ТЕХНОСФЕРА

Лекция 11
Байесовские методы

Всеволод Викулин

Содержание

1. Оптимальное байесовское правило и генеративные модели
2. Методы восстановления плотности и наивный байесовский классификатор
3. Байесовский подход в машинном обучении

Часть 1
Оптимальное байесовское правило

Вероятностное описание моделей

Вероятностное описание наиболее удобно для алгоритмов машинного обучение.

- Все модели описаны единым языком (см. Bishop Model Based Machine Learning)
- Мощный и выразительный язык теории вероятностей;
- Кодируем всю неопределенность в X -> y через вероятности;
- Позволяет строить модели, которые можно исследовать вероятностными методами;

Пример

Делаем банковский скоринг.

$p(y \mid x)$ — вероятность получить ответ y для объекта x, моделируем машинным обучением.

К нам пришли множество клиентов $X_{clients}$

Как посчитать возможные убытки?

$$RISK = \sum_{x \in X_{clients}} \lambda_{x,y} p(y \mid x)$$

Можно заранее рассчитать с помощью стохастического моделирования!

Байесовский подход

Считаем, что наши объекты и ответы описываются распределением $p(x, y)$, а у нас есть экземпляры из него (обучающая выборка)

Обучающая выборка взята **случайно и независимо** из этого распределения

$p(y)$ — **априорная** вероятность класса y

$p(x \mid y)$ — **правдоподобие** объекта x

$p(y \mid x)$ — **апостериорная** вероятность класса y

$p(x)$ — **обоснованность (evidence)** объекта x

A priori (латынь) — от предшествующего

A posteriori (латынь) — из последующего

Маргинализация

На самом деле достаточно знать только совместное распределение!

Правило суммирования (sum rule):

$$p(x) = \int p(x, y)dy, \;\; p(y) = \int p(x, y)dx$$

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)} = \frac{p(x, y)}{\int p(x, y)dx}$$

Формула Байеса

По определению условной вероятности

$$p(y|x) = \frac{p(x,y)}{p(x)}, p(x|y) = \frac{p(x,y)}{p(y)},$$

Если величины независимы, то $p(x,y) = p(x)p(y)$

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{p(x|y)p(y)}{\int p(x|y)p(y)dy}$$ — формула Байеса

Источник: en.wikipedia.org/wiki/Thomas_Bayes

Функционал среднего риска

Функция потерь $L(a, x, y)$ — неотрицательная функция, показывающая величину ошибки алгоритма a на объекте x с ответом y.

Функционал среднего риска — мат. ожидание функции потерь

$$R(a) = \int\int L(a, x, y)p(x, y)dxdy$$

$$a^* = \operatorname{argmin} R(a)$$

?

Какая связь с эмпирическим риском?

Связь с эмпирическим риском

Было:

$$Q(a, X, Y) = \frac{1}{N} \sum_{i=1}^{N} L(a, x_i, y_i), x_i \in X, y_i \in Y$$

$$a^* = \operatorname{argmin} Q(a, X_{train}, Y_{train})$$

Стало:

$$R(a) = \int\int L(a, x, y) p(x, y) dx dy$$

$$a^* = \operatorname{argmin} R(a)$$

Эмпирический риск это просто наша стохастическая оценка среднего риска, так как мы не знаем истинного распределения!

?

Почему просто не оптимизировать всегда средний риск?

Почему сложно оптимизировать средний риск

- Мы не знаем аналитический вид совместного распределения!
- Если бы знали, то мы бы учились брать интегралы, а не занимались машинным обучением :)
- Если бы знали, то машинное обучение было бы решенной задачей.

? Как же быть?

Как же быть?

- На самом деле, для оптимальной классификации совместное распределение знать и не нужно, достаточно только знать **апостериорное распределение** $p(y \mid x)$
- Но мы его тоже не знаем аналитически! Значит у нас возникает задача **оценки плотности распределения**
- Для регрессии есть аналогичный подход, разберите дома

Оптимальный байесовский классификатор

y — таргет, $\hat{y}$ — предсказание, функция потерь $L(y, \hat{y}) = [y \neq \hat{y}]$,

$$R(a) = \int\int L(a, x, y)p(x, y)dxdy = \sum_{s=1}^{Y} \int L(y, \hat{y})p(x, y)dx =$$

$$= \int \sum_{s=1}^{Y} [y \neq \hat{y}]p(x, y)dx = \int \sum_{y \neq \hat{y}} p(x, y)dx$$

По правилу суммирования:

$$\int \sum_{y \neq \hat{y}} p(x, y)dx + \int p(x, \hat{y})dx = 1$$

$$R(a) = 1 - \int p(x, \hat{y})dx \geq 1 - \int \max_{y \in Y} p(x, y) = R(a^*)$$

Оптимальный Байесовский классификатор:

$$a^*(x) = \operatorname*{argmax}_{y \in Y} p(x, y) = \operatorname*{argmax}_{y \in Y} p(y \,|\, x)p(x) = \operatorname*{argmax}_{y \in Y} p(y \,|\, x)$$

Maximum a posteriori (MAP) классификатор

Типы моделей в машинном обучении

Алгоритм $a(x)$ будем называть:

1. Генеративной вероятностной моделью, если он моделирует **совместное распределение** $p(x, y)$
2. Дискриминативной вероятностной моделью, если он моделирует **апостериорное распределение** $p(y \mid x)$
3. Дискриминативной моделью, если он моделирует только функцию $y = f(x)$

? Приведите примеры

Примеры

Дискриминативные вероятностные модели — логистическая регрессия

Дискриминативные модели — KNN, SVM, деревья

Генеративные вероятностные модели — еще не проходили, на этой лекции будет наивный байесовский классификатор!

? Какие плюсы и минусы дискриминативных и генеративных моделей?

Плюсы и минусы

$p(x, y) = p(y \mid x)p(x)$, разница, что генеративные помимо $p(y \mid x)$ моделируют $p(x)$

Плюсы дискриминативных моделей:

- **Намного** проще учить (что проще, научить модель отличать Ван Гога от Дега или научиться отличать, но еще уметь рисовать картину каждого художника?)
- Из-за этого, как правило, в типовых задачах работают лучше

Плюсы генеративных моделей:

- Более общие
- Легко работают с пропущенными значениями признаков
- Легко находить выбросы
- Способны **генерировать новые объекты**

?

Зачем генерировать объекты?

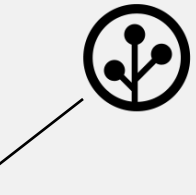
Зачем генерировать объекты?

- Это весело! (medium.com/coinmonks/celebrity-face-generation-using-gans-tensorflow-implementation-eaa2001eef86, thispersondoesnotexist.com)
- Получать объекты с нужными свойствами (например, молекулы, которые борются с определенными болезнями)
- Для датасета для другой модели
- Для отладки модели

Резюме первой части

- Вероятностное описание моделей позволяет более глубоко их исследовать.
- Мы должны оптимизировать средний риск, но из-за того, что у нас нету совместного распределения, мы считаем эмпирический.
- Оптимальный классификатор этот тот, который возвращает класс с наибольшей вероятностью $p(y \mid x)$ (регрессор смотрим дома)
- Бывают генеративные и дискриминативные модели. Генеративные моделируют $p(x)$, поэтому позволяют генерировать новые объекты

Часть 2
Методы восстановления плотности
Наивный байесовский классификатор



Оптимальный байесовский классификатор

Оптимальный Байесовский классификатор:

$$a^*(x) = \underset{y \in Y}{\operatorname{argmax}}\, p(y \mid x)$$

Аналитически не знаем, нужно сделать оценку

«Насколько вероятно, что целевая переменная равна классу y при условии, что этот объект имеет **вектор** весов x»

Считаем такую вероятность **по обучающей выборке**!

? Удобно ли считать такую вероятность напрямую?

Преобразовываем оптимальное байесовское правило

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}$$

$$a^*(x) = \underset{y \in Y}{\operatorname{argmax}}\, p(y \mid x) = \underset{y \in Y}{\operatorname{argmax}}\, \frac{p(x \mid y)p(y)}{p(x)} = \underset{y \in Y}{\operatorname{argmax}}\, p(x \mid y)p(y)$$

Вот так уже проще!

«Насколько вероятно, что этот объект имеет **вектор** весов x при условии, что целевая переменная равна классу y»

? Как оценить $p(y)$?

Как оценить $p(y)$

$p(y)$ это априорная вероятность понаблюдать объект из класса y

С какой вероятностью объект принадлежит классу y, если мы сами объект даже не видели?

Можно оценить как **долю** объектов из выборки.

Смысл формулы оптимального байесовского классификатора

$$a^*(x) = \underset{y \in Y}{\operatorname{argmax}}\, p(y \mid x) = \underset{y \in Y}{\operatorname{argmax}}\, p(x \mid y) p(y)$$

- $p(x \mid y)$ — насколько наш объект x похож на объекты из класса y, которые мы видели на обучении.
- $p(y)$ — насколько этот класс популярен

? Зачем тогда вообще нужен $p(y)$? Почему бы не брать просто $\underset{y \in Y}{\operatorname{argmax}}\, p(x \mid y)$

Потому что снова переобучение

$$a^*(x) = \underset{y \in Y}{\operatorname{argmax}}\, p(y \mid x) = \underset{y \in Y}{\operatorname{argmax}}\, p(x \mid y) p(y)$$

- $p(x \mid y)$ — насколько наш объект x похож на объекты из класса y, которые мы видели на обучении. Учимся оценивать далее
- $p(y)$ — позволяет уменьшить переобучение, то есть **регуляризация .** Уже умеем оценивать через долю класса

Методы восстановления плотности

Разбираемся, как оценить $p(x \mid y)$

1. Непараметрический — смотрим, какой процент точек из обучающей выборки лежит в окрестности оцениваемого значения.
2. Параметрический — предполагаем, что наше распределение из параметрического семейства. Настраиваем параметры распределения по обучающей выборке.
3. Восстановление из смеси распределений — предполагаем, что наше распределение смесь параметрических распределений.

На этой лекции обсудим первые два варианта.

Непараметрическое восстановление плотности

Хотим оценить $p(x \mid y)$ — насколько вероятно, что объект класса y принимает значение признака x (функция от x)

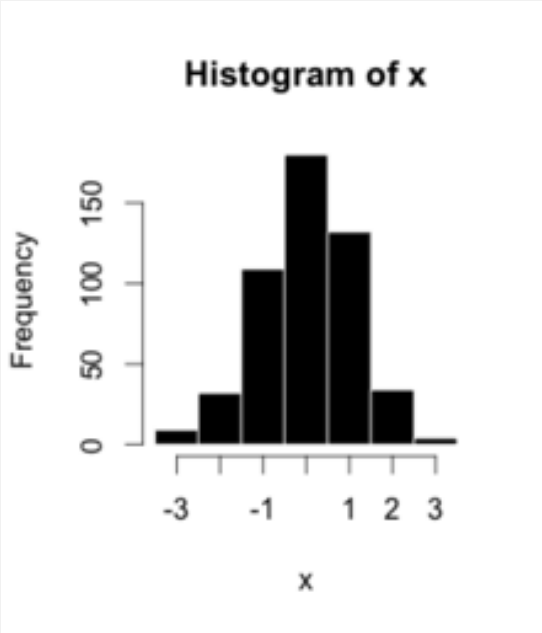
Для простоты считаем, что у нас один признак.

Основная идея — посмотрим, сколько объектов **на обучающей выборке** из класса y принимает такое значение или близкое.

Категориальный признак

Если признак категориальный, то просто строим гистограмму!

$p(x \mid y)$ — доля объектов класса y, у которые признак = x



Источник: en.wikipedia.org/wiki/Histogram

?

А если вещественный?

А если вещественный?

Точно так же, как и раньше, только дискретизируем вещественный признак с помощью окна ширины h, а потом считаем долю точек класса, которые попали в это окно!

?

Что в таком подходе может не устраивать?

Хотим гладкую плотность

Такая плотность будет кусочно-постоянной!

$$p(x\,|\,y) = \frac{1}{2Nh}\sum_{i=1}^{N}(|x - x_i| < h)[y_i = y]$$

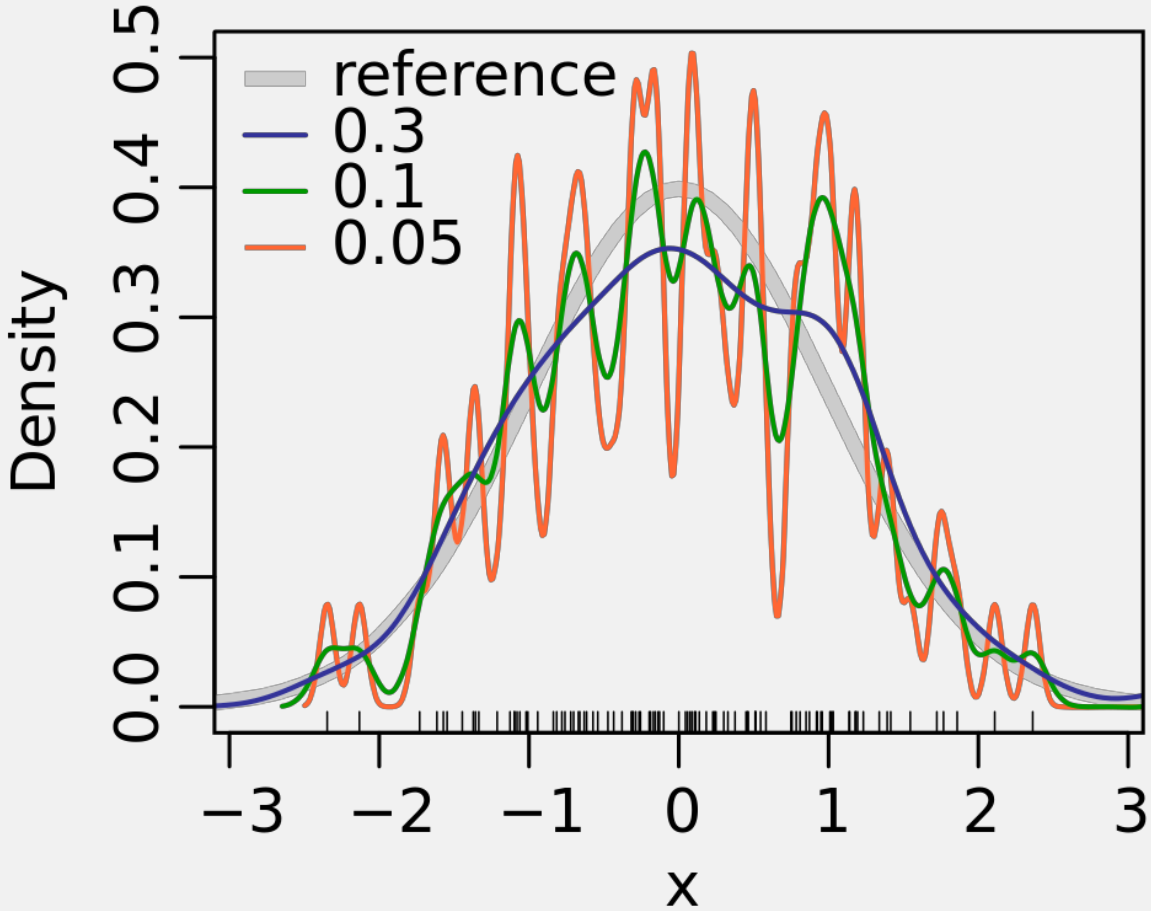
Можно использовать метод парзеновского окна:

$$p(x\,|\,y) = \frac{1}{Nh}\sum_{i=1}^{N}K(\frac{x - x_i}{h})[y_i = y]$$

где K - четная нормированная функция, которую называют ядром.

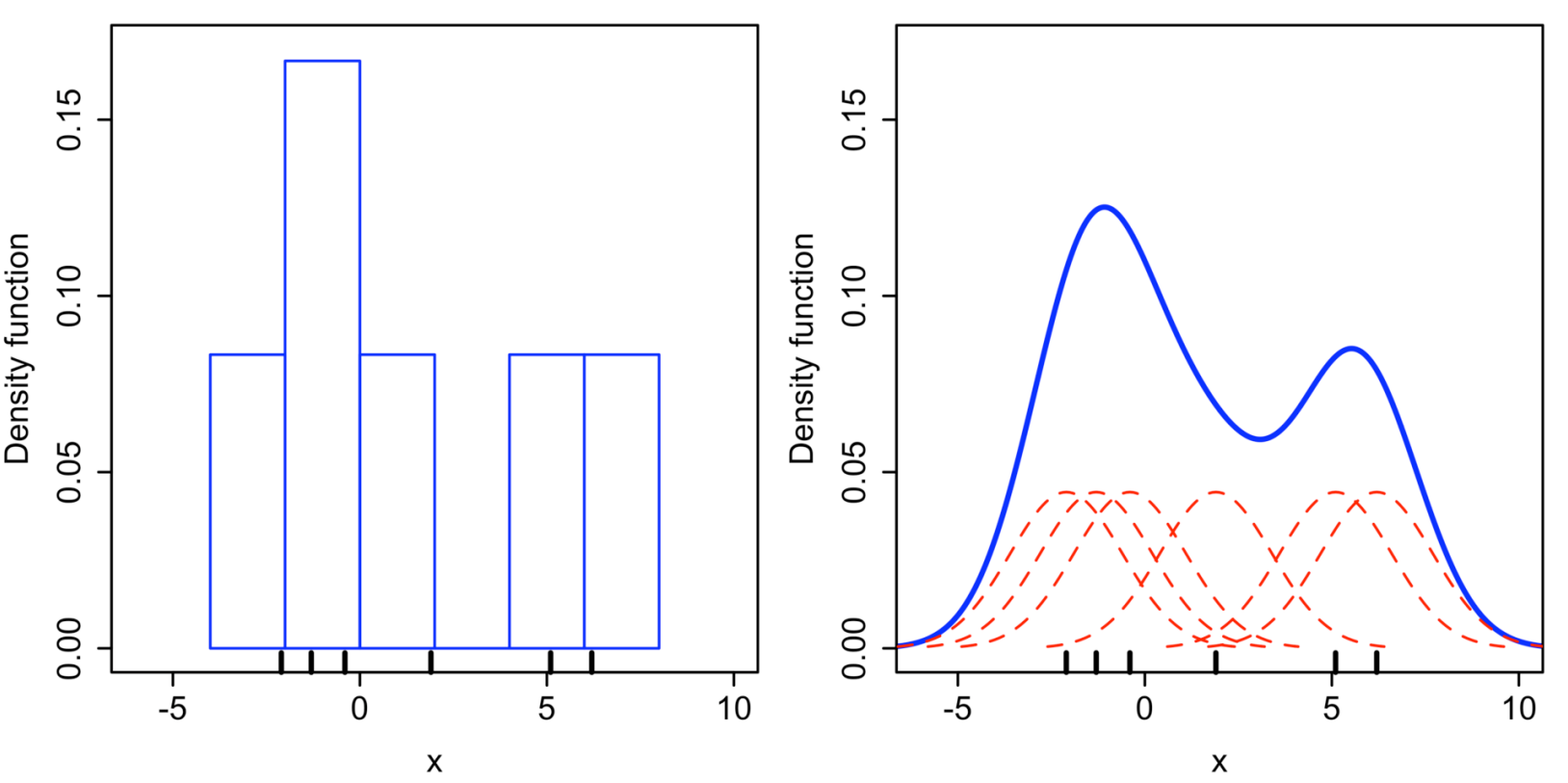
Например, Гауссовское — $K(r) = (2\pi)^{-0.5}\exp(-\frac{1}{2}r^2)$

Пример



Источник: en.wikipedia.org/wiki/Kernel_density_estimation

Пример



Источник: en.wikipedia.org/wiki/Kernel_density_estimation

А многомерно так можно?

Конечно! Вместо:

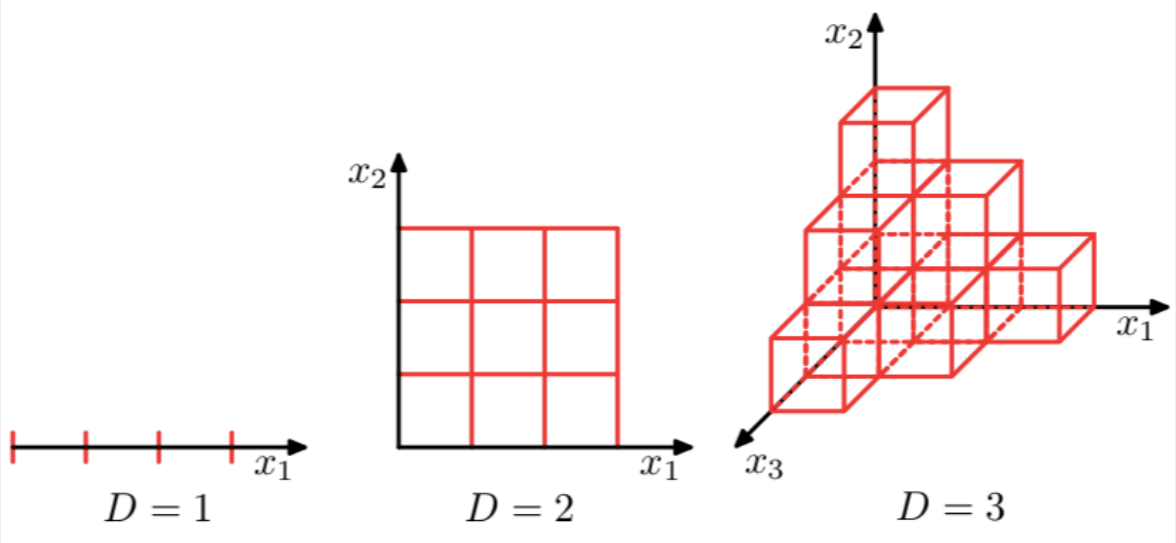
$$p(x|y) = \frac{1}{Nh}\sum_{i=1}^{N} K(\frac{x - x_i}{h})[y_i = y]$$

Делай:

$$p(x|y) = \frac{1}{Nh}\sum_{i=1}^{N} K(\frac{\rho(x, x_i)}{h})[y_i = y],$$ где ρ метрика в многомерном пространстве!

В чем может быть проблема?

Поклятие размерности!



Источник: Bishop

Все тоже самое, что в методе ближайшего соседа!

? Можно ли посчитать многомерную плотность с D измерениями, восстанавливая D раз одномерную?

Вспоминаем, что работаем с векторами

Задача: оценить $p(x \mid y)$, где x **вектор** признаков.

2 варианта восстановить плотность **многомерного** распределения:

- Восстанавливать в лоб многомерную плотность (проклятие размерности)
- Перейти от многомерной плотности к одномерным

$$p(x \mid y) = \prod_{j=1}^{D} p(x_j \mid y), \text{ где } D - \text{число признаков}$$

Предположение алгоритма наивного байеса — признаки независимы **при условии целевой переменной**

Параметрическое восстановление плотности

Хотим оценить $p(x \mid y)$ — насколько вероятно, что объект класса y принимает значение признака x (функция от x)

Для простоты считаем, что у нас один признак.

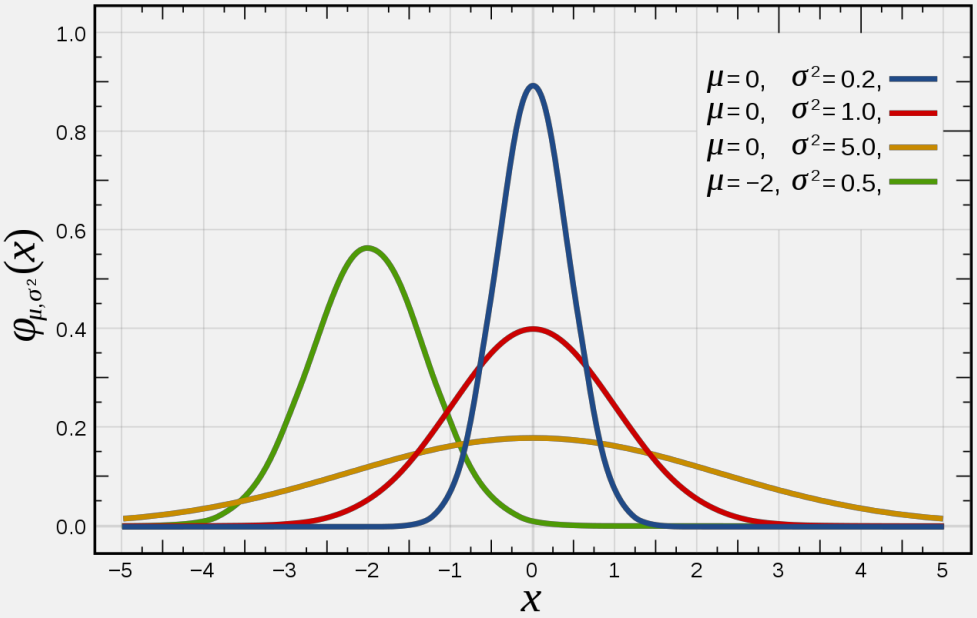
Основная идея — предположим, что наше $p(x \mid y)$ лежит в каком-то **параметрическом семействе** распределений $p(x \mid y, \theta)$, где θ — параметры, а затем находим параметры, используя **метод максимума правдоподобия.**

? Какие параметры у одномерного нормального распределения?

Параметры нормального распределения

Среднее μ и стандартное отклонение σ

$$p(x \mid y, \theta) = \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left(-\frac{(x-\mu_y)^2}{2\sigma_y^2}\right)$$



Источник: en.wikipedia.org/wiki/Normal_distribution

Метод максимума правдоподобия

Правильные те параметры, при которых пронаблюдать такую выборку максимально правдоподобно!

Случайная величина η — число луж во дворе

Параметр θ — (был дождь, поливали двор, ничего не было)

Выходите на улицу видите **выборку** — кругом лужи!

Чему **максимально вероятно** равна θ?

Метод максимума правдоподобия

Максимально вероятно, что «был дождь»

Пишем функцию правдоподобия, то есть вероятность пронаблюдать выборку, если все объекты берутся независимо:

$$L(\theta) = p(X|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

С суммой, как правило, удобнее работать. (Почему?)

$$\log L(\theta) = \sum_{i=1}^{N} \log p(x_i|\theta)$$

Правильные параметры те, которые максимизируют $L(\theta)$

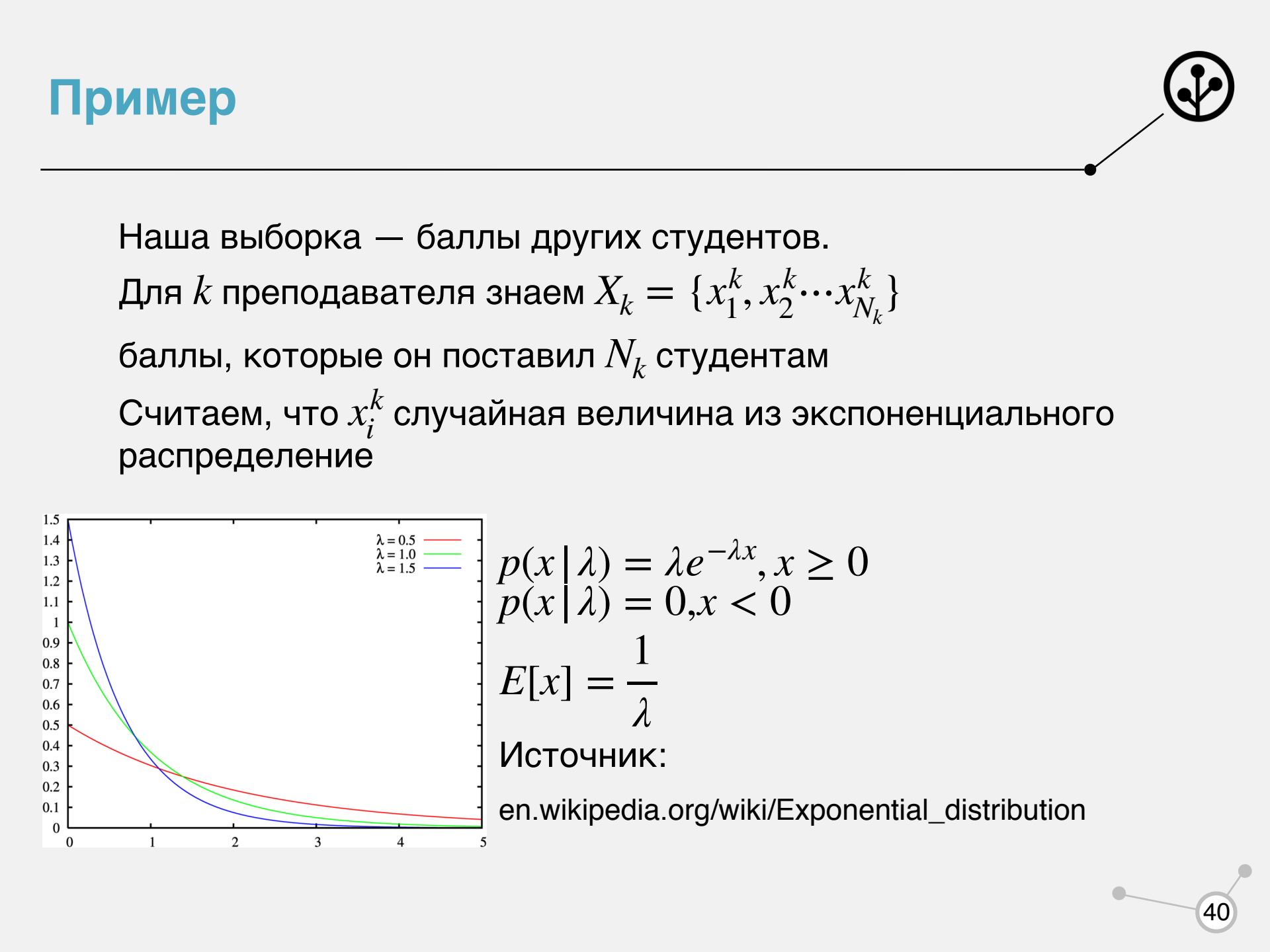
Пример

Идем на пересдачу. Преподаватель может поставить сколько угодно баллов (вещественное число) в зависимости от его «лояльности»

?

Как понять, к какому преподавателю идти?

Пример



Наша выборка — баллы других студентов.

Для k преподавателя знаем $X_k = \{x_1^k, x_2^k \cdots x_{N_k}^k\}$

баллы, которые он поставил N_k студентам

Считаем, что x_i^k случайная величина из экспоненциального распределение

$$p(x \mid \lambda) = \lambda e^{-\lambda x}, x \geq 0$$
$$p(x \mid \lambda) = 0, x < 0$$

$$E[x] = \frac{1}{\lambda}$$

Источник:

en.wikipedia.org/wiki/Exponential_distribution

Пример

Нужно просто оценить параметр λ_k по выборке из прошлых оценок преподавателя!

$$L(\lambda_k) = p(X_k \mid \lambda_k) = \prod_{i=1}^{N} p(x_i \mid \lambda_k), \text{ где } p(x_i \mid \lambda_k) = \lambda_k e^{-\lambda_k x_i}$$

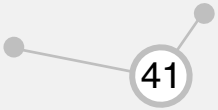
$$\log L(\lambda_k) = \sum_{i=1}^{N} \log p(x_i \mid \lambda_k) = \sum_{i=1}^{N} (\log \lambda_k - \lambda_k x_i) = N \log \lambda_k - \lambda_k \sum_{i=1}^{N} x_i$$

$$\frac{\partial \log L(\lambda_k)}{\partial \lambda_k} = \frac{N}{\lambda_k} - \sum_{i=1}^{N} x_i = 0$$

$$\lambda_k = \frac{N_k}{\sum_{i=1}^{N} x_i}, E[x_k] = \frac{1}{\lambda_k} = \frac{\sum_{i=1}^{N_k} x_i}{N_k}$$

Просто средние баллы, которые поставил преподаватель!

Выбираем того, у кого это среднее максимально.



Пример

Решив эту задачу, мы тем самым восстановили плотность!

Мы нашли параметр λ, то есть теперь, мы можем посчитать плотность вероятности **в любой точке**

Для классификации для каждого из класса считаем $p(x \mid y_k, \lambda_k) = \lambda_k e^{-\lambda_k x}$

Потом применяем оптимальное байесовское правило:

$$a^*(x) = \underset{k}{\operatorname{argmax}}\, p(x \mid y_k) p(y_k) = \underset{k}{\operatorname{argmax}}\, \lambda_k e^{-\lambda_k x} p(y_k)$$

ММП для нормального распределения

$$L(\mu,\sigma) = p(X\,|\,\mu,\sigma) = \prod_{i=1}^{N} p(x_i\,|\,\mu,\sigma), \text{ где } p(x_i\,|\,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x_i-\mu)^2}{2\sigma^2})$$

$$\log L(\mu,\sigma) = \sum_{i=1}^{N} \log p(x_i\,|\,\mu,\sigma) = -N\log\sigma - N\log\sqrt{2\pi} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2$$

$$\frac{\partial \log L(\mu,\sigma)}{\partial\mu} = -\frac{1}{\sigma^2}\sum_{i=1}^{N}(x_i-\mu) = N\mu - \frac{1}{\sigma^2}\sum_{i=1}^{N}(x_i) = 0$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\frac{\partial \log L(\mu,\sigma)}{\partial\sigma} = -\frac{N}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{N}(x_i-\mu)^2 = 0$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \frac{1}{N}\sum_{i=1}^{N} x_i)^2$$

Потом аналогично $a^*(x) = \underset{k}{\operatorname{argmax}}\, p(y_k)\frac{1}{\sqrt{2\pi}\sigma_k}\exp(-\frac{(x-\mu_k)^2}{2\sigma_k^2})$

Пример

Проводим серию испытаний — подбрасываем **нечестную** монетку.

Случайная величина $y = 1$ — орел, $y = 0$ — решка.

?

Подбросили N раз. Как оценить вероятность выпадения орла?

Пример

$$L(\theta) = p(Y|\theta) = \prod_{i=1}^{N} p(y_i|\theta), \text{ где}$$

$$p(y_i|\theta) = \theta, \quad \text{если } y_i = 1$$

$$p(y_i|\theta) = 1 - \theta, \text{ если } y_i = 0$$

Так как $[y_i = 1] \equiv y$ и $[y_i = 0] \equiv 1 - y$

$$L(\theta) = \prod_{i=1}^{N} \theta^{[y_i=1]} \cdot (1-\theta)^{[y_i=0]} = \prod_{i=1}^{N} \theta^{y} \cdot (1-\theta)^{1-y}$$

$$\log L(\theta) = \sum_{i=1}^{N} y_i \log\theta + (1-y_i)\log(1-\theta)$$

? Ничего не напоминает?

Пример

Это же логлосс!

$$\log L(\theta) = \sum_{i=1}^{N} y_i \log\theta + (1 - y_i)\log(1 - \theta)$$

$$\frac{\partial \log L(\theta)}{\partial\theta} = \sum_{i=1}^{N} \frac{y_i}{\theta} - \frac{1 - y_i}{1 - \theta} = \sum_{i=1}^{N} \frac{y_i - \theta y_i - \theta + \theta y_i}{\theta(1 - \theta)} = 0$$

$$\sum_{i=1}^{N} (y_i - \theta) = \sum_{i=1}^{N} y_i - N\theta = 0$$

$\theta = \frac{\sum_{i=1}^{N} y_i}{N}$, то есть просто процент орлов!

Таким образом восстанавливаем плотность для булевых признаков.

$$a^*(x) = \operatorname*{argmax}_k p(y_k)\theta^{[k=1]}(1 - \theta)^{[k=0]}$$

А многомерно для всех фичей так можно?

Конечно!

- Просто берем не одномерную плотность, а многомерную и точно так же с помощью ММП находим параметры.
- Или используем предположение алгоритма наивного байеса $p(x \mid y) = \prod_{j=1}^{D} p(x_j \mid y)$

? Будут ли проблемы при параметрическом восстановлении многомерной плотности?

И тут проблема с большим число признаков

Да, будут.

Существуют гарантии на оптимальность метода максимального правдоподобия только для случая, когда $\frac{n}{d} \to \infty$

Если это не так, он может не сойтись к истинному значению, а в многомерных распределениях параметров больше.

Наивный байесовский классификатор

$$a^*(x) = \underset{y \in Y}{\operatorname{argmax}}\, p(y \mid x) = \underset{y \in Y}{\operatorname{argmax}}\, p(x \mid y) p(y)$$

$$p(x \mid y) = \prod_{j=1}^{D} p(x_j \mid y)$$

$$a^*(x) = \underset{y \in Y}{\operatorname{argmax}}\, p(y) \prod_{j=1}^{D} p(x_j \mid y)$$

Это все маленькие числа, когда мы перемножим много маленьких, то есть шанс выйти за машинную точность.

Но логарифм монотонная функция, так что прологарифмируем.

$$a^*(x) = \underset{y \in Y}{\operatorname{argmax}} [\log p(y) + \sum_{j=1}^{D} \log p(x_j \mid y)]$$

Теперь мы складываем отрицательные числа, что для машины сильно проще.

Наивный байесовский классификатор

$$a^*(x) = \underset{y \in Y}{\operatorname{argmax}}\, p(y) \prod_{j=1}^{D} p(x_j \mid y)$$Делаем классификатор спам / не спам.

Как признаки берем D бинарных признаков «имеется ли это слово в письме»

На обучении для каждой бинарной фичи считаем плотность, как для распределения Бернулли:

$p(x_j \mid y)$ — процент спамовских (класс S) / нормальных писем со словом x_j

Хотим посчитать предсказание для тестового письма.

? Чему будет равно $p(y_{spam}) \prod_{j=1}^{D} p(x_j \mid y_{spam})$, если хотя бы одно слово из тестового письма не присутствовала в письмах со спамом?

Сглаживание

Какими бы не были спамовскими другими слова, общая вероятность $p(x \mid y_{spam})$ при гипотезе условной независимости равна нулю!

При восстановлении плотности для категориальных признаков нужно проводить **сглаживание**

$$\theta_j = \frac{\sum_{i=1}^{N} [y_i \in S][x_{i,j} = 1]}{N_s}$$ — доля спамовских документов встречалось слово j

$$\theta_j = \frac{\sum_{i=1}^{N} [y_i \in S][x_{i,j} = 1] + \alpha}{N_s + \alpha + \beta}$$ — **сглаживание Лапласа**

Считаем, что есть фейковые $\alpha + \beta$ документов, среди которых α спамовские

Почему вообще нужно сглаживание? Если в классе спама не было такого слова, значит вероятность 0 и все!

Потому что переобучение

Вспоминаем исходную задачу. Восстановление закона природы по **конечной** выборке. Задача по своей природе некорректно поставленная.

Ну вот не попалось нам на обучении спамовское письмо со словом «серобуромалиновый», но это не значит, что спамеры сознательно избегают этого слова.

Сглаживание — способ борьбы с переобучением, то есть своеобразная **регуляризация**

? Можно ли подобрать α, β на обучающей выборке?

Пожалуйста, не надо

Нет, это **структурные параметры** алгоритма.

Вы уже на обучающей выборке посчитали, что слово «серобуромалиновый» **не встречается** в классе спам, понятное дело, если вы зарегуляризуете это с помощью α, β, то на обучении будет только хуже.

Настраивать только на валидации!

Совмещаем все вместе

Байесовский классификатор, основанный на оценке плотности.

1. Разбили обучающую выборку по классам
2. Оценили $p(y)$, как долю данного класса в обучающей выборке для каждого класса
3. Решили, как будем считать $p(x \mid y)$

A. Непараметрически

B. Параметрически

- В лоб восстанавливаем многомерную плотность
- Используем предположение условной независимости

4. Посчитали $p(x \mid y)$ и умножили на $p(y)$ для каждого класса
5. Взяли класс с максимальным значением

$$a^*(x) = \underset{y \in Y}{\operatorname{argmax}}\, p(y \mid x) = \underset{y \in Y}{\operatorname{argmax}}\, p(x \mid y)p(y)$$

Резюме второй части

- Научились оценивать компоненты для оптимального байесовского классификатора

A. $p(y)$ — априорную вероятность класса

B. $p(x \mid y)$ — правдоподобие объекта в классе

- Вспомнили, что такое метод максимума правдоподобия, научились им оценивать параметры распределений
- Вспомнили, что такое логлосс
- Узнали, что такое сглаживание и как оно помогает с переобучением

Часть 3
Байесовский подход в машинном обучении

Байесовский подход

- Сейчас одна из самых быстроразвивающихся областей машинного обучения
- Позволяет получить важные практические результаты (например, вараиционный дропаут)
- Все параметры модели описываются вероятностными распределениями таким образом
- Использование формулы Байеса для оценки параметров θ нашей модели
- Расчет $p(y \mid x, \theta)$ для оптимального предсказания

Зачем нужна случайность

Мы хотим открыть закон природы, но мы работаем с эмпирическими данными!

- В данных могут быть аномальные объекты
- Мы не полностью описываем явление, выбирая какие-то признаки.
- В признаках будут ошибки
- Наша модель не полностью описывает данные

Это все объективное незнание! Закон природы есть, но мы не можем записать его точно.

Мы все это можем задать с помощью вероятностных распределений.

Смысл формулы Байеса

Оцениваем параметры модели.

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)}$$

$p(\theta)$ — наши априорные представления о параметрах

Провели эксперименты (собрали выборку), посчитали:

$p(X|\theta)$ — насколько вероятно получить такие результаты экспериментов?

$p(X)$ — насколько вообще наши исходы вероятны?

Обновили наши представления о параметрах $p(\theta|X)$ по формуле Байеса

?

А если провели еще серию экспериментов?

Если провели еще эксперименты

Возьмите прошлую апостериорную вероятность, как априорную и снова пересчитайте по формуле Байеса!

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)} = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta}$$

Таким образом мы вообще не теряем информации, просто итеративно обновляем распределение на параметры по формуле Байеса.

Очень удобно, когда, применив формула Байеса, мы остались в том же классе распределений, какое было априорным. Такие распределения называют **сопряженным**.

ММП vs Bayes

Подбрасываем монетку. Хотим посчитать вероятность выпадения орла. ММП дает ответ:

$$\theta = \frac{\sum_{i=1}^{N} y_i}{N}$$

? А если мы подбросили 3 раза, и 3 раза выпала решка, мы считаем, что монета орлом вообще не может выпасть?

ММП vs Bayes

Мы провели очень мало испытаний, метод максимума правдоподобия может дать неадекватную оценку

Свойства ММП:

- Состоятельность (оценка сходится по вероятности к истинному значению)
- Асимптотическая нормальность (оценка распределена нормально)
- Асимптотическая эффективность (обладает наименьшей дисперсией среди всех состоятельных асимптотически нормальных оценок)

При $n \to \infty$

?

У нас тут BigData. И зачем все это?

У нас много параметров

Современные нейронные сети имеют миллионы параметров. Каждый из них надо оценить!

Тогда нужно, чтобы:

$\frac{n}{d} \to \infty$, а это даже для BigData неправда.

Но если параметров мало, а данных много, то байесовский подход **переходит в ММП**: апостериорное распределение ведет себе как дельта функция в точке максимума правдоподобия.

Пример

Подбрасываем монетку. Хотим посчитать вероятность выпадения орла. ММП дает ответ:

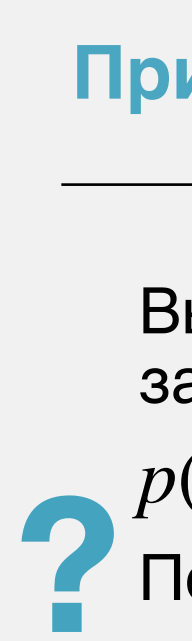
$$\theta_{ML} = \frac{\sum_{i=1}^{N} y_i}{N}$$

Хотим посчитать оценку формулой Байеса:

$$p(\theta \mid Y) = \frac{p(Y \mid \theta) p(\theta)}{p(Y)}$$

$$\theta_{MAP} = \operatorname{argmax} p(\theta \mid Y) = \operatorname{argmax} p(Y \mid \theta) p(\theta)$$

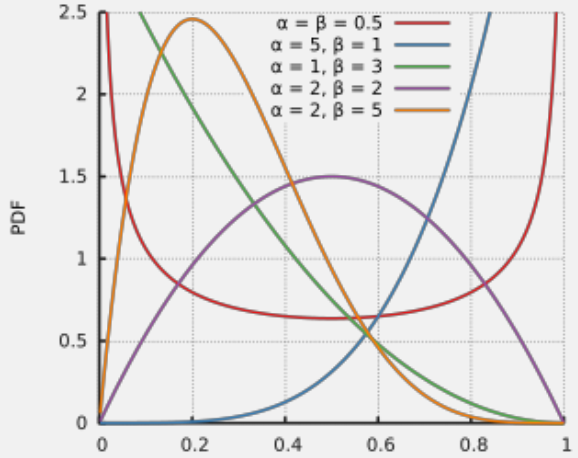
Пример



Выбираем априорное распределение (как мы заранее считаем, как часто она выпадет орлом)

$p(\theta) = C \cdot \theta^{\alpha-1}(1-\theta)^{\beta-1}$ — Бета распределение

Почему именно это?



Источник: en.wikipedia.org/wiki/Beta_distribution

Пример

Потому что оно сопряжено с Бернулли:

$$p(\theta)p(Y\,|\,\theta) = C \cdot \theta^{\alpha-1}(1-\theta)^{\beta-1}\theta^{n}(1-\theta)^{N-n} = C \cdot \theta^{\alpha-1+n}(1-\theta)^{\beta-1+N-n}$$

$$\log p(Y\,|\,\theta) = \sum_{i=1}^{N} y_i \log\theta + (1-y_i)\log(1-\theta)$$

$$\log p(\theta) = \log C + (\alpha-1)\log\theta + (\beta-1)\log(1-\theta)$$

$$\theta_{MAP} = \operatorname{argmax} p(Y\,|\,\theta)p(\theta) = \operatorname{argmax} \log p(Y\,|\,\theta) + \log p(\theta)$$

$$\sum_{i=1}^{N} y_i \log\theta + (1-y_i)\log(1-\theta) + (\alpha-1)\log\theta + (\beta-1)\log(1-\theta)$$

Производная по θ

$$\sum_{i=1}^{N} \frac{y_i}{\theta} - \frac{1-y_i}{1-\theta} + \frac{\alpha-1}{\theta} - \frac{\beta-1}{1-\theta}$$

$$\theta_{MAP} = \frac{\sum_{i=1}^{N} y_i + \alpha - 1}{N + \alpha + \beta - 2}$$

?

Ничего не напоминает?

Это и есть сглаживание

$$\theta_{ML} = \frac{\sum_{i=1}^{N} y_i}{N}, \theta_{MAP} = \frac{\sum_{i=1}^{N} y_i + \alpha - 1}{N + \alpha + \beta - 2}$$

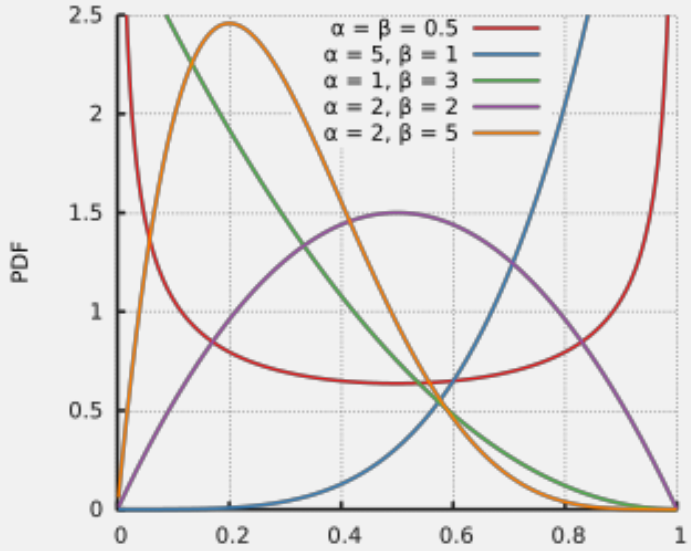
α, β — наши фейковые бросания!

Если взять $\alpha = 2, \beta = 2$

$$\theta_{MAP} = \frac{\sum_{i=1}^{N} y_i + 1}{N + 2}$$

Таким образом, Байесовский подход нам позволяет напрямую **влиять на решение**, выбирая prior. Это **уменьшает переобучение**.

Это и есть сглаживание



Чем больше α, тем больше мы доверяем, что монетка падает орлом

Чем больше β, тем больше мы доверяем, что монетка падает решкой

Логистическая регрессия

Подбрасываем монетку, получаем серию из единиц и нулей.
Правдоподобие:

$$L(\theta) = \prod_{i=1}^{N} p(y_i | x, \theta)$$

Когда моделировали монетку, считали что:

$$p(y_i | x, \theta) = \theta, \qquad \text{если } y_i = 1$$

$$p(y_i | x, \theta) = 1 - \theta, \text{ если } y_i = 0$$

То есть каждый бросок мы раньше никак не описывали, считали их все равнозначными.

Но мы то знаем, что у броска есть **признаки,** которые описывают бросок. Например, начальная скорость, стартовый угол, угловая скорость и т.д.

Тут то мы и переходим к задаче двухклассовой классификации!

Логистическая регрессия

Двухклассовая классификация.

Понаблюдали **обучающую выборку** из единиц и нулей, правдоподобие выборки:

$$\prod_{i=1}^{N} p(y|x,\theta)^{[y_i=1]} \cdot (1 - p(y|x,\theta))^{[y_i=0]} =$$

$$\prod_{i=1}^{N} p(y|x,\theta)^{y} \cdot (1 - p(y|x,\theta)^{1-y}$$

$$\log L(\theta) = \sum_{i=1}^{N} y_i \log p(y|x,\theta) + (1 - y_i)\log(1 - p(y|x,\theta))$$

Используем сигмоид, чтобы предки перевести в вероятность:

$$p(y|x,\theta) = \frac{1}{1 + exp(-a(x,\theta))}$$

Если $a(x,\theta) = x^T\theta,$ то получаем логистическую регрессию!

$\theta_{ML} = \text{argmax} \log L(\theta)$, находим оптимум градиентным спуском

Регуляризация

Хотим делать не ММП, а MAP, чтобы уменьшить переобучение.

$$\theta_{MAP} = \text{argmax} L(\theta)p(\theta) = \text{argmax} \log L(\theta) + \log p(\theta)$$

Пусть каждая компонента $p(\theta)$ распределена нормально с нулевым средним и все компоненты независимы и имеют одинаковую дисперсию σ^2:

$$p(\theta) = \prod_{j=1}^{D} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\theta_j^2}{2\sigma^2})$$

$$\theta_{MAP} = \text{argmax} \sum_{i=1}^{N} y_i \log p(y|x,\theta) + (1-y_i)\log(1-p(y|x,\theta)) - \sum_{j=1}^{D} C\frac{\theta_j^2}{\sigma^2}$$

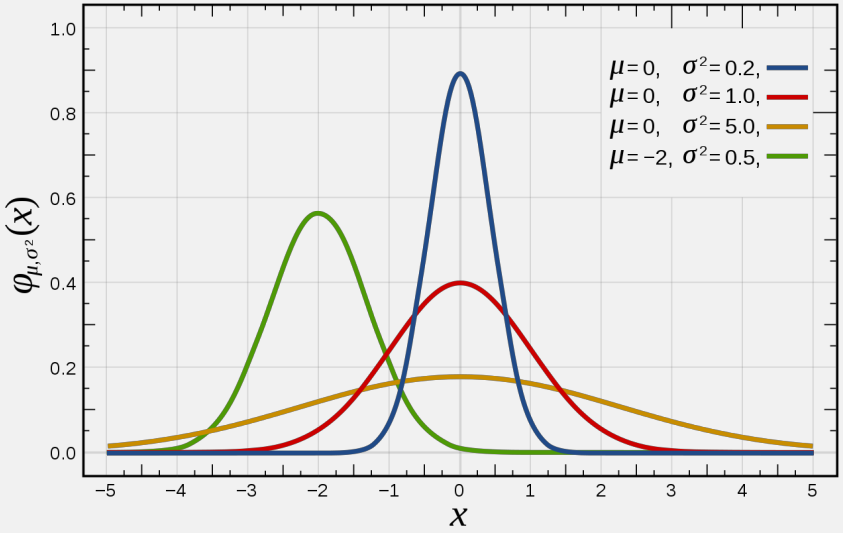
$$\theta_{MAP} = \text{argmax} \sum_{i=1}^{N} y_i \log p(y|x,\theta) + (1-y_i)\log(1-p(y|x,\theta)) - \frac{C}{\sigma^2}.||\theta||^2$$

Берем с минусом, получаем loglocc с l2 регуляризацией!

Какой смысл у дисперсии σ^2?

Регуляризация

Мы хотим, чтобы наши веса были маленькими, потому что большие веса — источник переобучения. Чем меньше сигма, тем уже нормальное распределение, то есть тем сильнее мы регуляризуем



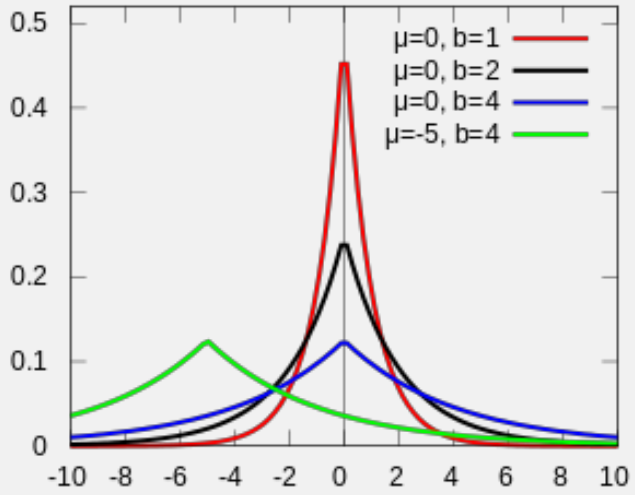
А как тогда получить l1 регуляризацию?

L1 регуляризация

Проделайте дома аналогичную процедуру для распределения Лапласа

$$p(\theta) = \prod_{j=1}^{D} \frac{1}{2b} exp(-\frac{|\theta|}{b})$$

Чем меньше b, тем уже распределение Лапласа, тем больше регуляризация



Источник:en.wikipedia.org/wiki/Laplace_distribution

Линейная регрессия

Для классификации мы считали правдоподобие распределения Бернулли, потому что наблюдали выборку из единиц и нулей. Для регрессии мы наблюдаем вещественное значение.

Предположим, что мы нашей моделью $a(x, \theta)$ можем восстанавливаем нашу переменную с точностью до нормального шума, то есть **каждый** y

Распределен нормально со средним в точке $a(x, \theta)$ и с какой-то дисперсией σ^2

$$p(y \mid x, \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y - a(x, \theta))^2}{2\sigma^2})$$

Правдоподобие:

$$L(\theta) = \prod_{i=1}^{N} p(y_i \mid x_i, \theta) = \frac{1}{\sqrt{2\pi}\sigma} \prod_{i=1}^{N} \exp(-\frac{(y - a(x, \theta))^2}{2\sigma^2})$$

$$\log L(\theta) = C - \sum_{i=1}^{N} (y - a(x, \theta))^2$$

Говорим, что $a(x, \theta) = x^T\theta$ — вот и линейная регрессия

$\theta_{ML} = \operatorname{argmax} \log L(\theta)$, находим оптимум аналитически или градиентным спуском

Как добавить сюда регуляризацию?

Регуляризация аналогично

Добавляем наши априорные знания на веса, переходим оценки ММП к MAP оценке.

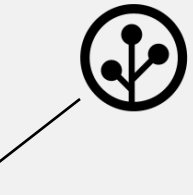
$$\theta_{MAP} = \text{argmax} L(\theta) p(\theta) = \text{argmax} \log L(\theta) + \log p(\theta)$$

Если $p(\theta)$ нормально, то будет L2

Если $p(\theta)$ по Лапласу, то будет L1

Дома проведите выкладки!

Предсказание

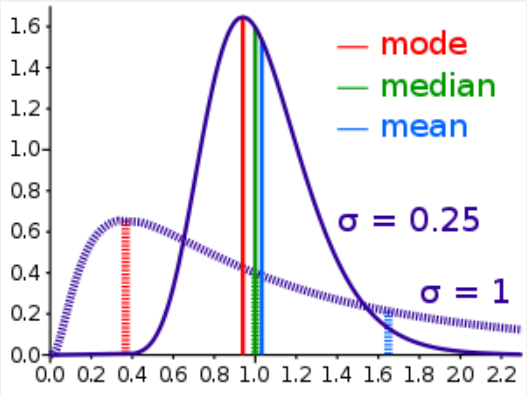


Пусть мы полностью выполнили Байесовскую процедуру:

$$p(\theta \mid X) = \frac{p(X \mid \theta)p(\theta)}{p(X)} = \frac{p(X \mid \theta)p(\theta)}{\int p(X \mid \theta)p(\theta)d\theta}$$

Как сделать предсказание?

Наивный вариант: взять моду этого распределения θ_{MAP} и подставить в формулу модели $p(y \mid x, \theta)$. Но мода распределения не всегда лучший вариант!



Источник:

en.wikipedia.org/wiki/Mode_(statistics)

Предсказание

Чтобы не терять информацию, мы можем честно посчитать распределение $p(y \mid x)$ (которое требует оптимальный Байесовский классификатор)

$$p(y \mid x, X_{train}) = \int p(y \mid x, \theta) p(\theta \mid X_{train}) d\theta$$

Для линейной регрессии можно посчитать.

Посмотрите внимательно:

Это просто усреднение предсказаний модели с разными параметрами с весом, который равен вероятности параметров!

Это ансамбль моделей.

Байесовские методы на практике

Байесовский подход позволяет решать многие прикладные задачи:

- Метод релевантных векторов
- Вариационный дропаут
- Байесовский автокодировщик
- Байесовская оптимизация
- etc

Однако, в большинстве задач его применить тяжело:

- Никто не знает, как правильно задавать априорное распределение.
- Если prior и likelihood не сопряжены, то аналитически ничего не посчитается.
- Численные методы вычислительно трудоемкие
- Проще собрать больше данных/больше признаков, чем использовать полный байесовский вывод

Резюме третьей части

- Познакомились с байесовским подходом в машинном обучению
- Сравнили байесовский подход с приниципом максимума правдоподобия
- Провели вероятностную интерпретацию для линейных моделей
- Научились делать оптимальное предсказание
- Байесовский подход позволяет решать практические задачи, однако полный байесовский вывод на практике применять очень тяжело

ТЕХНОСФЕРА

Спасибо за внимание!