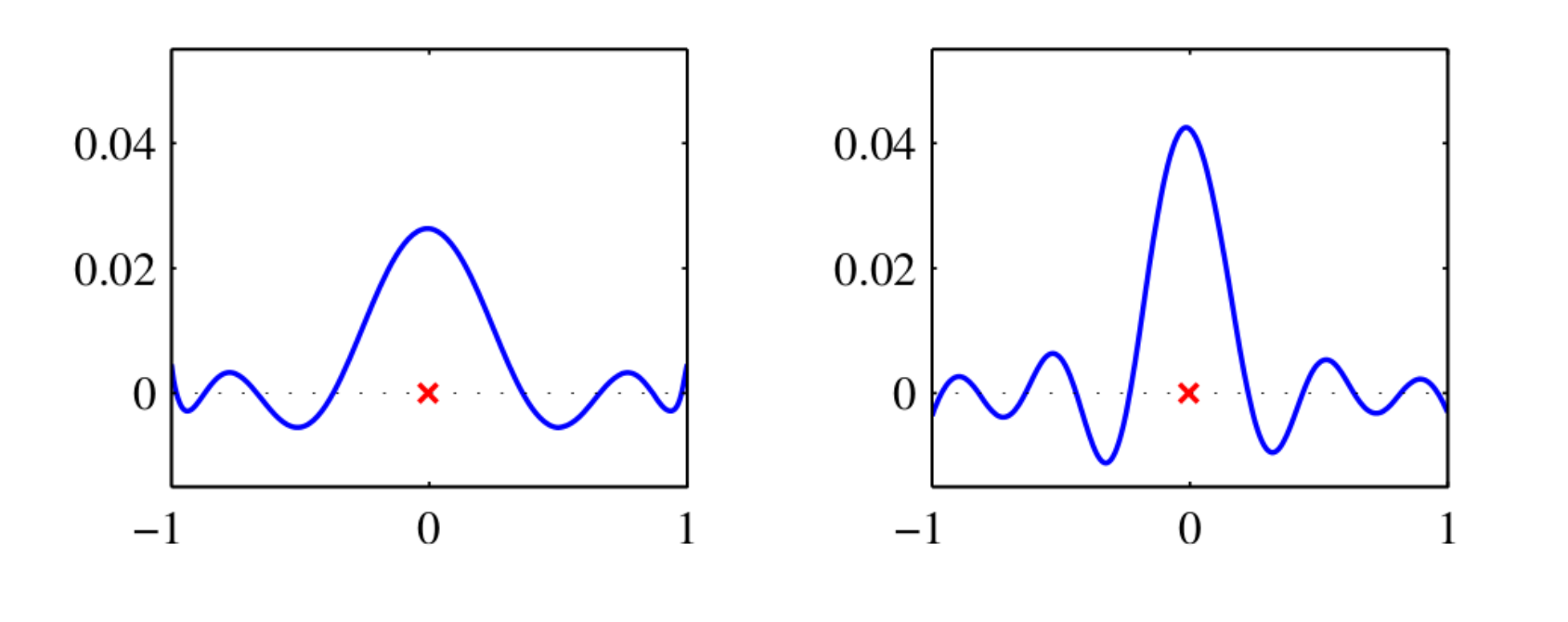
$$y(x, \mu_N) = \sum_{n=1}^{N} k(x, x_n) t_n.$$

- Т.е. мы предсказываем следующую точку как линейную комбинацию значений в известных точках.
- Функция $k(x, x') = \beta\phi(x)^\top \Sigma_N \phi(x')$ называется *эквивалентным ядром* (equivalent kernel).

Эквивалентное ядро



Эквивалентное ядро



Выводы про эквивалентное ядро

- Эквивалентное ядро $k(x, x')$ локализовано вокруг x как функция x', т.е. каждая точка оказывает наибольшее влияние около себя и затухает потом.
- Можно было бы с самого начала просто определить ядро и предсказывать через него, безо всяких базисных функций ϕ – такой подход мы ещё будем рассматривать.

Упражнение. Докажите, что $\sum_{n=1}^{N} k(x, x_n) = 1$.

Байесовское сравнение моделей

- Мы говорили о том, что при увеличении числа параметров модели возникает оверфиттинг.
- Как этого избежать? Как сравнить модели с разным числом параметров?
- Теория байесовского вывода предлагает такой выход: давайте будем не точечные оценки параметров модели рассматривать, а тоже интегрировать по параметрам модели.

Байесовское сравнение моделей

- Пусть мы хотим сравнить модели из множества $\{M_i\}_{i=1}^{L}$.
- Модель – это распределение вероятностей над данными D.
- По тестовому набору D можно оценить апостериорное распределение

$$p(M_i \mid D) \propto p(M_i)p(D \mid M_i).$$

Байесовское сравнение моделей

- Если знать апостериорное распределение, то можно сделать предсказание:

$$p(t \mid x, D) = \sum_{i=1}^{L} p(t \mid x, M_i, D) p(M_i \mid D).$$

- *Model selection* (выбор модели) – это когда мы приближаем предсказание, выбирая просто самую (апостериорно) вероятную модель.

Байесовское сравнение моделей

- Если модель определена параметрически, через w, то

$$p(D \mid M_i) = \int p(D \mid w, M_i) p(w \mid M_i) dw.$$

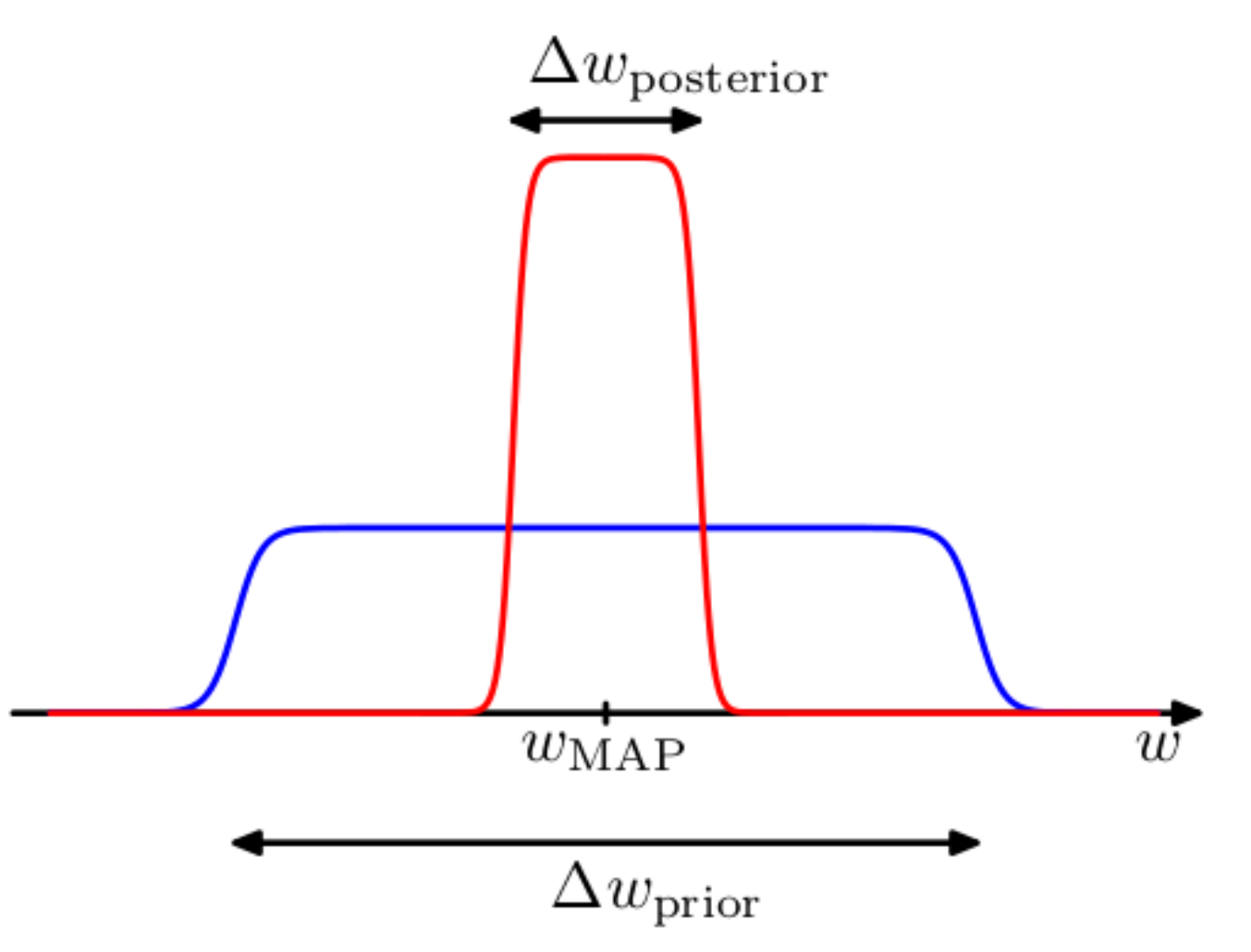
- Т.е. это вероятность сгенерировать D, если выбирать параметры модели по её априорному распределению, а потом накидывать данные.
- Это, кстати, в точности знаменатель из теоремы Байеса:

$$p(w \mid M_i, D) = \frac{p(D \mid w, M_i) p(w \mid M_i)}{p(D \mid M_i)}.$$

Байесовское сравнение моделей

- Предположим, что у модели один параметр w, а апостериорное распределение – это острый пик вокруг w_{MAP} шириной $\Delta w_{\mathrm{posterior}}$.
- Тогда можно приблизить $p(D) = \int p(D \mid w)p(w)dw$ как значение в максимуме, умноженное на ширину.
- Предположим ещё, что априорное распределение тоже плоское, $p(w) = \frac{1}{\Delta w_{\mathrm{prior}}}$.

Приближение $p(D)$



Приближение $p(D)$

- Тогда получится

$$p(D) = \int p(D \mid w) p(w) dw \approx p\left(D \mid w_{\mathrm{MAP}}\right) \frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}},$$

$$\ln p(D) \approx \ln p\left(D \mid w_{\mathrm{MAP}}\right) + \ln\left(\frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}}\right).$$

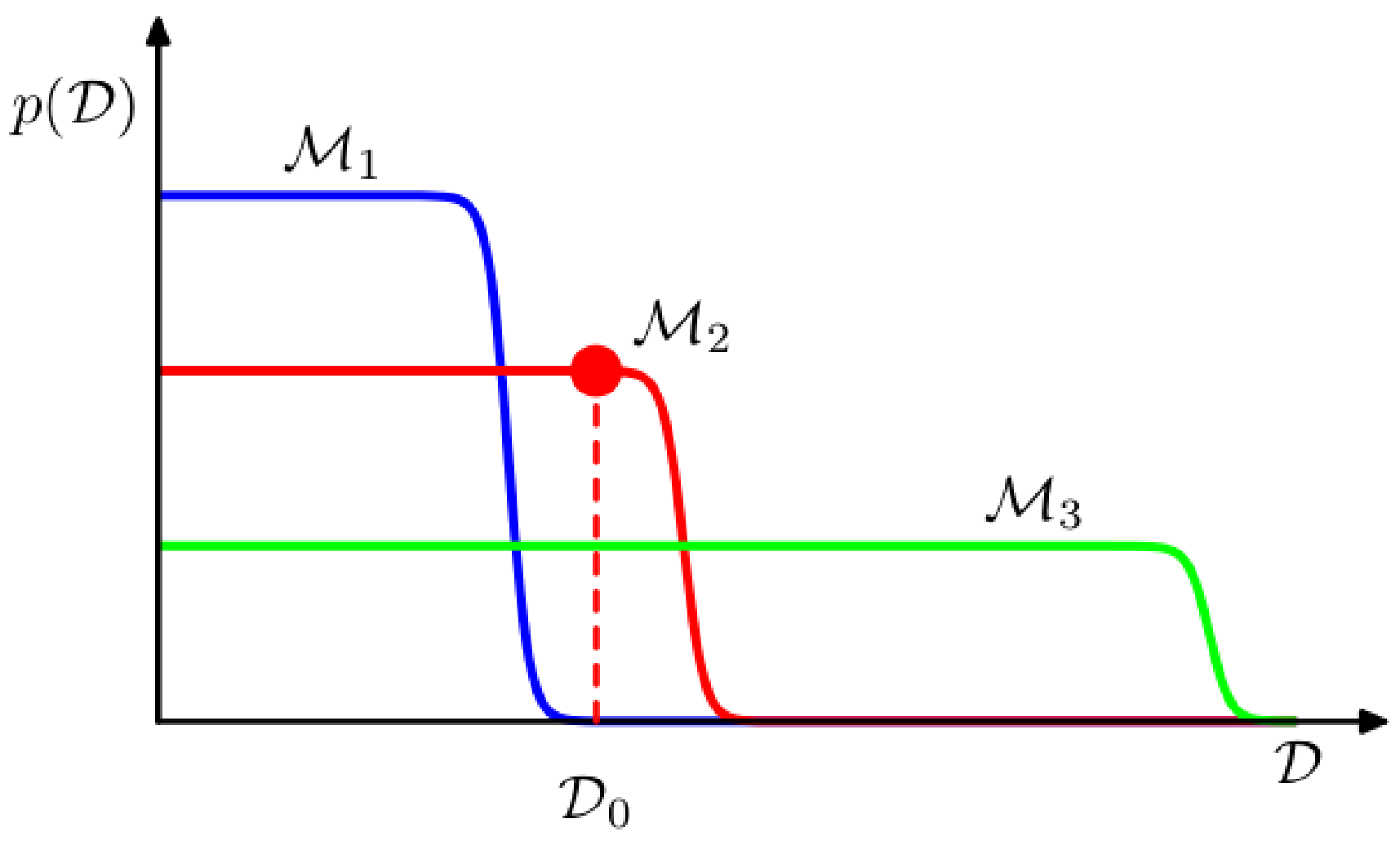
- Это значит, что мы добавляем штраф за «слишком узкое» апостериорное распределение – то есть в точности штраф за оверфиттинг!
- Для модели из M параметров, если предположить, что у них одинаковые $\Delta w_{\mathrm{posterior}}$, получим

$$\ln p(D) \approx \ln p\left(D \mid w_{\mathrm{MAP}}\right) + M \ln\left(\frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}}\right).$$

Другой взгляд

- Другими словами: давайте посмотрим, какие датасеты может генерировать та или иная модель.
- Простая модель (e.g., линейная) генерирует похожие датасеты, «мало» разных датасетов, у неё высокая $p(D \mid M)$.
- Сложная модель (e.g., многочлен девятой степени) генерирует «много» разных датасетов, у неё низкая $p(D \mid M)$.
- Но сложная может хорошо выразить датасеты, которые не может выразить простая; поэтому в сумме надо выбирать «среднюю».

Приближение $p(D)$



Правильный ответ лучше

- Sanity check: тут какие-то штрафы мы навводили; будет ли истинный правильный ответ $p(D \mid M_{\text{true}})$ всегда оптимальным в этом смысле?
- Конечно, для конкретного датасета может так повезти, что не будет.
- Но если усреднить по всем датасетам, выбранным по $p(D \mid M_{\text{true}})$...

Правильный ответ лучше

- ...то получится

$$\mathrm{E}\left[\ln \frac{p(D \mid M_{\text{true}})}{p(D \mid M)}\right] = \int p(D \mid M_{\text{true}}) \ln \frac{p(D \mid M_{\text{true}})}{p(D \mid M)} dD.$$

- Это называется *расстоянием Кульбака-Лейблера* (Kullback-Leibler divergence) между распределениями $p(D \mid M_{\text{true}})$ и $p(D \mid M)$.

Упражнение. Докажите, что расстояние Кульбака-Лейблера всегда неотрицательно, т.е. что $p(D \mid M_{\text{true}}) \geq p(D \mid M)$ для любой M.

Thank you!

Thank you for your attention!