

СТАТИСТИЧЕСКАЯ ТЕОРИЯ ПРИНЯТИЯ РЕШЕНИЙ

Сергей Николенко

СПбГУ — Санкт-Петербург

18 октября 2022 г.

Random facts:

- 18 октября 1009 г. «безумный халиф» Аль-Хаким Биамриллах разрушил Храм Гроба Господня, что позже послужило одним из поводов для начала крестовых походов
- 18 октября 1685 г. Людовик XIV отменил действие Нантского эдикта, дававшего вероисповедные права гугенотам: «Мы запрещаем допускать что-либо, сколько-нибудь похожее на уступку в пользу реформатской религии»; в итоге торговля пришла в упадок, а протестанты эмигрировали сотнями тысяч, в том числе в Россию и Америку
- 18 октября 1867 г. Аляска перешла от России к США, а 18 октября 1883 г. в Петербурге на месте убийства Александра II был заложен храм Спаса на крови
- 18 октября 1919 г. Лев Троцкий издал приказ обороняющей город 7-й армии: «Не писать ложных сведений о жестоких боях там, где была жестокая паника. За неправду карать, как за измену. Военное дело допускает ошибки, но не ложь, обман и самообман»
- 18 октября 1978 г. Анатолий Карпов сохранил титул чемпиона мира, выиграв у Виктора Корчного матч в Багио; перед последней, 32-й партией Карпов съездил в Манилу на финальный матч чемпионата мира по баскетболу, а затем получил белыми в защите Пирца-Уфимцева большое преимущество, и Корчной сдался без доигрывания

ЛАПЛАСОВСКАЯ АППРОКСИМАЦИЯ
И БАЙЕСОВСКАЯ
ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

- Небольшое лирическое отступление: как приблизить сложное распределение простым?
- Например, как приблизить гауссианом возле максимума? (естественная задача)
- Рассмотрим пока распределение от одной непрерывной переменной $p(z) = \frac{1}{Z}f(z)$.

- Первый шаг: найдём максимум z_0 .
- Второй шаг: разложим в ряд Тейлора

$$\ln f(z) \approx \ln f(z_0) - \frac{1}{2}A(z - z_0)^2, \text{ где } A = -\frac{d^2}{dz^2} \ln f(z) \Big|_{z=z_0}.$$

- Третий шаг: приблизим

$$f(z) \approx f(z_0)e^{-\frac{A}{2}(z-z_0)^2},$$

и после нормализации это будет как раз гауссиан.

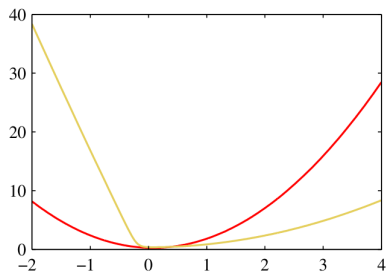
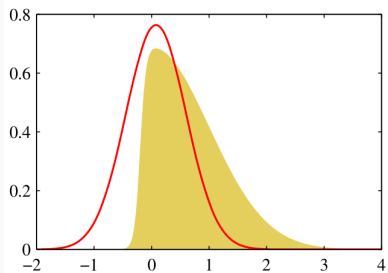
- Это можно обобщить на многомерное распределение $p(\mathbf{z}) = \frac{1}{Z} f(\mathbf{z})$:

$$f(\mathbf{z}) \approx f(\mathbf{z}_0) e^{-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^\top \mathbf{A}(\mathbf{z}-\mathbf{z}_0)},$$

$$\text{где } \mathbf{A} = -\nabla \nabla \ln f(\mathbf{z}) \big|_{z=z_0}.$$

Упражнение. Какая здесь будет нормировочная константа?

ЛАПЛАСОВСКАЯ АППРОКСИМАЦИЯ



СРАВНЕНИЕ МОДЕЛЕЙ ПО ЛАПЛАСУ

- Вооружившись лапласовской аппроксимацией, давайте применим её сначала к выбору моделей.
- Напомним: чтобы сравнить модели из множества $\{M_i\}_{i=1}^L$, по тестовому набору D оценим апостериорное распределение

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

- Если модель определена параметрически, то $p(D | M_i) = \int p(D | \theta, M_i)p(\theta | M_i)d\theta$.
- Это вероятность сгенерировать D , если выбирать параметры модели по её априорному распределению; знаменатель из теоремы Байеса:

$$p(\theta | M_i, D) = \frac{p(D | \theta, M_i)p(\theta | M_i)}{p(D | M_i)}.$$

- Мы раньше приближали фактически кусочно-постоянной функцией.
- Теперь давайте гауссианом приблизим; возьмём интеграл:

$$Z = \int f(\mathbf{z}) d\mathbf{z} \approx \int f(\mathbf{z}_0) e^{-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^\top \mathbf{A}(\mathbf{z}-\mathbf{z}_0)} d\mathbf{z} = f(\mathbf{z}_0) \frac{(2\pi)^{M/2}}{|\mathbf{A}|^{1/2}}.$$

- А у нас $Z = p(D)$, $f(\theta) = p(D | \theta)p(\theta)$.

- Получаем

$$\ln p(D) \approx \ln p(D | \theta_{\text{MAP}}) + \ln P(\theta_{\text{MAP}}) + \frac{M}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{A}|.$$

- $\ln P(\theta_{\text{MAP}}) + \frac{M}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{A}|$ – фактор Оккама.
- $\mathbf{A} = -\nabla\nabla \ln p(D | \theta_{\text{MAP}})p(\theta_{\text{MAP}}) = -\nabla\nabla \ln p(\theta_{\text{MAP}} | D)$.

- Получаем

$$\ln p(D) \approx \ln p(D | \theta_{\text{MAP}}) + \ln P(\theta_{\text{MAP}}) + \frac{M}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{A}|.$$

- Если гауссовское априорное распределение $p(\theta)$ достаточно широкое, и \mathbf{A} полного ранга, то можно грубо приблизить (докажите это!)

$$\ln p(D) \approx \ln p(D | \theta_{\text{MAP}}) - \frac{1}{2} M \ln N,$$

где M – число параметров, N – число точек в D , а аддитивные константы мы опустили.

- Это *байесовский информационный критерий* (Bayesian information criterion, BIC), он же *критерий Шварца* (Schwarz criterion).

- Теперь давайте обработаем логистическую регрессию по-байесовски.
- Логистическую регрессию так просто не выпишешь, как линейную – точного ответа из произведения логистических сигмоидов не получается.
- Будем приближать по Лапласу.

- Априорное распределение выберем гауссовским:

$$p(\mathbf{w}) = N(\mathbf{w} \mid \mu_0, \Sigma_0).$$

- Тогда апостериорное будет

$$\begin{aligned} p(\mathbf{w} \mid \mathbf{t}) &\propto p(\mathbf{w})p(\mathbf{t} \mid \mathbf{w}), \text{ и} \\ \ln p(\mathbf{w} \mid \mathbf{t}) &= -\frac{1}{2} (\mathbf{w} - \mu_0)^\top \Sigma_0^{-1} (\mathbf{w} - \mu_0) \\ &\quad + \sum_{n=1}^N [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)] + \text{const}, \\ \text{где } y_n &= \sigma(\mathbf{w}^\top \phi_n). \end{aligned}$$

- Чтобы приблизить, сначала находим максимум \mathbf{w}_{MAP} , а потом матрица ковариаций – это матрица вторых производных

$$\Sigma_N = -\nabla\nabla \ln p(\mathbf{w} | \mathbf{t}) = \Sigma_0^{-1} + \sum_{n=1}^N y_n(1 - y_n)\phi_n\phi_n^\top.$$

- Наше приближение – это

$$q(\mathbf{w}) = N(\mathbf{w} | \mathbf{w}_{\text{MAP}}, \Sigma_N).$$

- Теперь можно описать байесовское предсказание:

$$p(C_1 | \phi, \mathbf{t}) = \int p(C_1 | \phi, \mathbf{w})p(\mathbf{w} | \mathbf{t})d\mathbf{w} \approx \int \sigma(\mathbf{w}^\top \phi)q(\mathbf{w})d\mathbf{w}.$$

- Заметим, что $\sigma(\mathbf{w}^\top \phi)$ зависит от \mathbf{w} только через его проекцию на ϕ .
- Обозначим $a = \mathbf{w}^\top \phi$:

$$\sigma(\mathbf{w}^\top \phi) = \int \delta(a - \mathbf{w}^\top \phi)\sigma(a)da.$$

- $\sigma(\mathbf{w}^\top \phi) = \int \delta(a - \mathbf{w}^\top \phi) \sigma(a) da$, а значит,

$$\int \sigma(\mathbf{w}^\top \phi) q(\mathbf{w}) d\mathbf{w} = \int \sigma(a) p(a) da,$$

$$\text{где } p(a) = \int \delta(a - \mathbf{w}^\top \phi) q(\mathbf{w}) d\mathbf{w}.$$

- $p(a)$ – это маргинализация гауссиана $q(\mathbf{w})$, где мы интегрируем по всему, что ортогонально ϕ .

- $p(a)$ – это маргинализация гауссиана $q(\mathbf{w})$, где мы интегрируем по всему, что ортогонально ϕ .
- Значит, $p(a)$ – тоже гауссиан; найдём его моменты:

$$\mu_a = \mathbb{E}[a] = \int ap(a)da = \int q(\mathbf{w})\mathbf{w}^\top\phi d\mathbf{w} = \mathbf{w}_{\text{MAP}}^\top\phi,$$

$$\begin{aligned}\sigma_a^2 &= \int (a^2 - \mathbb{E}[a])^2 p(a)da = \\ &= \int q(\mathbf{w}) [(\mathbf{w}^\top\phi)^2 - (\mu_N^\top\phi)^2]^2 d\mathbf{w} = \phi^\top \Sigma_N \phi.\end{aligned}$$

- Итого получили, что

$$p(C_1 | \mathbf{t}) = \int \sigma(a)p(a)da = \int \sigma(a)N(a | \mu_a, \sigma_a^2)da.$$

- $p(C_1 | \mathbf{t}) = \int \sigma(a)N(a | \mu_a, \sigma_a^2)da.$
- Этот интеграл так просто не взять, потому что сигмоид сложный, но можно приблизить, если приблизить $\sigma(a)$ через пробит: $\sigma(a) \approx \Phi(\lambda a)$ для $\lambda = \sqrt{\pi/8}$.

Упражнение. Докажите, что для $\lambda = \sqrt{\pi/8}$ у σ и Φ одинаковый наклон в нуле.

- А если мы перейдём к пробит-функции, то её свёртка с гауссианом будет просто другим пробитом:

$$\int \Phi(\lambda a) N(a \mid \mu, \sigma^2) da = \Phi\left(\frac{\mu}{\sqrt{\frac{1}{\lambda^2} + \sigma^2}}\right).$$

Упражнение. Докажите это.

- В итоге получается аппроксимация

$$\int \sigma(a) N(a | \mu, \sigma^2) da \approx \sigma(\kappa(\sigma^2)\mu),$$

$$\text{где } \kappa(\sigma^2) = \frac{1}{\sqrt{1 + \frac{\pi}{8}\sigma^2}}.$$

- И теперь, собирая всё вместе, мы получили распределение предсказаний:

$$p(C_1 | \phi, \mathbf{t}) = \sigma(\kappa(\sigma^2)\mu_a), \text{ где}$$

$$\mu_a = \mathbf{w}_{\text{MAP}}^\top \phi,$$

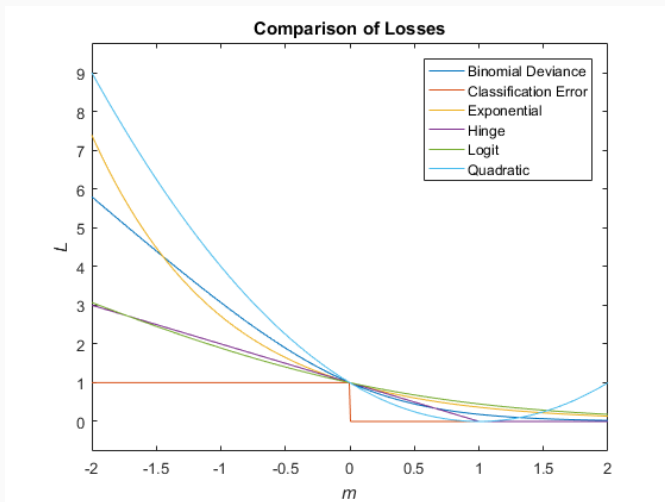
$$\sigma_a^2 = \phi^\top \Sigma_N \phi,$$

$$\kappa(\sigma^2) = \frac{1}{\sqrt{1 + \frac{\pi}{8}\sigma^2}}.$$

- Кстати, разделяющая поверхность $p(C_1 | \phi, \mathbf{t}) = \frac{1}{2}$ задаётся уравнением $\mu_a = 0$, и тут нет никакой разницы с просто использованием \mathbf{w}_{MAP} . Разница будет только для более сложных критериев.

- И напоследок немножко другой взгляд: разные методы классификации отличаются друг от друга тем, какую функцию ошибки они оптимизируют.
- У классификации проблема с «правильной» функцией ошибки, то есть ошибкой собственно классификации:
 - она и не везде дифференцируема,
 - и производная её никому не нужна.
- Давайте посмотрим на разные функции потерь (loss functions); мы уже несколько видели, но ещё немало осталось.

ФУНКЦИИ ПОТЕРЬ В КЛАССИФИКАЦИИ



ПРОКЛЯТИЕ РАЗМЕРНОСТИ

- 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.

- 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 $(\mathbf{x}_i, y_i)_{i=1}^N$.

- 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?

- 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).

- 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.

- 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.

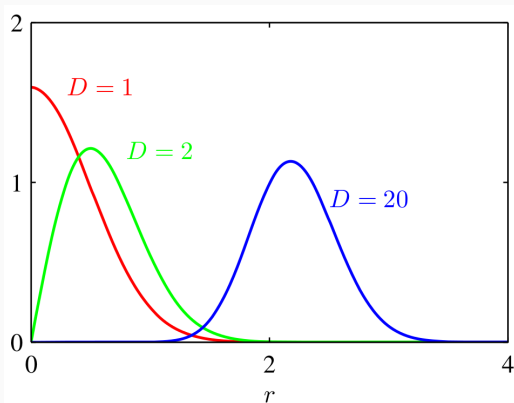
- Большинство точек в результате ближе к границе носителя, чем к другим точкам, а это для ближайших соседей проблема – придётся не интерполировать внутри существующих точек, а экстраполировать наружу.

- 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 $(\frac{1}{\epsilon})^d$ вычислений функции.
- В численном интегрировании – чтобы интегрировать функцию с точностью ϵ , нужно тоже примерно $(\frac{1}{\epsilon})^d$ вычислений.

- 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

- 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 *потери* (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

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

и, значит, можно теперь минимизировать ERE поточечно:

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

а это можно решить и получить

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

- 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.

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

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

- Перепишем как раньше:

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

- 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:

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

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

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

- А noise, $\int (E[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy$, – 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)² + variance + noise,

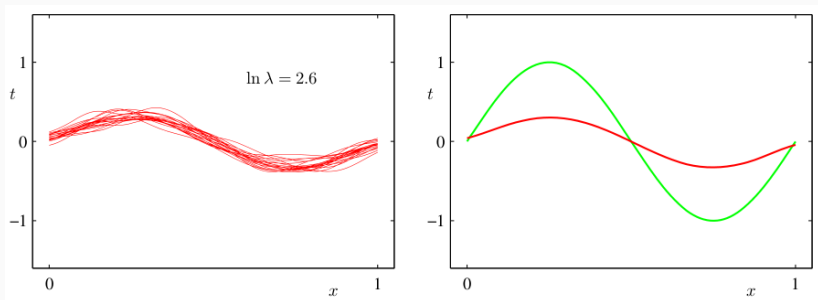
где

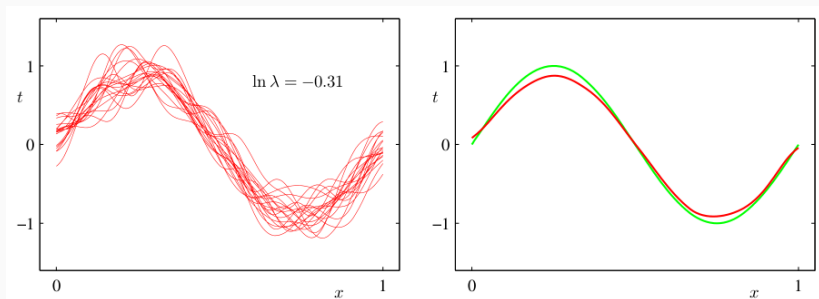
$$(\text{bias})^2 = \left(\mathbb{E}_D [f(\mathbf{x}; D)] - \hat{f}(\mathbf{x}) \right)^2,$$

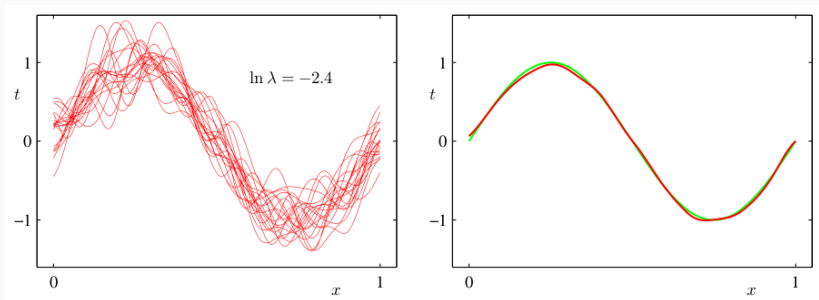
$$\text{variance} = \mathbb{E}_D \left[\left(f(\mathbf{x}; D) - \mathbb{E}_D [f(\mathbf{x}; D)] \right)^2 \right],$$

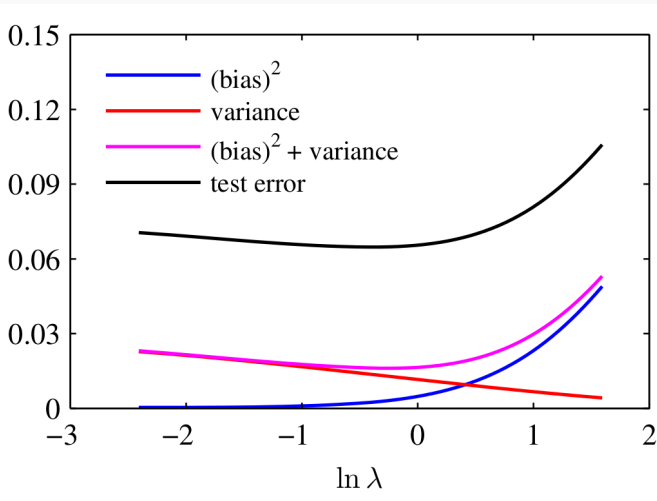
$$\text{noise} = \int (\mathbb{E}[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) d\mathbf{x}dy.$$

- 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 α .
- Будем набрасывать много датасетов и смотреть, что меняется при этом.









Спасибо за внимание!