

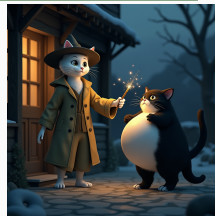
НЕСКОЛЬКО ОБЩИХ СЮЖЕТОВ

Сергей Николенко

СПбГУ — Санкт-Петербург

17 октября 2024 г.

Random facts:



- 17 октября в ООН — Международный день борьбы за ликвидацию нищеты; 17 октября 1987 года около 100 тысяч человек собрались в Париже, на площади Прав человека (Трокадеро), и поддержали лозунг отца Жозефа Вресински: «Там, где человек вынужден жить в нищете, нарушаются права человека. Объединиться в борьбе за уважение этих прав есть священный долг каждого»
- 17 октября 1825 г. в Париже прошла премьера оперы Ференца Листа «Дон Санчо, или Замок любви»; автору тогда ещё не было и четырнадцати
- 17 октября 1831 г. Майкл Фарадей открыл электромагнитную индукцию, 17 октября 1855 г. Генри Бессемер запатентовал свой процесс приготовления стали, 17 октября 1897 г. Константин Циолковский сообщил о постройке аэродинамической трубы, а 17 октября 1985 г. был выпущен Intel 80386, первый 32-разрядный процессор для IBM PC
- 17 октября 1869 г. по заданию газеты «New York Herald» Генри Стэнли отправился на поиски пропавшей в Африке экспедиции Дэвида Ливингстона
- 17 октября 1961 г. в нью-йоркском MOMA была выставлена картина Анри Матисса «Лодка» (Le Bateau); только 3 декабря кто-то заметил, что картина висит вверх ногами

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.

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

- 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.

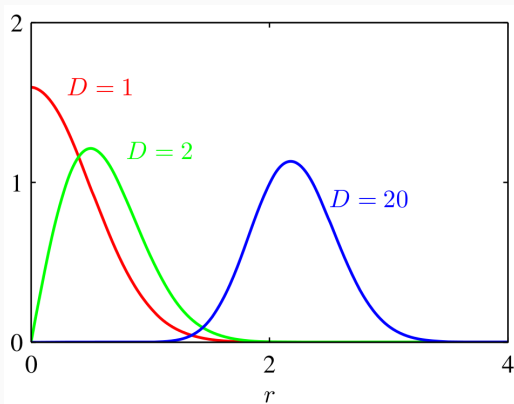
- Большинство точек в результате ближе к границе носителя, чем к другим точкам, а это для ближайших соседей проблема – придётся не интерполировать внутри существующих точек, а экстраполировать наружу.

- 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 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

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

и, значит, можно теперь минимизировать EPE 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:

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

- 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 $g_1, \dots, g_k$ предсказывается.
- Введём функцию потери, равную 1 за каждый неверный ответ. Получим

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

- Перепишем как раньше:

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

- Опять достаточно оптимизировать поточечно:

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

- 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

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

- 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:

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

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

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

- А noise, $\int (\mathbb{E}[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy$, – 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 $\hat{f}(\mathbf{x}) = \mathbb{E}[y \mid \mathbf{x}]$, 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:

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

и в ожидании получится...

- ...и в ожидании получится

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

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

Expected loss = (bias)² + variance + noise,

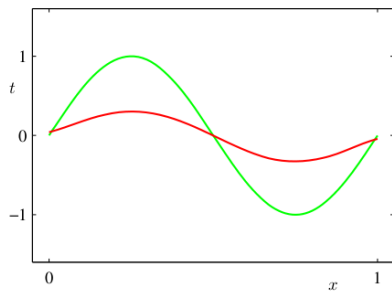
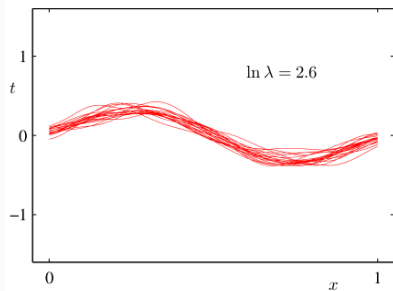
где

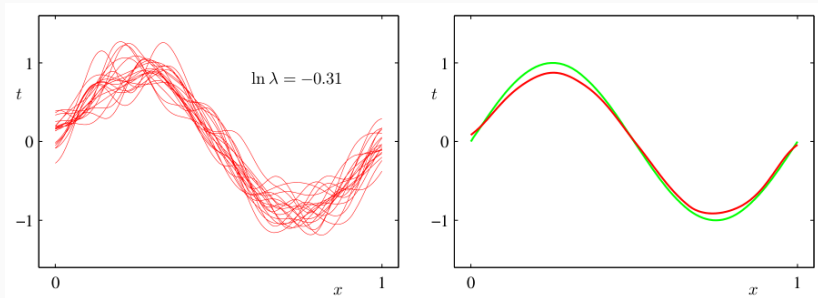
$$(\text{bias})^2 = \left(\mathbb{E}_D [f(\mathbf{x}; D)] - \hat{f}(\mathbf{x}) \right)^2,$$

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

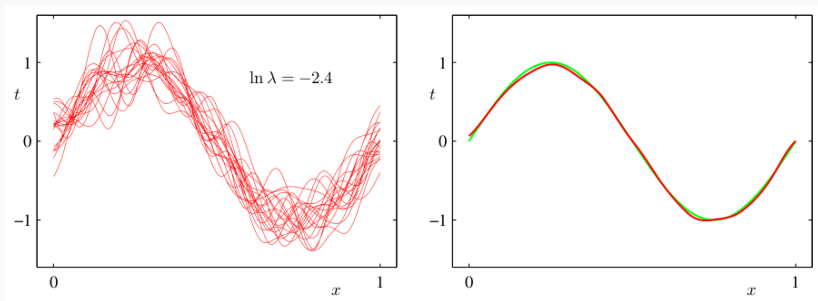
$$\text{noise} = \int (\mathbb{E}[y | \mathbf{x}] - y)^2 p(\mathbf{x}, y) d\mathbf{x} dy.$$

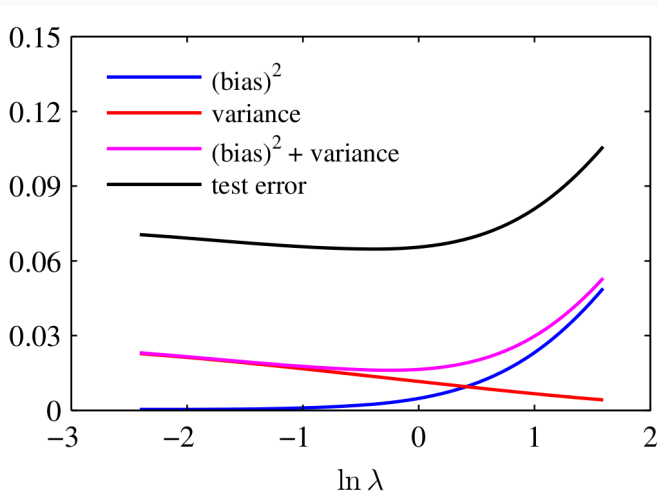
- Теперь давайте посмотрим на пример: опять та же синусоида, опять приближаем её линейной регрессией с полиномиальными признаками (максимальным их числом).
- И мы регуляризуем эту регрессию с параметром α .
- Будем набрасывать много датасетов и смотреть, что меняется при этом.





РЕГУЛЯРИЗАТОР И BIAS-VARIANCE





СПАСИБО!

Спасибо за внимание!

