Статистическая теория принятий решений

Сергей Николенко

Академия MADE — Mail.Ru

24 апреля 2020 г.

Random facts:

- 24 апреля — Международный день солидарности молодёжи, приуроченный к заключительному заседанию Бандунгской конференции стран Азии и Африки в 1955 году, а также Всемирный день защиты лабораторных животных, приуроченный к дню рождения бывшего президента Национального анививисекционного общества (NAVS) Хью Касвелл Трименхир Даудинга, 1-го барона Даудинга
- 24 апреля 1803 г. указом императора Александра I было утверждено положение о Кавказских Минеральных Водах, а 24 апреля 1833 г. в США была запатентована газированная вода
- 24 апреля 1846 г. началась война США с Мексикой, 24 апреля 1898 г. США объявила войну Испания, а 24 апреля 1918 г. американские интервенты высадились в Мурманске
- 24 апреля 1985 г. Верховный суд Канады признал законной работу магазинов по воскресеньям

Предсказания
в линейной регрессии

Предсказание в линейной регрессии

- Теперь давайте вернёмся к байесовской постановке:
  1. найти апостериорное распределение на гипотезах/параметрах:

$$p(\theta \mid D) \propto p(D|\theta)p(\theta)$$

     (и/или найти максимальную апостериорную гипотезу $\arg\max_\theta p(\theta \mid D)$);
  2. найти апостериорное распределение исходов дальнейших экспериментов:

$$p(x \mid D) \propto \int_{\theta \in \Theta} p(x \mid \theta)p(D|\theta)p(\theta)\mathrm{d}\theta.$$

Предсказание в линейной регрессии

- В прошлый раз мы нашли апостериорное распределение: для гауссовского априорного

$$p(\mathbf{w} \mid \alpha) = \mathcal{N}(\mathbf{w} \mid \mathbf{0}, \frac{1}{\alpha} I)$$

мы нашли

$$\begin{aligned} p(\mathbf{w} \mid \mathbf{t}, \alpha, \beta) &= \mathcal{N}(\mathbf{w} \mid \boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N), \\ \boldsymbol{\mu}_N &= \boldsymbol{\Sigma}_N \left(\boldsymbol{\Sigma}_0^{-1} \mu_0 + \beta \boldsymbol{\Phi}^\top \mathbf{t} \right), \\ \boldsymbol{\Sigma}_N &= \left(\boldsymbol{\Sigma}_0^{-1} + \beta \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \right)^{-1}, \end{aligned}$$

где $\beta = \frac{1}{\sigma^2}$ (precision нормального распределения).

Предсказание в линейной регрессии

- Теперь сделаем следующий шаг – найдём апостериорное распределение наших предсказаний

$$p(t \mid \mathbf{t}, \alpha, \beta) = \int p(t \mid \mathbf{w}, \beta) p(\mathbf{w} \mid \mathbf{t}, \alpha, \beta) d\mathbf{w}.$$

- Это свёртка двух гауссианов, и получается...

Предсказание в линейной регрессии

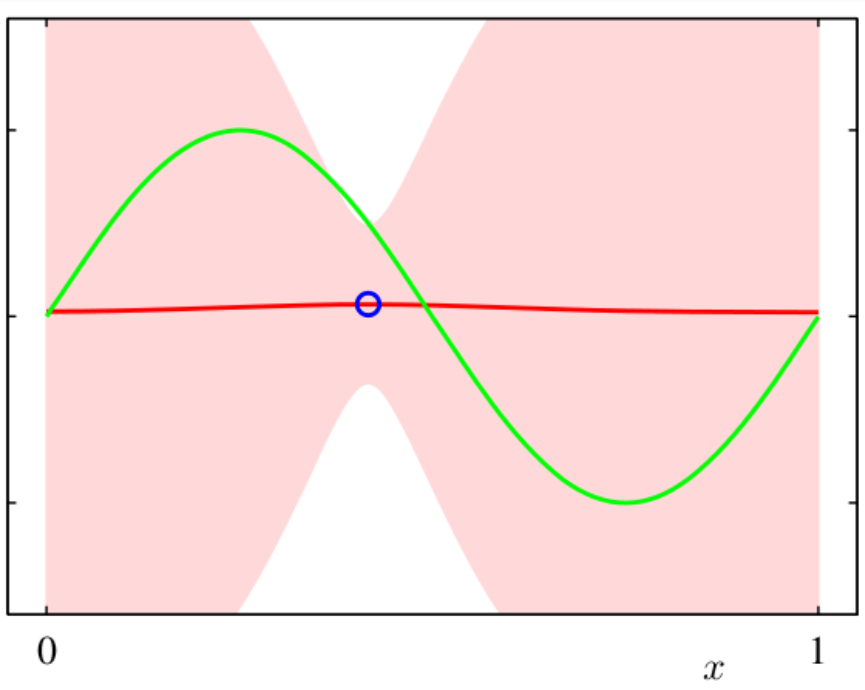
- ...тоже гауссиан:

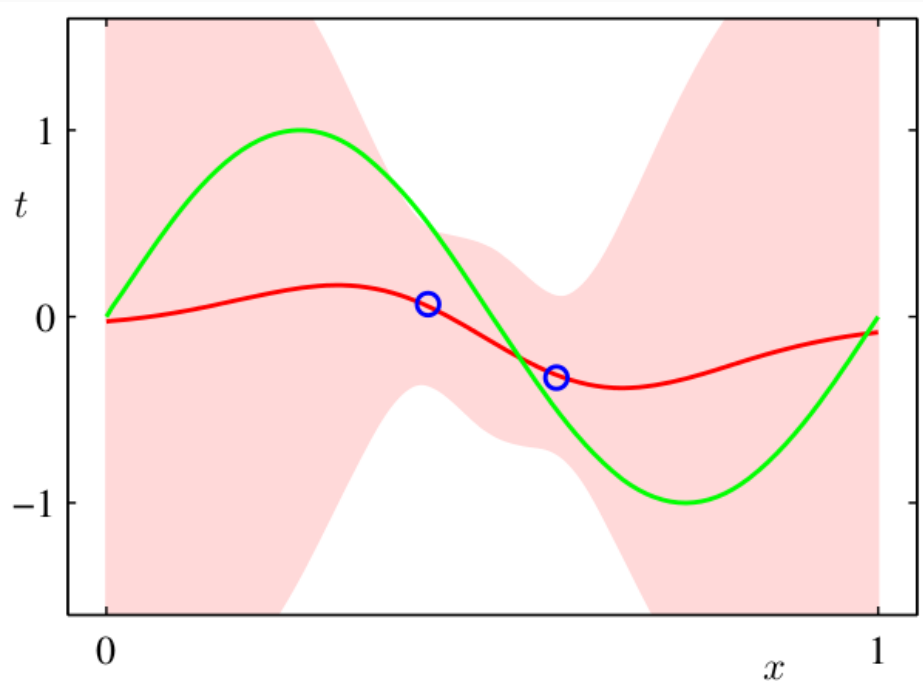
$$p(t \mid \mathbf{t}, \alpha, \beta) = \mathcal{N}(t \mid \boldsymbol{\mu}_N^\top \boldsymbol{\phi}(\mathbf{x}), \sigma_N^2),$$
$$\text{где } \sigma_N^2 = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma}_N \boldsymbol{\phi}(\mathbf{x}).$$

- Т.е. дисперсия складывается из шума в данных β и дисперсии параметров $\mathbf{w}$; гауссианы независимы, и их дисперсии просто складываются.

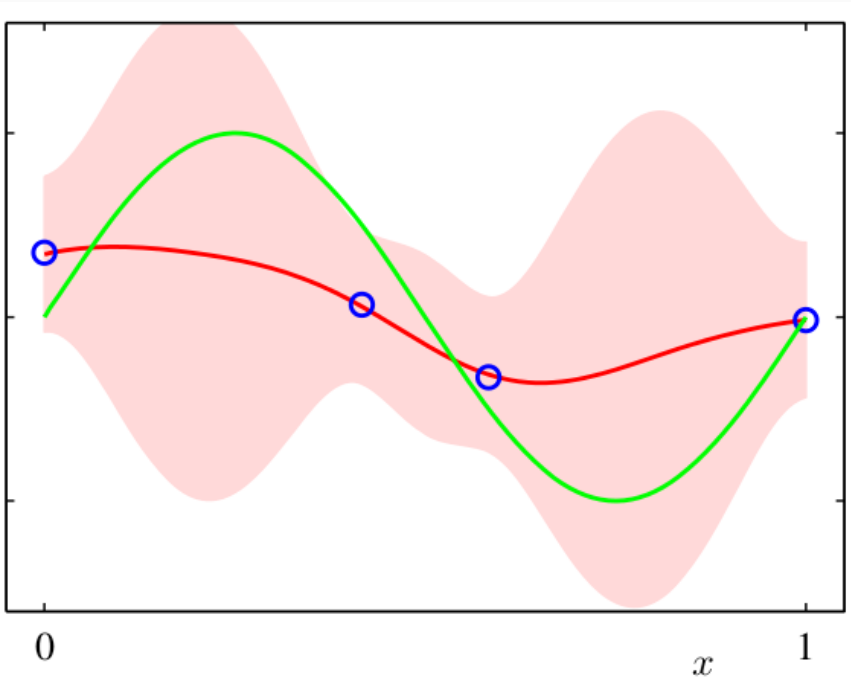
Упражнение. Оценка всё время уточняется: $\sigma_{N+1}^2 \leq \sigma_N^2$.

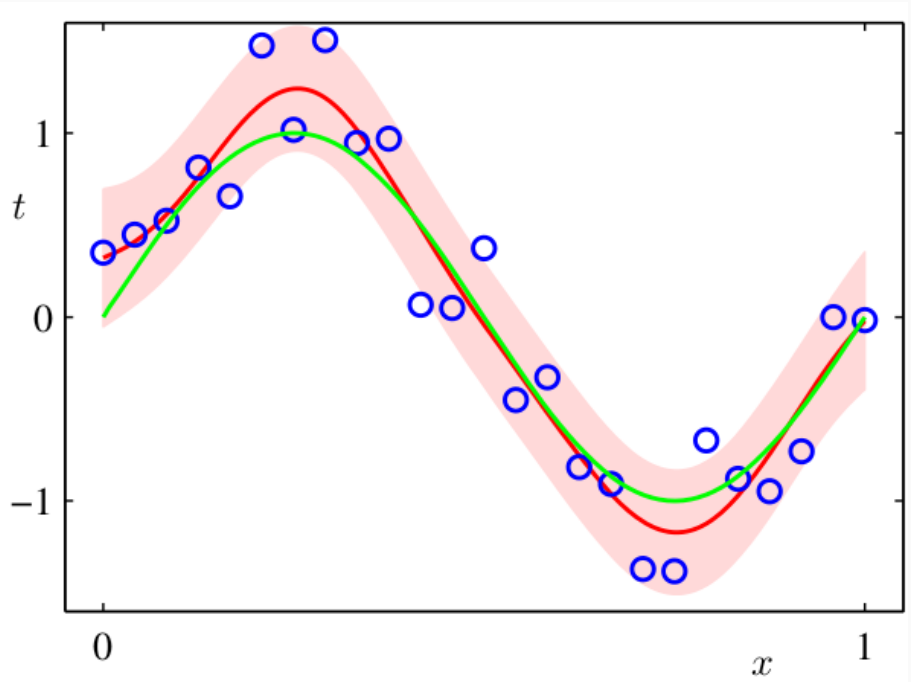
Предсказания



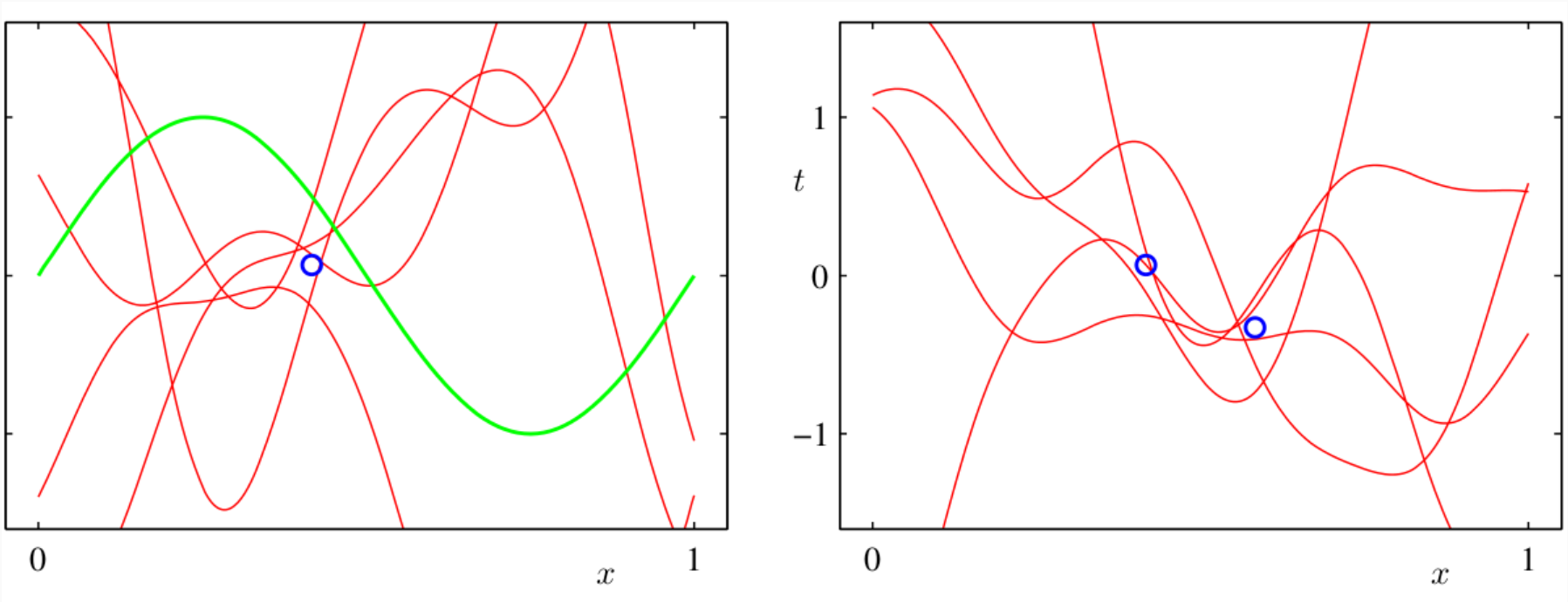


Предсказания

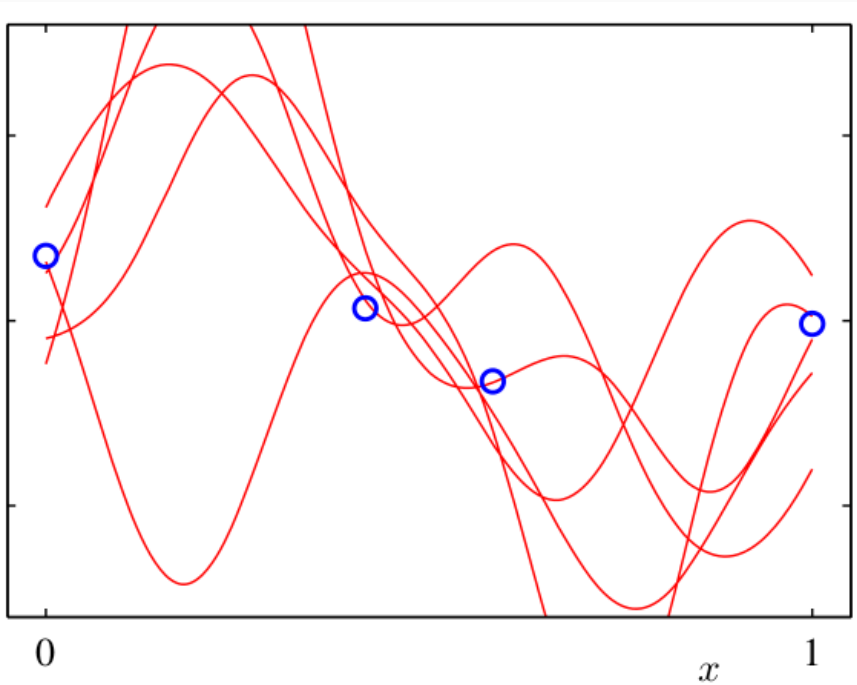


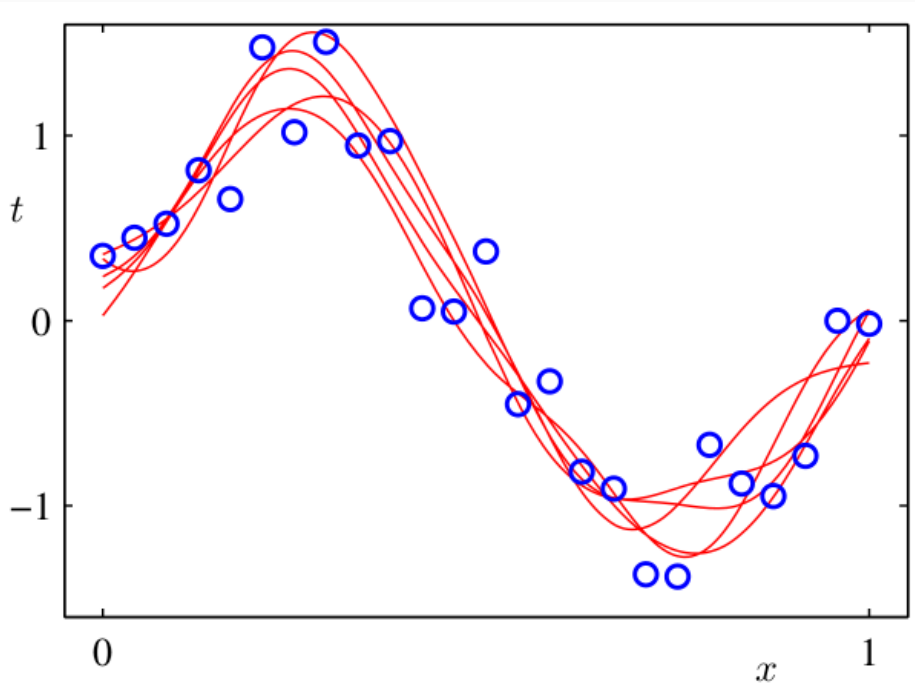


Предсказания



Предсказания





Эквивалентное ядро

Эквивалентное ядро

- Вспомним наши байесовские предсказания:

$$p(t \mid \mathbf{t}, \alpha, \beta) = \mathcal{N}(t \mid \boldsymbol{\mu}_N^\top \boldsymbol{\phi}(\mathbf{x}), \sigma_N^2),$$
$$\text{где } \sigma_N^2 = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma}_N \boldsymbol{\phi}(\mathbf{x}).$$

- Давайте перепишем среднее апостериорного распределения в другой форме (вспомним, что $\boldsymbol{\mu}_N = \beta \boldsymbol{\Sigma}_N \boldsymbol{\Phi}^\top \mathbf{t}$):

$$y(\mathbf{x}, \boldsymbol{\mu}_N) = \boldsymbol{\mu}_N^\top \boldsymbol{\phi}(\mathbf{x}) = \beta \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma}_N \boldsymbol{\Phi}^\top \mathbf{t} =$$
$$= \sum_{n=1}^{N} \beta \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma}_N \boldsymbol{\phi}(\mathbf{x}_n) t_n.$$

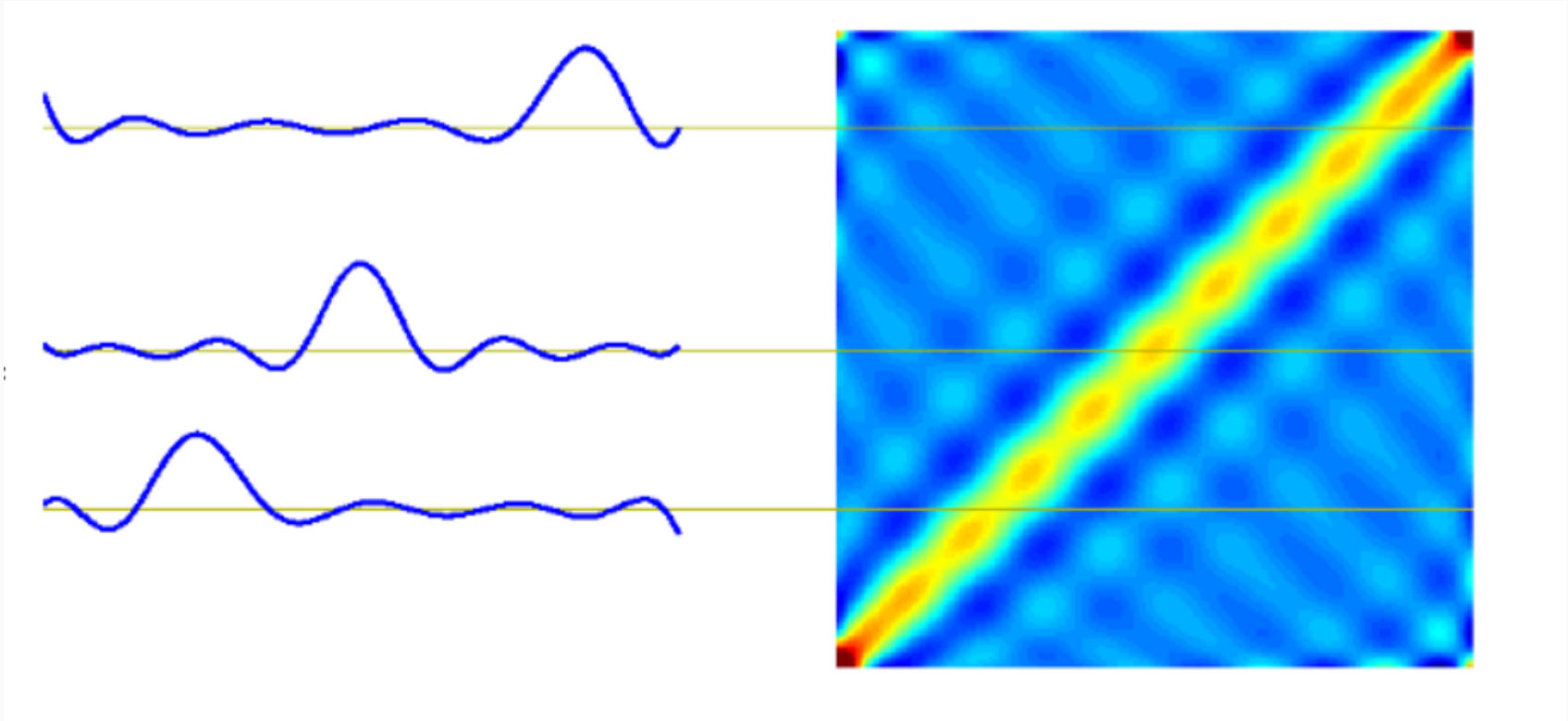
Эквивалентное ядро

- $y(\mathbf{x}, \boldsymbol{\mu}_N) = \sum_{n=1}^{N} \beta \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\Sigma}_N \boldsymbol{\phi}(\mathbf{x}_n) t_n$.
- Это значит, что предсказание можно переписать как

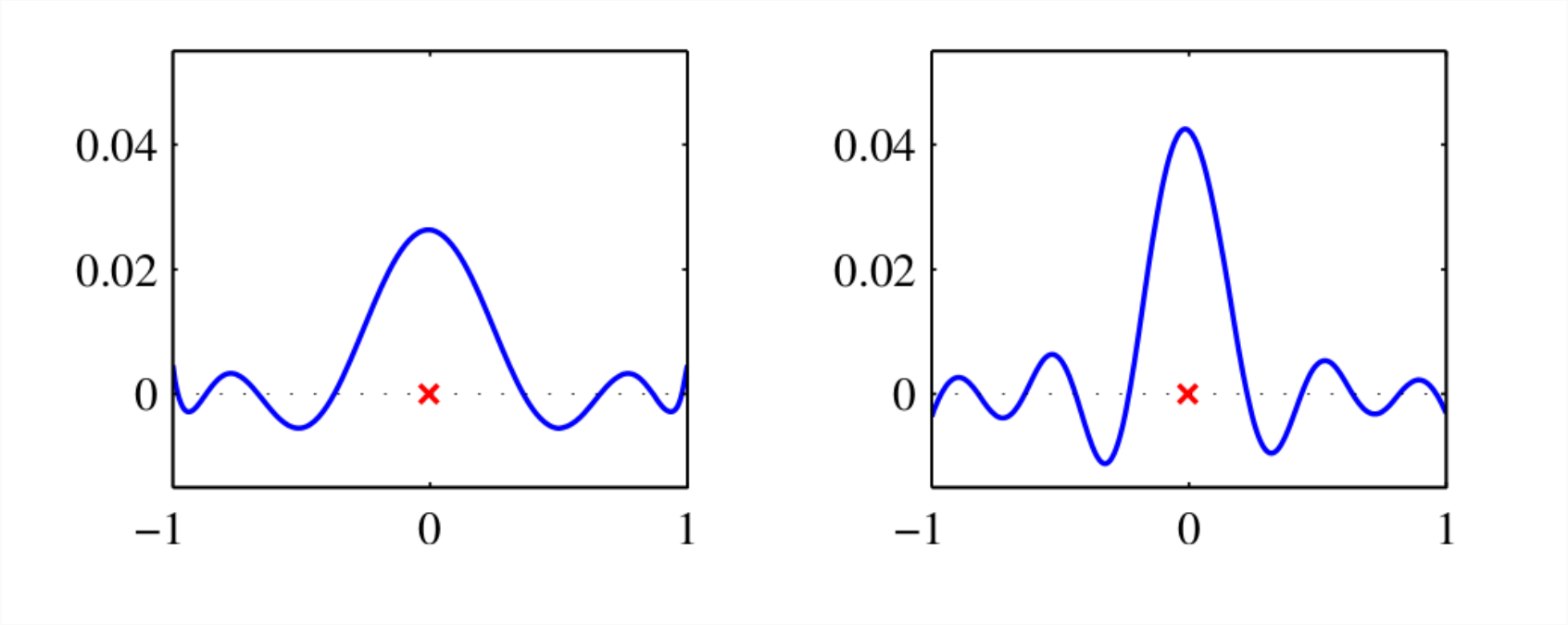
$$y(\mathbf{x}, \boldsymbol{\mu}_N) = \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) t_n.$$

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

Эквивалентное ядро



Эквивалентное ядро



Выводы про эквивалентное ядро

- Эквивалентное ядро $k(\mathbf{x}, \mathbf{x}')$ локализовано вокруг $\mathbf{x}$ как функция $\mathbf{x}'$, 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 $\sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) = 1$.

Эмпирический байес

Эмпирический байес

- Откуда берутся гиперпараметры?
- Оказывается, их тоже можно оптимизировать!
- У линейной регрессии, например, два гиперпараметра: $\beta = \frac{1}{\sigma^2}$ и α (точность регуляризатора, пусть гребневого).
- Давайте просто попробуем оптимизировать $p(D \mid \alpha, \beta)$ (marginal likelihood).

Эмпирический байес

- Получается:

$$p(D \mid \alpha, \beta) = \int p(\mathbf{w}) p(D \mid \mathbf{w}) \mathrm{d}\mathbf{w},$$

$$\ln p(D \mid \alpha, \beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \left(\frac{\alpha}{2\pi}\right)^{\frac{d}{2}} \int e^{-\frac{\beta}{2}\|\mathbf{y} - X\mathbf{w}\|^2 - \frac{\alpha}{2}\mathbf{w}^\top \mathbf{w}} \mathrm{d}\mathbf{w}.$$

- Выделяем полный квадрат так же, как раньше:

$$A = \beta X^\top X + \alpha \boldsymbol{I},$$

$$\mathbf{m}_N = \beta A^{-1} X^\top \mathbf{y}.$$

Эмпирический байес

- Теперь

$$\int e^{-\frac{1}{2}(\mathbf{w}-\mathbf{m}_N)^\top A(\mathbf{w}-\mathbf{m}_N)}\mathrm{d}\mathbf{w} = (2\pi)^{\frac{d}{2}}\sqrt{\det A^{-1}}.$$

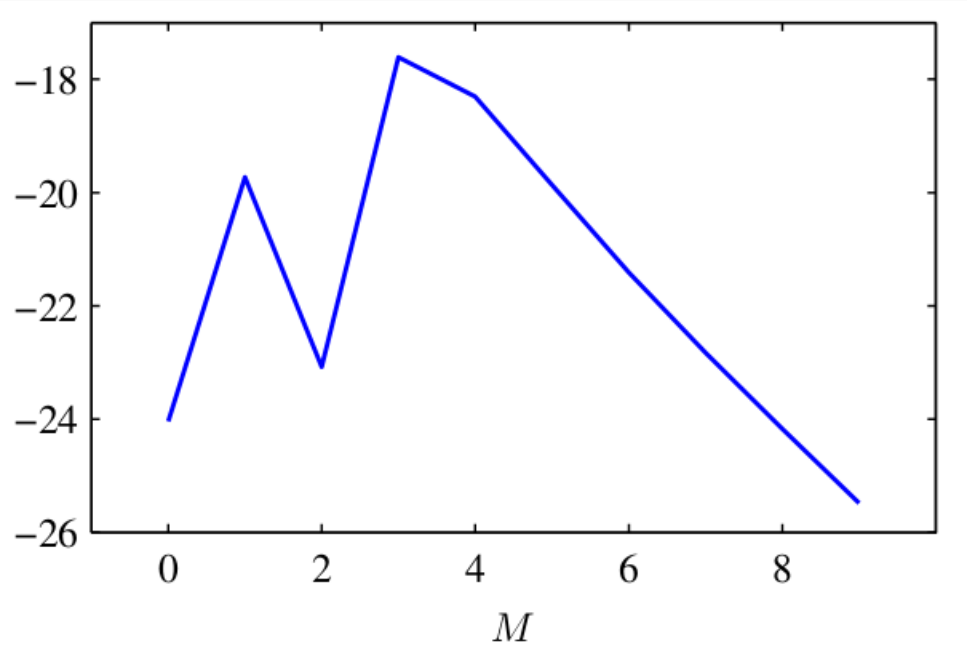
- Получается:

$$\ln p(D \mid \alpha, \beta) =$$
$$\frac{d}{2}\ln\alpha + \frac{N}{2}\ln\beta - \frac{\beta}{2}\|\mathbf{y}-X\mathbf{m}_N\|^2 - \frac{\alpha}{2}\mathbf{m}_N^\top\mathbf{m}_N - \frac{1}{2}\ln\det A - \frac{N}{2}\ln(2\pi).$$

- Это теперь надо максимизировать по α и β, а можно и разные d перебирать, если речь идёт о том, как выбрать оптимальное число признаков.

Эмпирический байес

- Пример графика по числу параметров:



- О том, как оптимизировать, поговорим позже.

Проклятие размерности

Параметрические и непараметрические модели

- Последнее замечание: модели бывают параметрические и непараметрические.
- Мы в основном будем заниматься моделями с фиксированным числом параметров, которые делают сильные предположения.
- Но есть класс непараметрических моделей, которые не делают предположений почти никаких (это не совсем правда), а основаны непосредственно на данных; они в некоторых ситуациях очень хороши, но плохо обобщаются на высокие размерности и большие датасеты.

Метод ближайших соседей

- Пример непараметрической модели: метод ближайших соседей.
- Давайте на примере задачи классификации.
- Не будем строить вообще никакой модели, а будем классифицировать новые примеры как

$$\hat{y}(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} y_i,$$

где $N_k(\mathbf{x})$ – множество k ближайших соседей точки $\mathbf{x}$ среди имеющихся данных $(\mathbf{x}_i, y_i)_{i=1}^N$.

Метод ближайших соседей

- Единственный «параметр» – это k, но от него многое зависит.
- Для разумно большого k у нас в нашем примере стало меньше ошибок.
- Но это не предел – для $k = 1$ на тестовых данных вообще никаких ошибок нету!
- Что это значит? В чём недостаток метода ближайших соседей при $k = 1$?
- Как выбрать k? Можно ли просто подсчитать ошибку классификации и минимизировать её?

Проклятие размерности

- В прошлый раз k-NN давали гораздо более разумные результаты, чем линейная модель, особенно если хорошо выбрать k.
- Может быть, нам в этой жизни больше ничего и не нужно?
- Давайте посмотрим, как k-NN будет вести себя в более высокой размерности (что очень реалистично).

Проклятие размерности

- Давайте поищем ближайших соседей у точки в единичном гиперкубе. Предположим, что наше исходное распределение равномерное.
- Чтобы покрыть долю α тестовых примеров, нужно (ожидаемо) покрыть долю α объёма, и ожидаемая длина ребра гиперкуба-окрестности в размерности p будет $e_p(\alpha) = \alpha^{1/p}$.
- Например, в размерности 10 $e_{10}(0.1) = 0.8$, $e_{10}(0.01) = 0.63$, т.е. чтобы покрыть 1% объёма, нужно взять окрестность длиной больше половины носителя по каждой координате!
- Это скажется и на k-NN: трудно отвергнуть по малому числу координат, быстрые алгоритмы хуже работают.

Проклятие размерности

- Второе проявление the curse of dimensionality: пусть N точек равномерно распределены в единичном шаре размерности p. Тогда среднее расстояние от нуля до точки равно

$$d(p, N) = \left(1 - \frac{1}{2}^{1/N}\right)^{1/p},$$

  т.е., например, в размерности 10 для $N = 500$ $d \approx 0.52$, т.е. больше половины.
- Большинство точек в результаты ближе к границе носителя, чем к другим точкам, а это для ближайших соседей проблема – придётся не интерполировать внутри существующих точек, а экстраполировать наружу.

Проклятие размерности

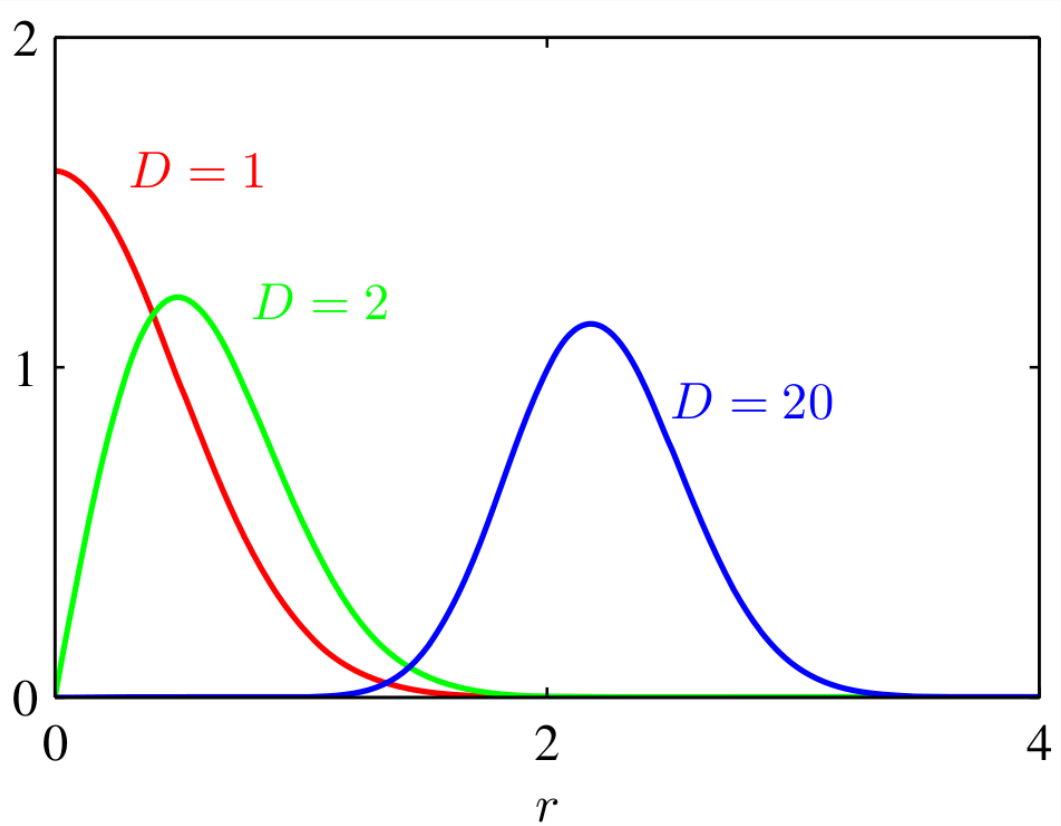
- Третье проявление: проблемы в оптимизации, которые и имел в виду Беллман.
- Если нужно примерно оптимизировать функцию от d переменных, на решётке с шагом ϵ понадобится примерно $\left(\frac{1}{\epsilon}\right)^d$ вычислений функции.
- В численном интегрировании – чтобы интегрировать функцию с точностью ϵ, нужно тоже примерно $\left(\frac{1}{\epsilon}\right)^d$ вычислений.

Проклятие размерности

- Плотные множества становятся очень разреженными. Например, чтобы получить плотность, создаваемую в размерности 1 при помощи $N = 100$ точек, в размерности 10 нужно будет 100^{10} точек.
- Поведение функций тоже усложняется с ростом размерности – чтобы строить регрессии в высокой размерности с той же точностью, может потребоваться экспоненциально больше точек, чем в низкой размерности.
- А у линейной модели ничего такого не наблюдается, она не подвержена проклятию размерности.

Проклятие размерности

- Ещё пример: нормально распределённая величина будет сосредоточена в тонкой оболочке.



Упражнение. Переведите плотность нормального распределения в полярные координаты и проверьте это утверждение.

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 $\mathbf{x} \in \mathbb{R}^p$, непрерывный вещественный выход $y \in \mathbb{R}$; у них есть некоторое совместное распределение $p(\mathbf{x}, y)$.
- Мы хотим найти функцию $f(\mathbf{x})$, которая лучше всего предсказывает y.

Функция потери

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

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

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

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

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

Функция потери

- Это можно переписать как

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

  и, значит, можно теперь минимизировать EPE поточечно:

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

  а это можно решить и получить

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

- 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:

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

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

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

- Это сразу два приближения: ожидание через среднее и среднее в точке через среднее в ближних точках.
- Иначе говоря, k-NN предполагает, что в окрестности $\mathbf{x}$ функция $y(\mathbf{x})$ 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:

$$f(\mathbf{x}) \approx \mathbf{x}^\top \mathbf{w}.$$

- Теперь мы не берём условие по $\mathbf{x}$, как в k-NN, а просто собираем много значений для разных $\mathbf{x}$ и обучаем модель.

Классификация

- То же самое можно и с задачей классификации сделать. Пусть у нас переменная g с K возможными значениями $g_1, \ldots, g_k$ предсказывается.
- Введём функцию потери, равную 1 за каждый неверный ответ. Получим

$$\mathrm{EPE} = \mathsf{E}\left[L(g, \hat{g}(\mathsf{x}))\right].$$

- Перепишем как раньше:

$$\mathrm{EPE} = \mathsf{E}_{\mathsf{x}} \sum_{k=1}^{K} L(g_k, \hat{g}(\mathsf{x})) p(g_k \mid \mathsf{x}).$$

- Опять достаточно оптимизировать поточечно:

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

Классификация

- Опять достаточно оптимизировать поточечно:

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

- Для 0-1 функции потери это упрощается до

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

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

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

Bias-variance decomposition

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

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

Bias-variance decomposition

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

$$\mathsf{E}[L] = \mathsf{E}[(y - f(\mathbf{x}))^2] = \mathsf{E}[(y - \mathsf{E}[y \mid \mathbf{x}] + \mathsf{E}[y \mid \mathbf{x}] - f(\mathbf{x}))^2] =$$
$$= \int (f(\mathbf{x}) - \mathsf{E}[y \mid \mathbf{x}])^2 \, p(\mathbf{x}) d\mathbf{x} + \int (\mathsf{E}[y \mid \mathbf{x}] - y)^2 \, p(\mathbf{x}, y) d\mathbf{x} dy,$$

потому что

$$\int (f(\mathbf{x}) - \mathsf{E}[y \mid \mathbf{x}]) (\mathsf{E}[y \mid \mathbf{x}] - y) \, p(\mathbf{x}, y) d\mathbf{x} dy = 0.$$

Bias-variance decomposition

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

$$\mathsf{E}[L] = \int \left(f(\mathbf{x}) - \mathsf{E}\left[y \mid \mathbf{x}\right]\right)^2 p(\mathbf{x}) d\mathbf{x} + \int \left(\mathsf{E}\left[y \mid \mathbf{x}\right] - y\right)^2 p(\mathbf{x}, y) d\mathbf{x} dy,$$

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

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

- A noise, $\int \left(\mathsf{E}\left[y \mid \mathbf{x}\right] - y\right)^2 p(\mathbf{x}, y) d\mathbf{x} dy$, – 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 $\hat{f}(\mathbf{x}) = \mathbf{E}\left[y \mid \mathbf{x}\right]$, и всё.
- Однако жизнь – борьба, и у нас есть только ограниченный датасет из N точек. Предположим, что этот датасет берётся по распределению $p(\mathbf{x}, y)$ – т.е. фактически рассмотрим много-много экспериментов такого вида:
  - взяли датасет D из N точек по распределению $p(\mathbf{x}, y)$;
  - подсчитали нашу чудо-регрессию;
  - получили новую функцию предсказания $f(\mathbf{x}; D)$.
- 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 $\left(f(\mathbf{x}) - \hat{f}(\mathbf{x})\right)^2$, а теперь будет $\left(f(\mathbf{x}; D) - \hat{f}(\mathbf{x})\right)^2$, и его можно усреднить по D, применив такой же трюк:

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

и в ожидании получится...

Bias-variance decomposition

- ...и в ожидании получится

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

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

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

Bias-variance-noise

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

где

$$(\text{bias})^2 = \left(\mathbf{E}_D\left[f(\mathbf{x}; D)\right] - \hat{f}(\mathbf{x})\right)^2,$$

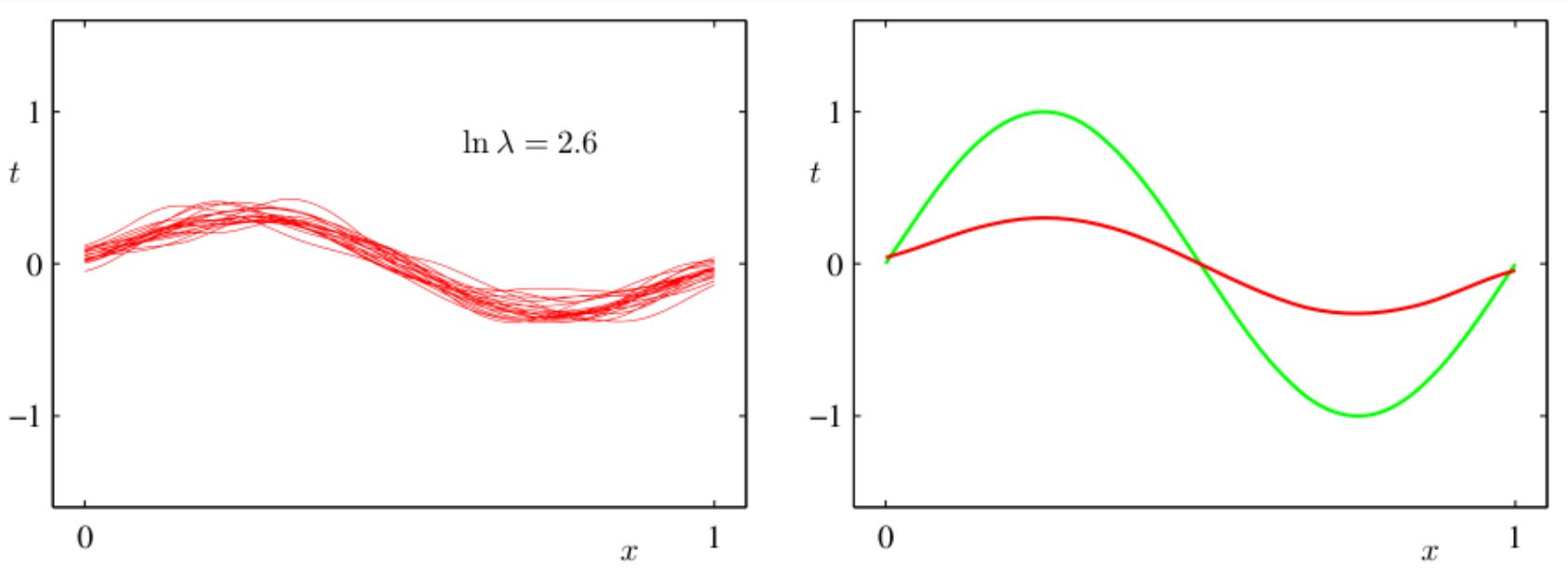
$$\text{variance} = \mathbf{E}_D\left[\left(f(\mathbf{x}; D) - \mathbf{E}_D\left[f(\mathbf{x}; D)\right]\right)^2\right],$$

$$\text{noise} = \int \left(\mathbf{E}\left[y \mid \mathbf{x}\right] - y\right)^2 p(\mathbf{x}, y) d\mathbf{x} dy.$$

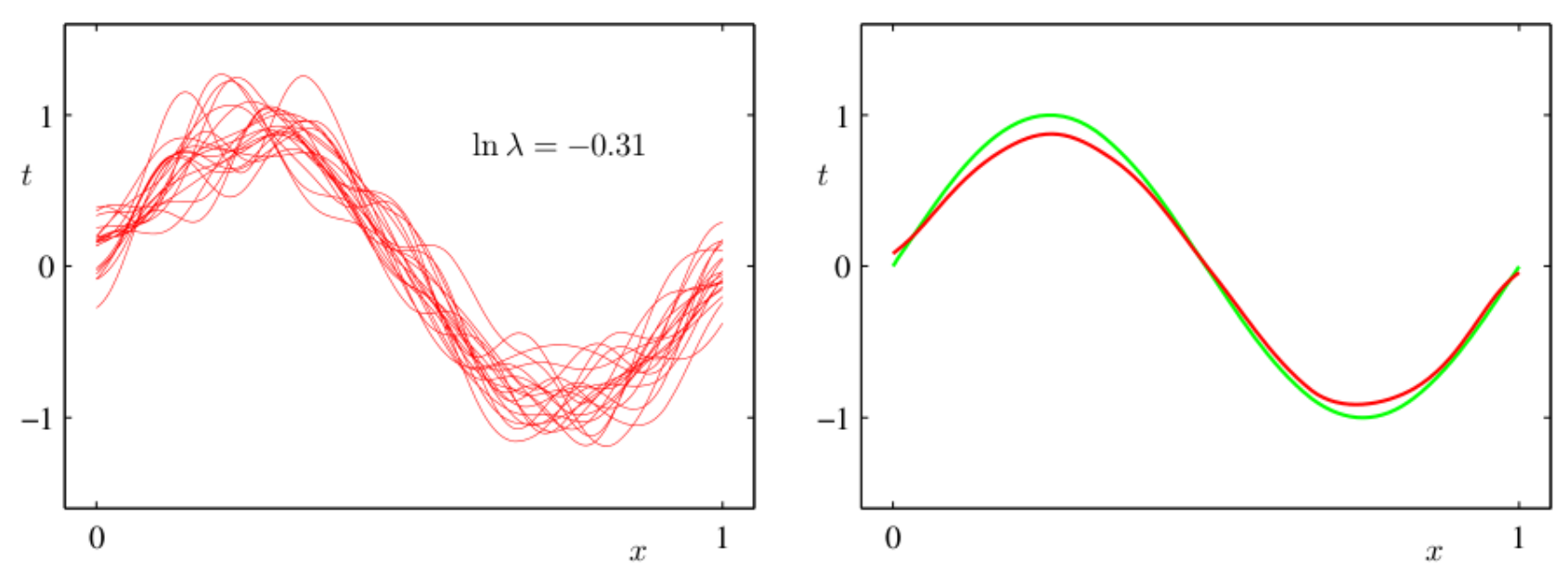
Пример

- 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.

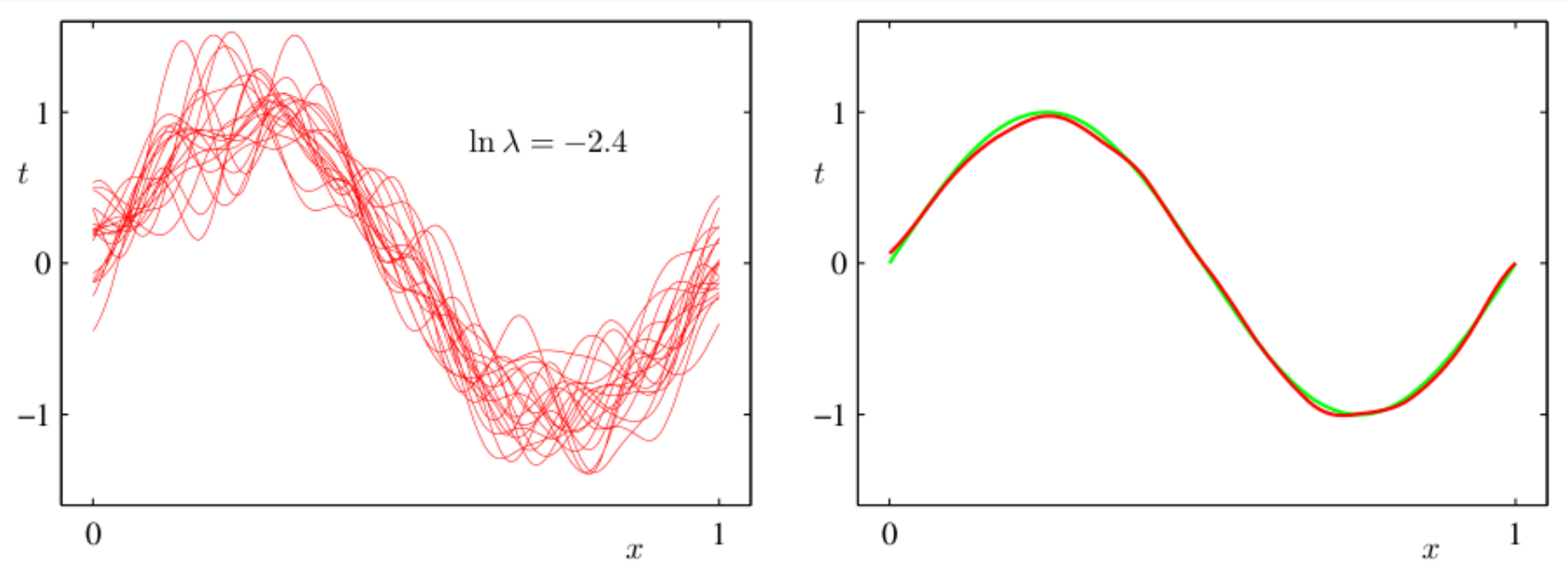
Регуляризатор и bias-variance



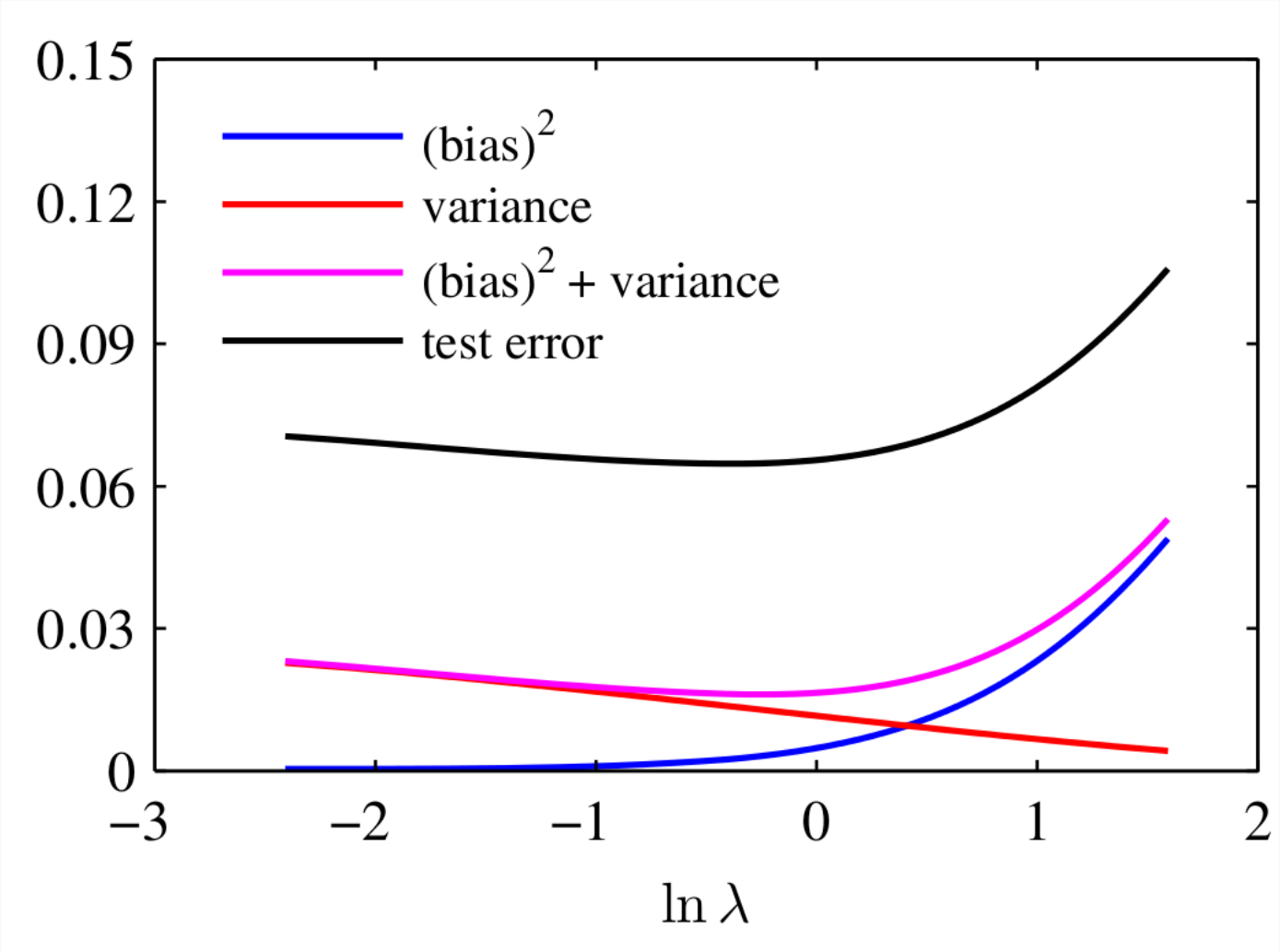
Регуляризатор и bias-variance



Регуляризатор и bias-variance



Регуляризатор и bias-variance



Спасибо!

Спасибо за внимание!