

# СТАТИСТИЧЕСКАЯ ТЕОРИЯ ПРИНЯТИЯ РЕШЕНИЙ

---

Сергей Николенко

СПбГУ — Санкт-Петербург

22 сентября 2017 г.

---

## *Random facts:*

- 22 сентября -- День защиты слонов, а также Всемирный день без автомобиля
- 22 сентября 1692 г. последние восемь ведьм были повешены в Салеме
- 22 сентября 1823 г. ангел Мороний явился Джозефу Смиту-младшему, поведал ему о золотых пластинах, зарытых на холме недалеко от дома Смита, и разрешил позаимствовать пластины для перевода

СТАТИСТИЧЕСКАЯ  
ТЕОРИЯ ПРИНЯТИЯ РЕШЕНИЙ

---

- 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  $y$ .

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

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

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

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

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

- Это можно переписать как

$$\text{ERE}(f) = E_{\mathbf{x}} E_{y|\mathbf{x}} [(y - f(\mathbf{x}))^2 | \mathbf{x}],$$

и, значит, можно теперь минимизировать ERE поточечно:

$$\hat{f}(\mathbf{x}) = \arg \min_c E_{y|\mathbf{x}'} [(y - c)^2 | \mathbf{x}' = \mathbf{x}],$$

а это можно решить и получить

$$\hat{f}(\mathbf{x}) = E_{y|\mathbf{x}'} (y | \mathbf{x}' = \mathbf{x}).$$

- Это решение называется *функцией регрессии* и является наилучшим предсказанием  $y$  в любой точке  $\mathbf{x}$ .

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

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

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

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

- 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.

- А линейная регрессия – это модельный подход, мы предполагаем, что функция регрессии линейна от своих аргументов:

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

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

- 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

$$\text{EPE} = \text{E} [L(g, \hat{g}(\mathbf{x}))].$$

- Перепишем как раньше:

$$\text{EPE} = \text{E}_{\mathbf{x}} \sum_{k=1}^K L(g_k, \hat{g}(\mathbf{x})) p(g_k | \mathbf{x}).$$

- Опять достаточно оптимизировать поточечно:

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



- Опять достаточно оптимизировать поточечно:

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

- Для 0-1 функции потери это упрощается до

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

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

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

- 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

$$\hat{f}(\mathbf{x}) = \mathbb{E}[y | \mathbf{x}] = \int yp(y | \mathbf{x})dx.$$

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

$$\begin{aligned} E[L] &= E[(y - f(\mathbf{x}))^2] = E[(y - E[y | \mathbf{x}] + E[y | \mathbf{x}] - f(\mathbf{x}))^2] = \\ &= \int (f(\mathbf{x}) - E[y | \mathbf{x}])^2 p(\mathbf{x}) d\mathbf{x} + \int (E[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy, \end{aligned}$$

ПОТОМУ ЧТО

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

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

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

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

$$f(\mathbf{x}) = \hat{f}(\mathbf{x}) = E[y | \mathbf{x}].$$

- А noise,  $\int (E[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) dx dy$ , – это просто свойство данных, дисперсия шума.

- Если бы у нас был всемогущий компьютер и неограниченный датасет, мы бы, конечно, на этом и закончили, посчитали бы  $\hat{f}(\mathbf{x}) = \mathbb{E}[y | \mathbf{x}]$ , 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...

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

$$\begin{aligned} & (f(\mathbf{x}; D) - \hat{f}(\mathbf{x}))^2 \\ &= (f(\mathbf{x}; D) - \mathbb{E}_D[f(\mathbf{x}; D)] + \mathbb{E}_D[f(\mathbf{x}; D)] - \hat{f}(\mathbf{x}))^2 \\ &= (f(\mathbf{x}; D) - \mathbb{E}_D[f(\mathbf{x}; D)])^2 + (\mathbb{E}_D[f(\mathbf{x}; D)] - \hat{f}(\mathbf{x}))^2 + 2(\dots)(\dots), \end{aligned}$$

и в ожидании получится...

- ...и в ожидании получится

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

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

Expected loss = (bias)<sup>2</sup> + variance + noise,

где

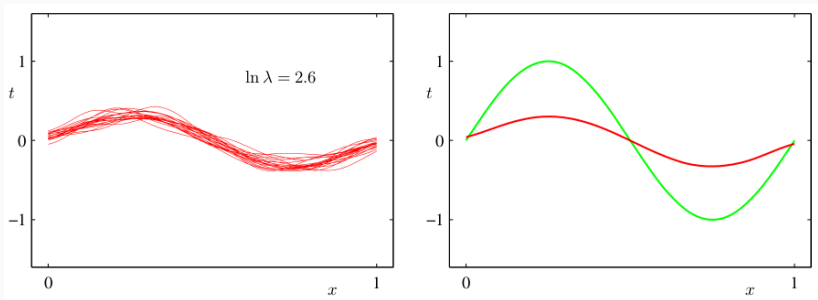
$$(\text{bias})^2 = \left( \mathbb{E}_D [f(\mathbf{x}; D)] - \hat{f}(\mathbf{x}) \right)^2,$$

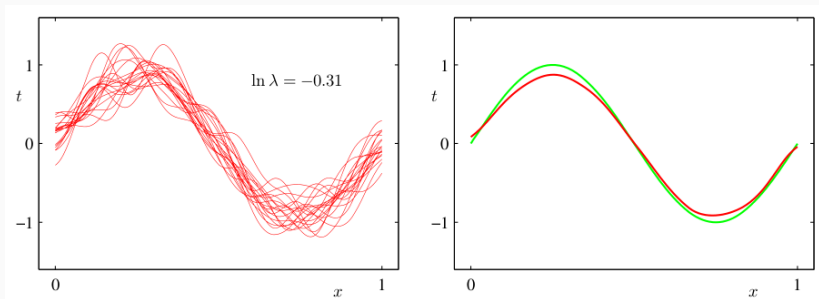
$$\text{variance} = \mathbb{E}_D \left[ (f(\mathbf{x}; D) - \mathbb{E}_D [f(\mathbf{x}; D)])^2 \right],$$

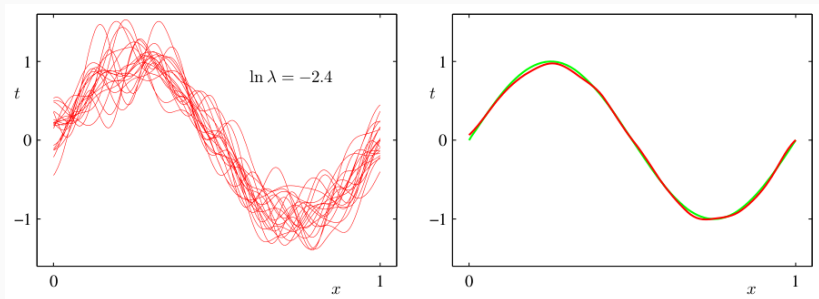
$$\text{noise} = \int (\mathbb{E}[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy.$$

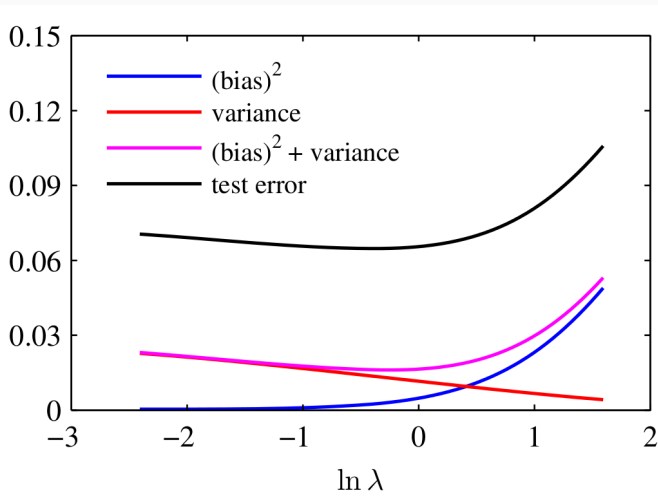


- Теперь давайте посмотрим на пример: опять та же синусоида, опять приближаем её линейной регрессией с полиномиальными признаками (максимальным их числом).
- И мы регуляризуем эту регрессию с параметром  $\alpha$ .
- Будем набрасывать много датасетов и смотреть, что меняется при этом.









ЭКВИВАЛЕНТНОЕ ЯДРО И  
СРАВНЕНИЕ МОДЕЛЕЙ

---

- Вспомним наши байесовские предсказания:

$$p(t | \mathbf{t}, \alpha, \beta) = \mathcal{N}(t | \mu_N^\top \phi(\mathbf{x}), \sigma_N^2),$$

$$\text{где } \sigma_N^2 = \frac{1}{\beta} + \phi(\mathbf{x})^\top \Sigma_N \phi(\mathbf{x}).$$

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

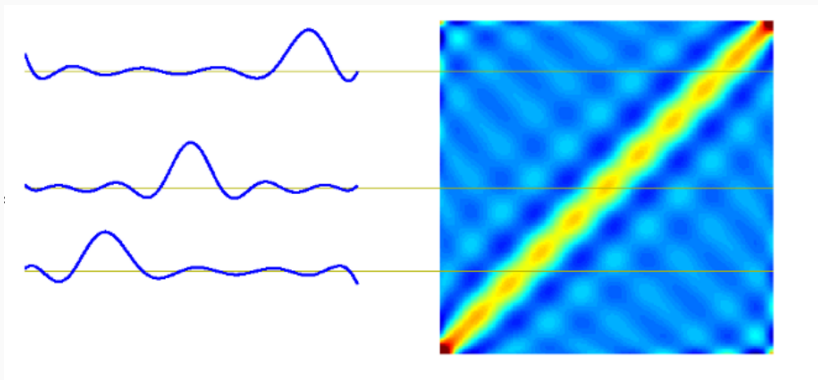
$$\begin{aligned} y(\mathbf{x}, \mu_N) &= \mu_N^\top \phi(\mathbf{x}) = \beta \phi(\mathbf{x})^\top \Sigma_N \Phi^\top \mathbf{t} = \\ &= \sum_{n=1}^N \beta \phi(\mathbf{x})^\top \Sigma_N \phi(\mathbf{x}_n) t_n. \end{aligned}$$

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

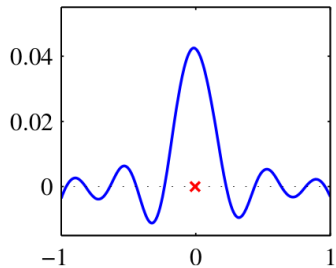
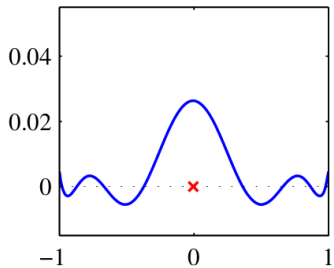
$$y(\mathbf{x}, \mu_N) = \sum_{n=1}^N k(\mathbf{x}, \mathbf{x}_n) t_n.$$

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





# 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  $\phi$  – такой подход мы ещё будем рассматривать.

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

- 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.

- 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

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

- Если знать апостериорное распределение, то можно сделать предсказание:

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

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

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

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

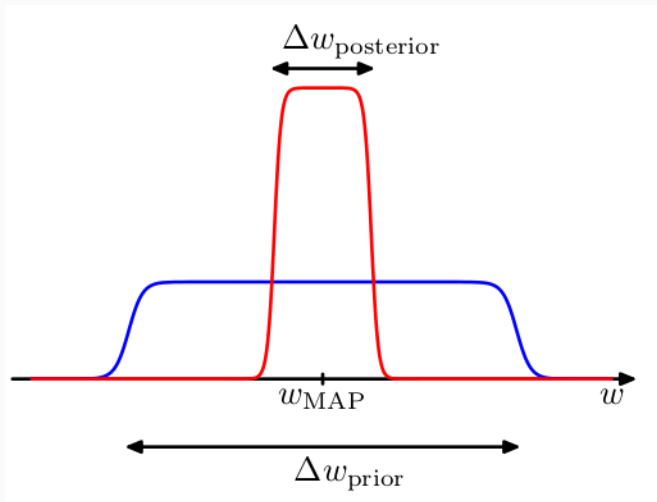
- Т.е. это вероятность сгенерировать  $D$ , если выбрать параметры модели по её априорному распределению, а потом накидывать данные.
- Это, кстати, в точности знаменатель из теоремы Байеса:

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

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



# ПРИБЛИЖЕНИЕ $p(d)$



- Тогда получится

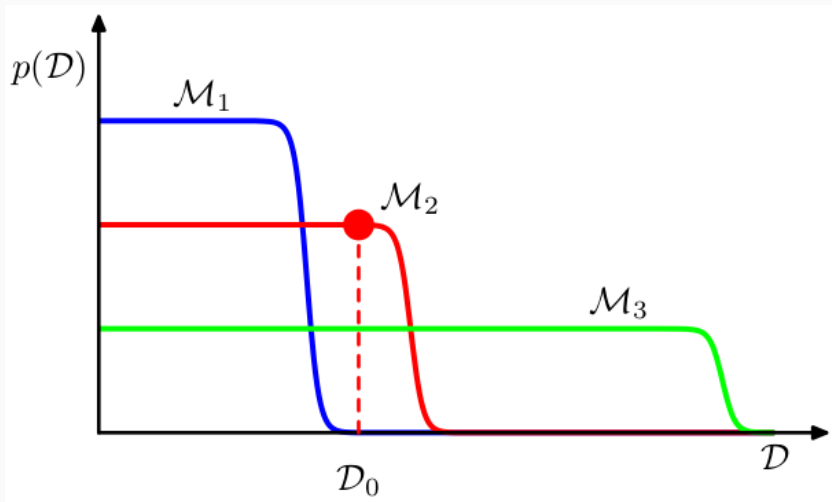
$$p(D) = \int p(D | w)p(w)dw \approx p(D | w_{\text{MAP}}) \frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}},$$
$$\ln p(D) \approx \ln p(D | w_{\text{MAP}}) + \ln \left( \frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}} \right).$$

- 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  $\Delta w_{\text{posterior}}$ , получим

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

- 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  $p(D | \mathcal{M})$ .
- Сложная модель (e.g., многочлен девятой степени) генерирует «много» разных датасетов, у неё низкая  $p(D | \mathcal{M})$ .
- Но сложная может хорошо выразить датасеты, которые не может выразить простая; поэтому в сумме надо выбирать «среднюю».

# ПРИБЛИЖЕНИЕ $p(d)$



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

- ...то получится

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

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

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

Спасибо за внимание!