# Статистическая теория принятия решений

Сергей Николенко

образование

Академия больших данных MADE — VK

14 февраля 2022 г.

*Random facts*:

- 14 февраля — День святого Валентина, происходящий от луперкалий, праздника плодородия в честь богини «лихорадочной» любви Juno Februata и Фавна; во время луперкалий из шкур жертвенных животных изготавливались бичи, молодые люди брали их и голыми бежали по городу, ударяя бичом встретившихся на пути женщин; так и появился праздник любви; а 14 февраля 269 г. был казнён полевой врач и священник Валентин за нарушение приказа императора Клавдия о безбрачии для воинов: Валентин тайно венчал желающих по христианскому обряду
- 14 февраля у славян — Трифонов день; в Болгарии этот день зовётся Трифон Зарезан, так как отмечается во время первой обрезки виноградника; Трифон был братом Богородицы, оскорбил её и младенца по дороге в церковь, а затем раскаялся и отрезал себе нос
- 14 февраля 1349 г. по обвинению в распространении чумы в Страсбурге было убито около 2000 евреев, из них 900 сожжено заживо; а 14 февраля 1896 г. Теодор Герцль опубликовал в Берлине и Вене книгу «Еврейское государство. Опыт современного решения еврейского вопроса» (Der Judenstaat), которая стала идейным фундаментом раннего сионизма

# Байесовское сравнение моделей

# 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 $\{\mathcal{M}_i\}_{i=1}^{L}$.
- Модель – это распределение вероятностей над данными $D$.
- По тестовому набору $D$ можно оценить апостериорное распределение

$$p(\mathcal{M}_i \mid D) \propto p(\mathcal{M}_i)p(D \mid \mathcal{M}_i).$$

# Байесовское сравнение моделей

- Если знать апостериорное распределение, то можно сделать предсказание:

$$p(t \mid \mathbf{x}, D) = \sum_{i=1}^{L} p(t \mid \mathbf{x}, \mathcal{M}_i, \mathcal{D}) p(\mathcal{M}_i \mid D).$$

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

# Байесовское сравнение моделей

- Если модель определена параметрически, через $\mathbf{w}$, то

$$p(D \mid \mathcal{M}_i) = \int p(D \mid \mathbf{w}, \mathcal{M}_i) p(\mathbf{w} \mid \mathcal{M}_i) d\mathbf{w}.$$

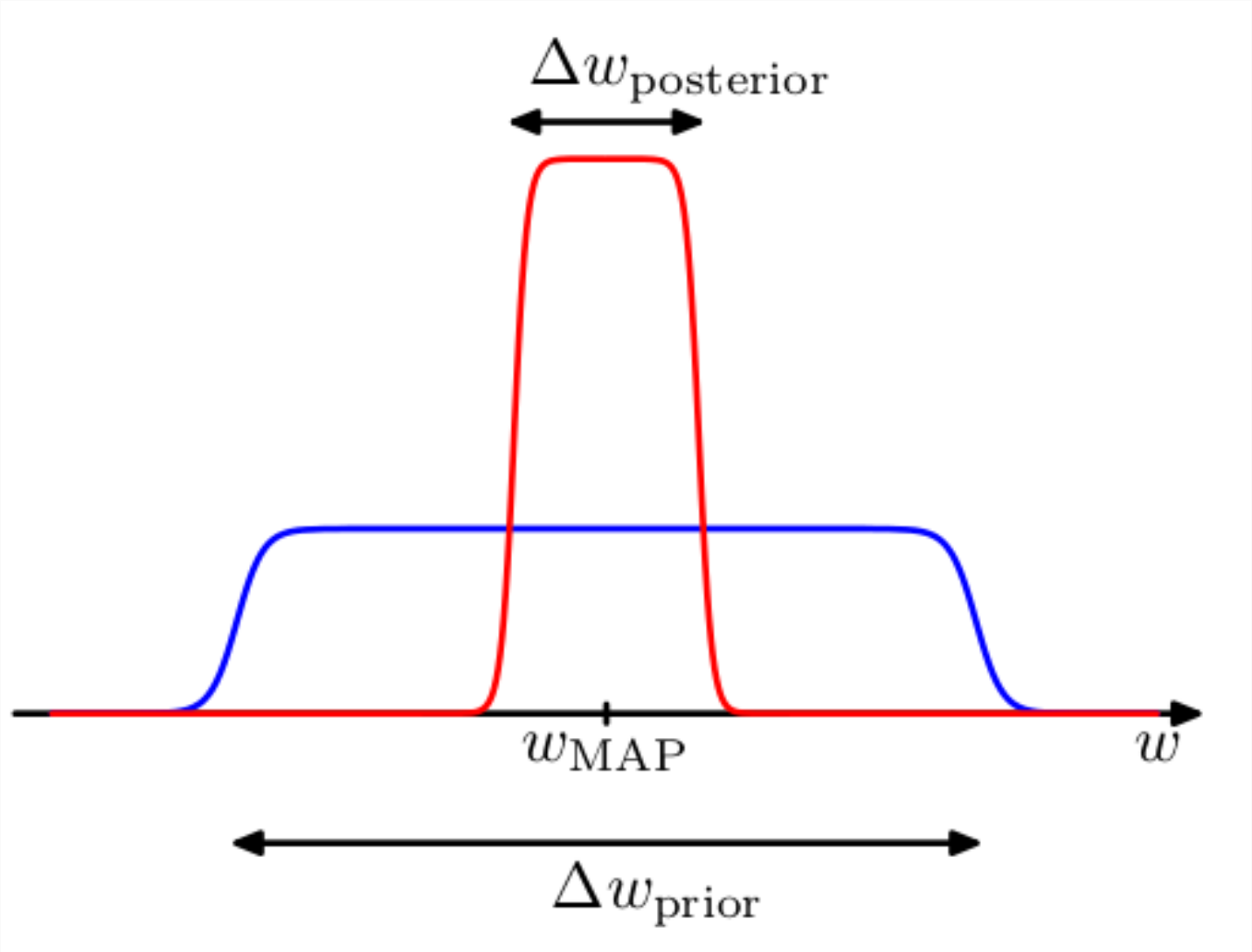
- Т.е. это вероятность сгенерировать $D$, если выбирать параметры модели по её априорному распределению, а потом накидывать данные.
- Это, кстати, в точности знаменатель из теоремы Байеса:

$$p(\mathbf{w} \mid \mathcal{M}_i, D) = \frac{p(D \mid \mathbf{w}, \mathcal{M}_i) p(\mathbf{w} \mid \mathcal{M}_i)}{p(D \mid \mathcal{M}_i)}.$$

# Байесовское сравнение моделей

- Предположим, что у модели один параметр $w$, а апостериорное распределение – это острый пик вокруг $w_{\mathrm{MAP}}$ 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

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

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

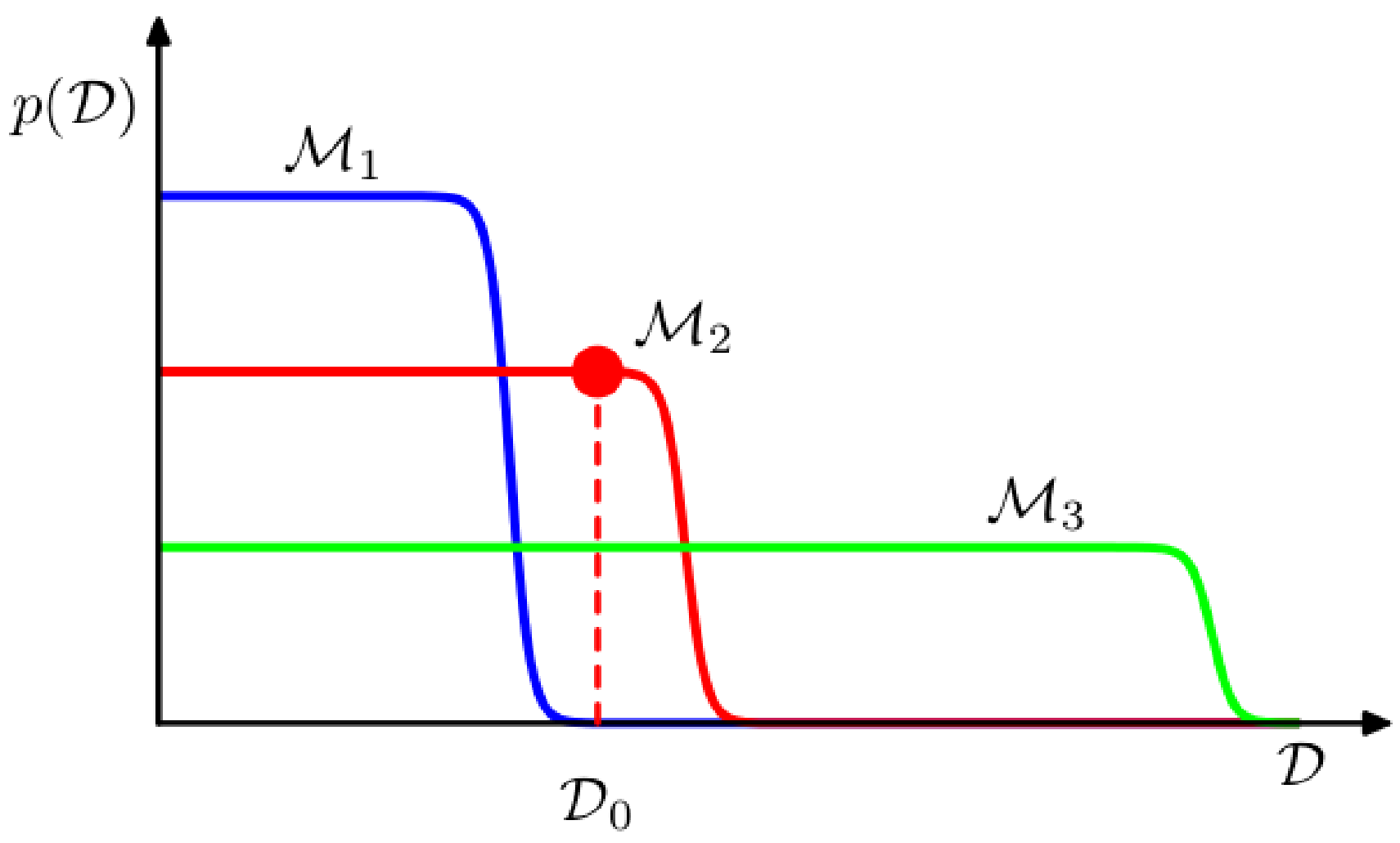
- Это значит, что мы добавляем штраф за «слишком узкое» апостериорное распределение – то есть в точности штраф за оверфиттинг!
- Для модели из $M$ параметров, если предположить, что у них одинаковые $\Delta w_{\mathrm{posterior}}$, получим

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

# 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 $p(D \mid \mathcal{M})$.
- Сложная модель (e.g., многочлен девятой степени) генерирует «много» разных датасетов, у неё низкая $p(D \mid \mathcal{M})$.
- 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 $p(D \mid \mathcal{M}_{\text{true}})$ всегда оптимальным в этом смысле?
- Конечно, для конкретного датасета может так повезти, что не будет.
- Но если усреднить по всем датасетам, выбранным по $p(D \mid \mathcal{M}_{\text{true}})$...

# Правильный ответ лучше

- ...то получится

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

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

# Эквивалентное ядро

# Эквивалентное ядро

- Вспомним наши байесовские предсказания:

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

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

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

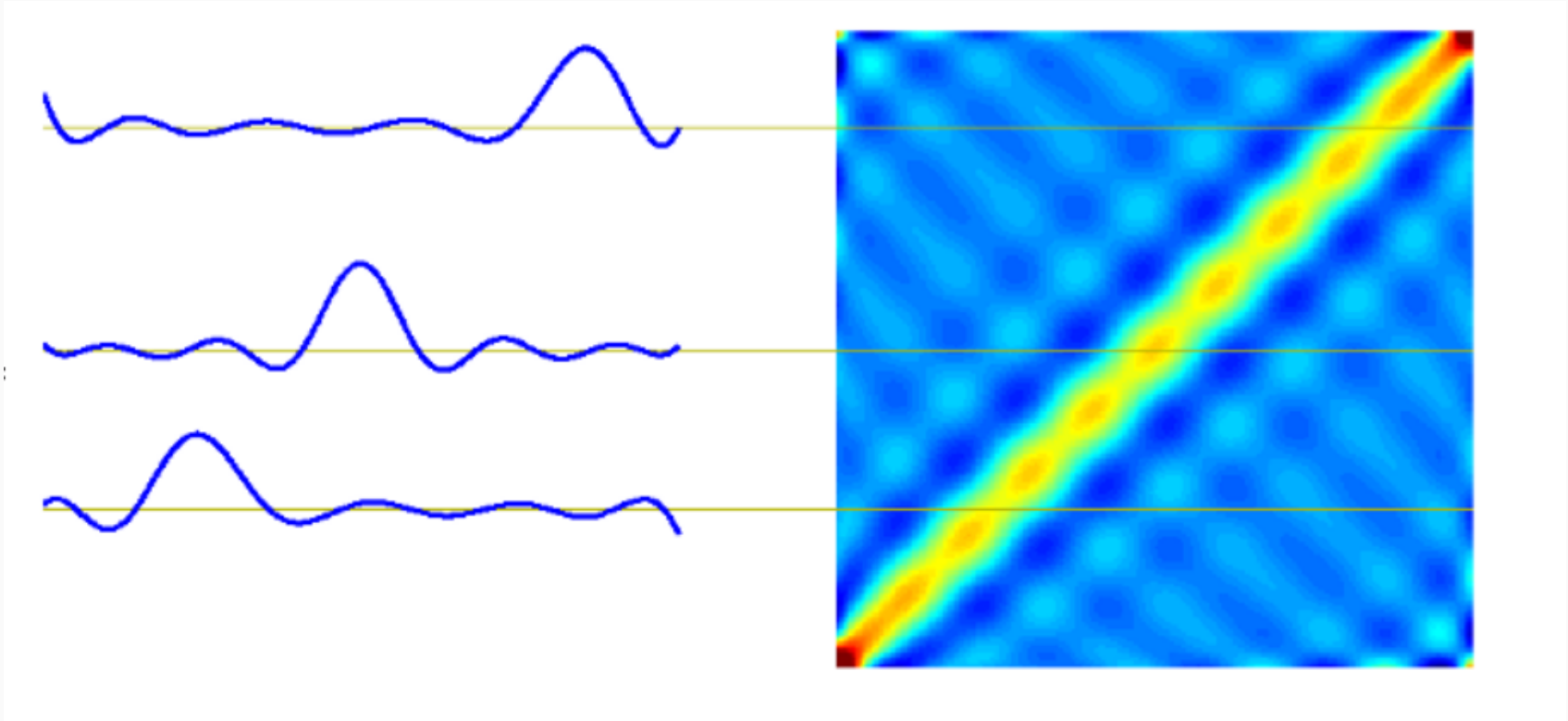
# Эквивалентное ядро

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

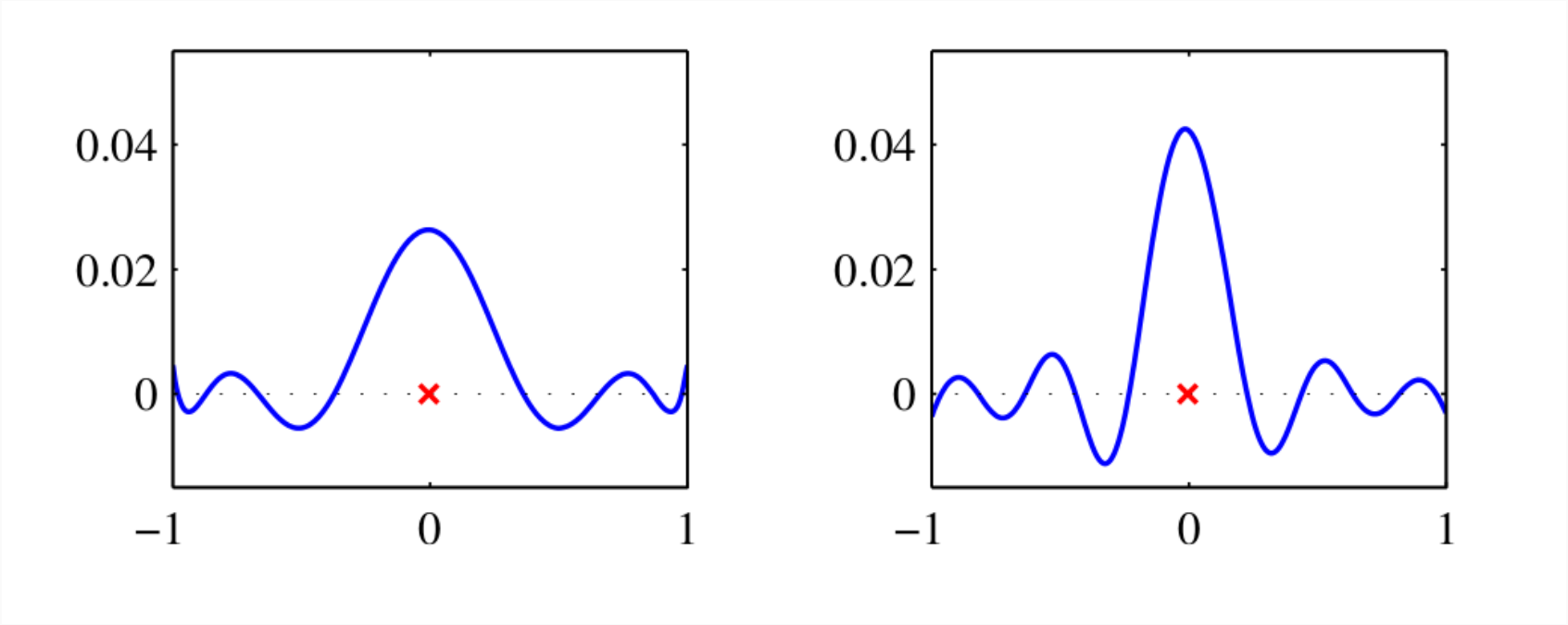
$$y(\mathbf{x}, \boldsymbol{\mu}_N) = \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) t_n.$$

- Т.е. мы предсказываем следующую точку как линейную комбинацию значений в известных точках.
- Функция $k(\mathbf{x}, \mathbf{x}') = \beta \boldsymbol{\phi}(\mathbf{x})^\top \Sigma_N \boldsymbol{\phi}(\mathbf{x}')$ 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 $\boldsymbol{\phi}$ – 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 $\alpha$ (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

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

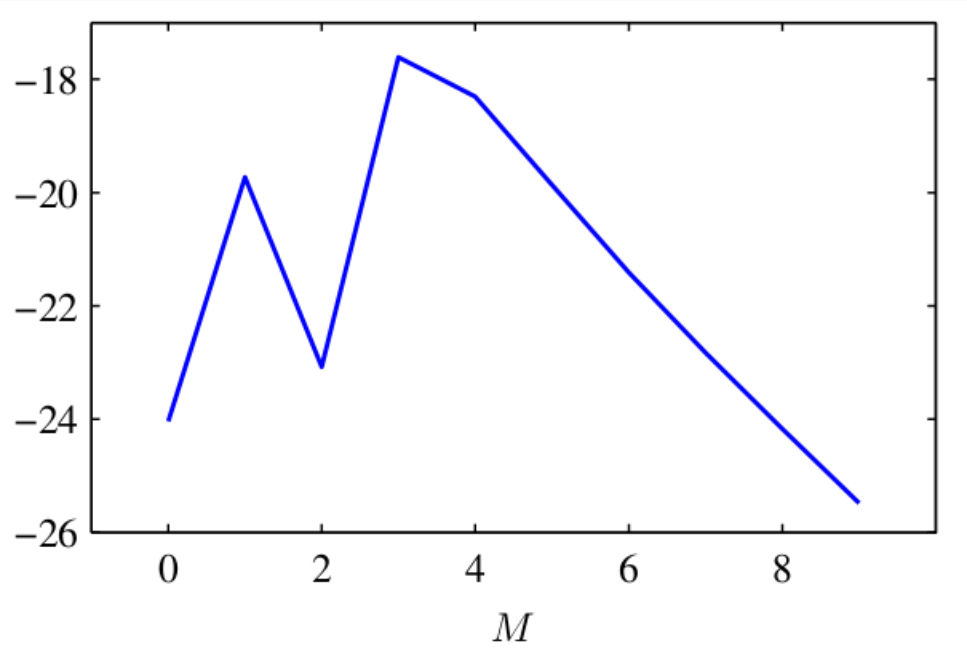
- Получается:

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

- Это теперь надо максимизировать по $\alpha$ и $\beta$, 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.

# Параметрические и непараметрические модели

- Ещё одно 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

# 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 $\alpha$ тестовых примеров, нужно (ожидаемо) покрыть долю $\alpha$ 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

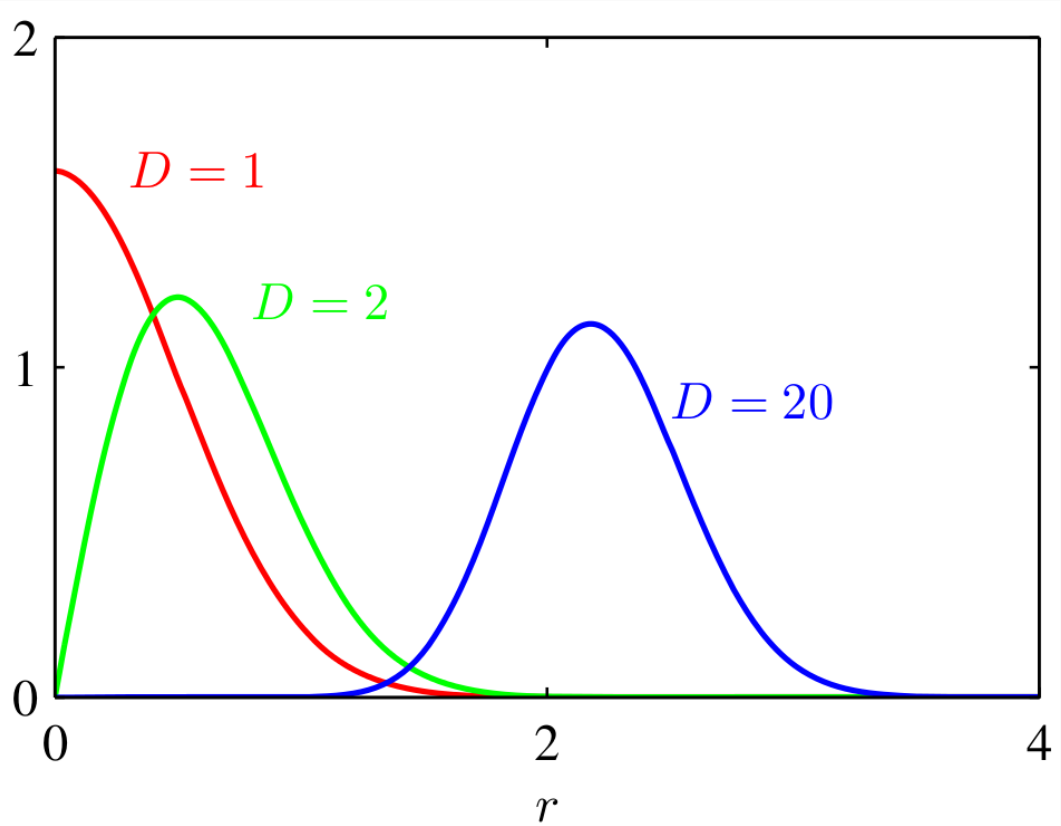
- Третье проявление: проблемы в оптимизации, которые и имел в виду Беллман.
- Если нужно примерно оптимизировать функцию от $d$ переменных, на решётке с шагом $\epsilon$ понадобится примерно $\left(\frac{1}{\epsilon}\right)^d$ вычислений функции.
- В численном интегрировании – чтобы интегрировать функцию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

# $k$-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.

# Спасибо!

Спасибо за внимание!