ТЕХНОСФЕРА

Лекция 10
EM алгоритм

Всеволод Викулин

Содержание

1. Модели со скрытыми переменными
2. EM алгоритм
3. Разделение смеси распределений

Часть 1
Модели со скрытыми переменными

Метод максимума правдоподобия

Пишем функцию правдоподобия, то есть вероятность пронаблюдать выборку. Если все объекты берутся независимо из одинакового распределения:

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

С суммой, как правило, удобнее работать.

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

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

$$\theta_{ML} = \operatorname{argmax} \log L(\theta)$$

Метод максимума правдоподобия

Показали, что для распределения Бернулли:

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

Для нормального:

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

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

Метод максимума правдоподобия

Если 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 $\frac{n}{d} \to \infty$

Для каких распределений мы можем легко посчитать оценку максимума правдоподобия?

Экспоненциальный класс

$$\theta_{ML} = \operatorname{argmax} \log p(X|\theta)$$

$p(x|\theta)$ лежит в экспоненциальном классе, если:

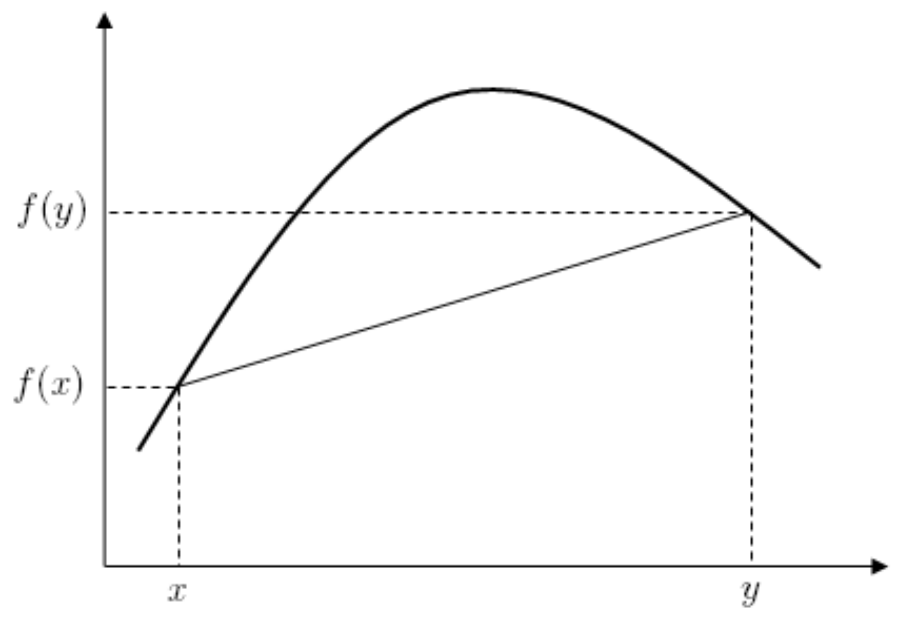
$$p(x|\theta) = \frac{f(x)}{g(\theta)} exp(\theta^T u(x)), \text{ где}$$

$$g(\theta) = \int f(x) exp(\theta^T u(x)) dx$$

? Что мы хотим от $\log p(X|\theta)$, чтобы быть уверенными, что мы сможем посчитать оценки ММП?

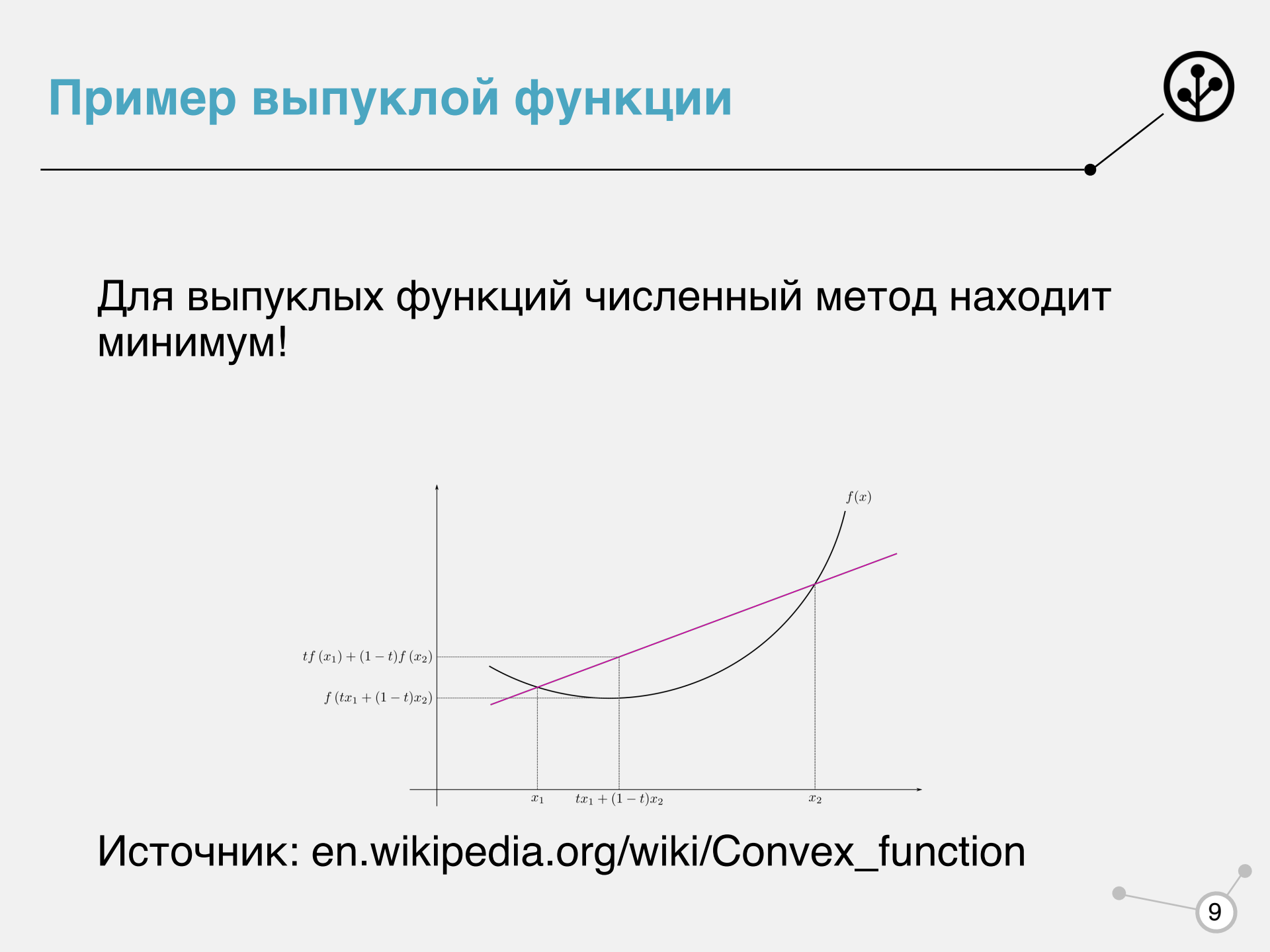
Пример вогнутой функции

Для вогнутых функций численный метод находит максимум!



Источник: en.wikipedia.org/wiki/Concave_function

Пример выпуклой функции



Для выпуклых функций численный метод находит минимум!

Источник: en.wikipedia.org/wiki/Convex_function

Экспоненциальный класс

$$\log L(\theta) = \sum_{i=1}^{N} f(x_i) - N \log g(\theta) + \theta^T \sum_{i=1}^{N} u(x_i)$$

Можно показать, что:

$$\frac{\partial^2 g(\theta)}{\partial \theta_i \partial \theta_j} = cov(u_i, u_j)$$

Ковариационная матрица неотрицательно определена, то есть $\log L(\theta)$ функция **вогнутая**

? Ну и что, наши то любимые распределения ни из какого не экспоненциального класса.

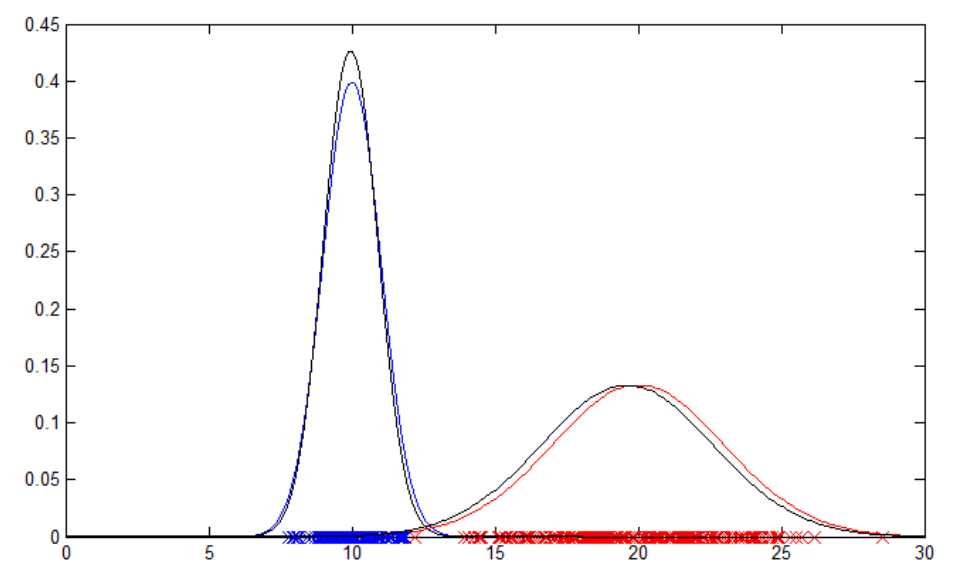
Распределения из экспоненциального класса

Возможно, все распределения, которые вы знаете оттуда

Distribution	Parameter(s) θ
Bernoulli distribution	p
binomial distribution with known number of trials n	p
Poisson distribution	λ
negative binomial distribution with known number of failures r	p
exponential distribution	λ
Pareto distribution with known minimum value x_m	α
Weibull distribution with known shape k	λ
Laplace distribution with known mean μ	b
chi-squared distribution	v
normal distribution known variance	μ

Источник: en.wikipedia.org/wiki/Exponential_family

Пример распределения не из экспоненциального класса



Источник: chrisjmccormick.files.wordpress.com/2014/08/1d_example.png

? Что бы нам дополнительно хотелось знать, чтобы мы могли построить 2 гауссианы?

Какая точка из какой гауссианы

Тогда бы мы рассмотрели **независимо** точки из **каждой** K гауссианы и по формулам из ММП:

$$\mu_k = \frac{\sum_{i=1}^{N} x_i [x_i \in K]}{N_k}$$

$$\sigma_k^2 = \frac{\sum_{i=1}^{N} [x_i \in K](x_i - \mu_k)^2}{N_k}$$

Одна проблемка — мы не знаем из какой гауссианы пришла каждая точка!

Модели со скрытыми переменными

Наша выборка пришла из какого-то сложного распределения, которое зависит не только от наблюдаемых переменных X, но и от каких-то **скрытых** переменных Z, которые мы **не наблюдаем**.

Есть два варианта:

- Не обращаем на это внимание, просто решаем задачу $\operatorname{argmax} \log p(X \mid \theta)$ выбирая какое-то параметрическое семейство
- Смиряемся, что Z мы не наблюдаем, моделируем все равно $\operatorname{argmax} \log p(X, Z \mid \theta)$ и находим одновременно и параметры модели, и скрытые переменные

В чем преимущество

Скрытые переменные мы выбираем сами!

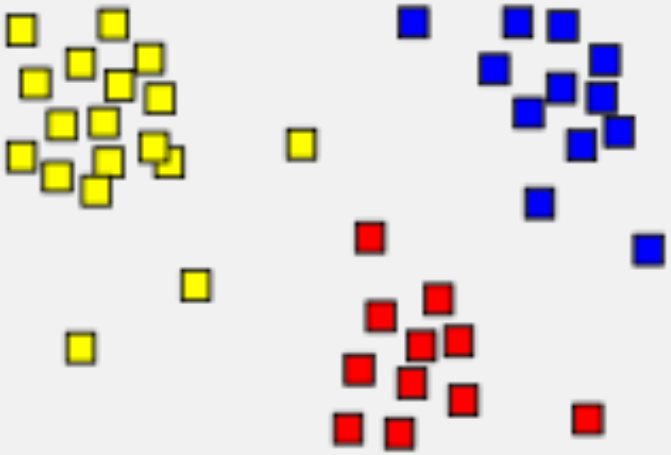
То есть мы можем сделать так, что совместное распределение $p(X, Z|\theta)$ будет из экспоненциального класса, а значит $\log p(X, Z|\theta)$ будет вогнутой функцией и задача

$\underset{\theta}{\operatorname{argmax}} \log p(X, Z|\theta)$ может быть решена эффективно!

- $p(X|\theta)$ — неполное правдоподобие
- $p(X, Z|\theta)$ — полное правдоподобие

Иногда сами параметры не нужны

В некоторых задачах нам на самом деле только скрытые переменные и нужны!



Источник: en.wikipedia.org/wiki/Cluster_analysis

?

В задаче кластеризации какие скрытые переменные?

Примеры задач

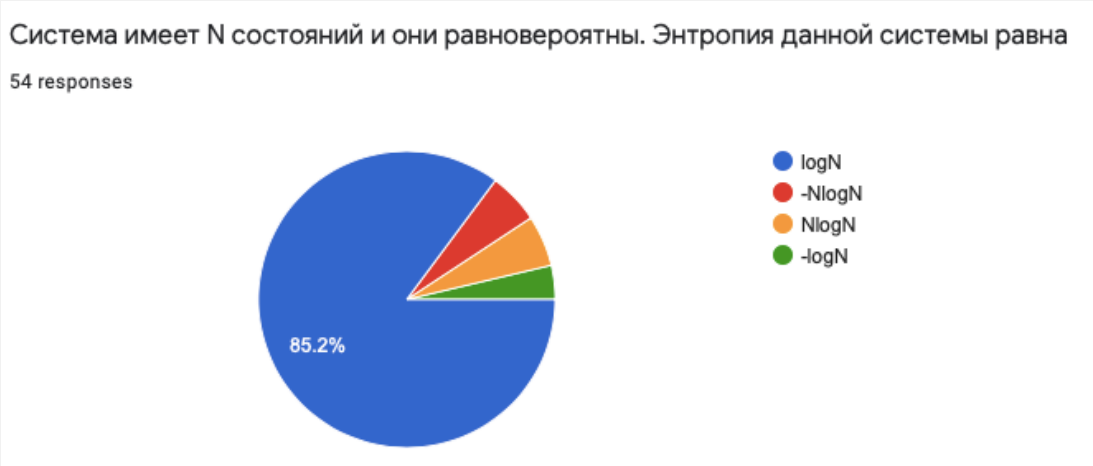
Скрытая переменная — номер кластера! Нам ее и нужно восстановить

Примеры задач со скрытыми переменными:

- Кластеризация
- Машинный перевод
- Распознавание речи
- Распознавание частей речи
- Тематическое моделирование
- Умная разметка в краудсорсинге
- Восстановление плотности

Пример

Провели тест, теперь хотим узнать правильные ответы.



? И какой ответ правильный? Какие в задаче наблюдаемые, а какие скрытые переменные?

Пример

Правильный logN

Наблюдаемые — ответы, которые дают студенты

Скрытые — правильные ответы

Параметры — «оценка студента»

Такие задачи можно решать в 2 шага

На первой итерации всех студентов оцениваем одинаково.

1. Оценили ответы через взвешенную сумму голосов с большими весами у отличников
2. По этим ответам пересчитали оценки студентов

А какое практическое применение? Ответы то мы знаем.

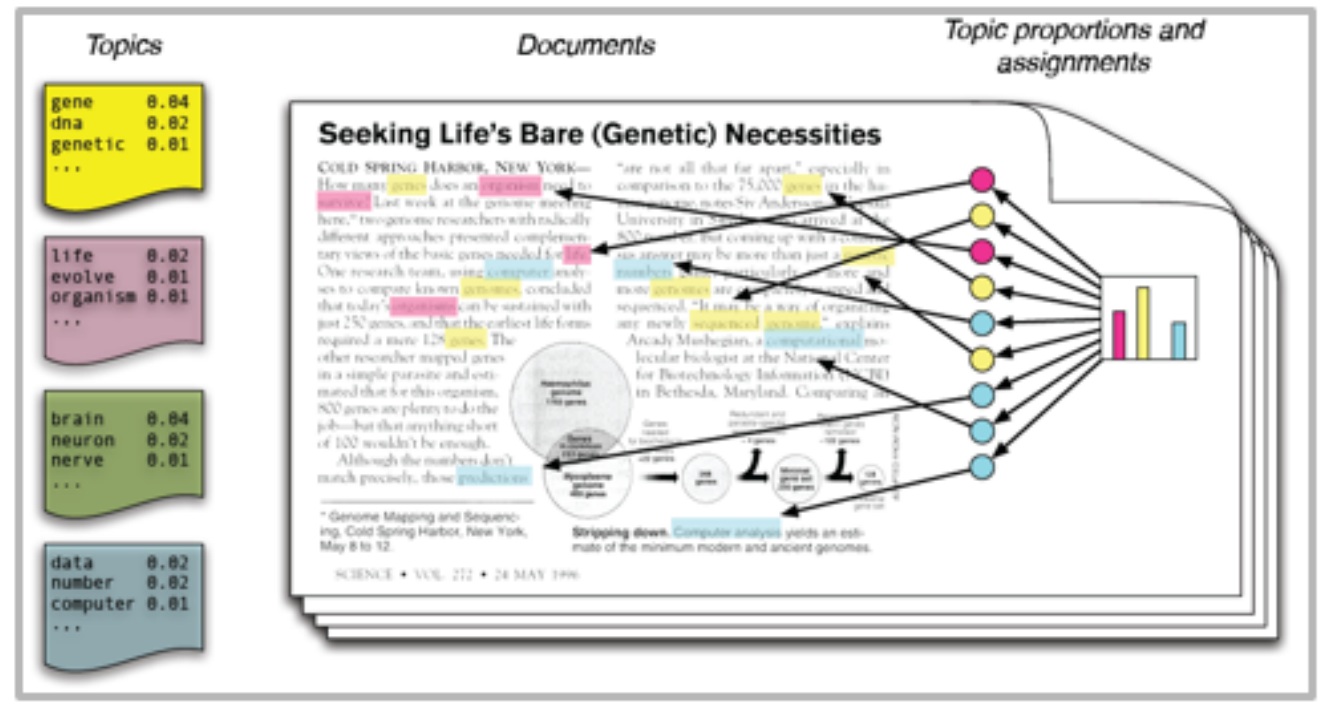
Пример

Для краудсорсинга!

Хотим разметить огромный датасет на кошечек и собачек. Одну картинку оценивают сразу **несколько** людей.

Применяем такой алгоритм, чтобы найти халтурщиков (ставят метки просто так) и восстановить правильные метки!

Тематические модели



Источник: analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python

Тематические модели

Один из современных подходов к анализу текстов.

Тема — семантический кластер текстов

Цель тематического моделирования: понять, к каким темам относится документ и какие слова образуют тему.

Примеры применения:

- Категоризация документов
- Рекомендательные системы
- Информационный поиск
- Выделение трендов в новостных потоках

?

Какие параметры наблюдаемые, а какие скрытые?

Тематические модели

Тема — скрытая переменная. Число тем задаем сами.

Наблюдаем слова в документе.

Настраиваем параметры:

- Φ — матрица вероятности темы для каждого слова
- Θ — матрица вероятности темы для каждого документа

Строим генеративную модель порождения текстов:

$$L(W, Z \mid \Phi, \Theta) = \prod_{d=1}^{D} \prod_{i=1}^{N_d} p(w_{d,i} \mid z_{d,i}, \Phi) p(z_{d,i} \mid \Theta_d)$$

Если на матрицу Θ наложить prior в виде распределения Дирихле, то модель будет называться **latent dirichlet allocation**

Позволяет поощрять разреженность тематического профиля документа

Находим параметры $\log L(W, Z \mid \Phi, \Theta) \to \max_{\Phi,\Theta}$

Тематические модели

```
# Build LDA model
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=20,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha='auto',
                                            per_word_topics=True)
```

```
(17,
 '0.027*"game" + 0.027*"team" + 0.020*"year" + 0.017*"play" + 0.016*"win" + '
 '0.010*"good" + 0.009*"season" + 0.008*"fan" + 0.007*"run" + 0.007*"score"'),
(18,
 '0.036*"drive" + 0.024*"card" + 0.020*"mac" + 0.017*"sale" + 0.014*"cpu" + '
 '0.010*"price" + 0.010*"disk" + 0.010*"board" + 0.010*"pin" + 0.010*"chip"'),
(19,
 '0.030*"space" + 0.010*"sphere" + 0.010*"earth" + 0.009*"item" + '
 '0.008*"launch" + 0.007*"moon" + 0.007*"mission" + 0.007*"nasa" + '
 '0.007*"orbit" + 0.006*"research"')]
```

Источник: machinelearningplus.com/nlp/topic-modeling-gensim-python/

Резюме первой части

- Узнали, что такое экспоненциальный класс распределений, почему для него просто считать оценку ММП
- Познакомились с формулировкой задачи со скрытыми переменными
- Разобрали несколько примеров таких задач

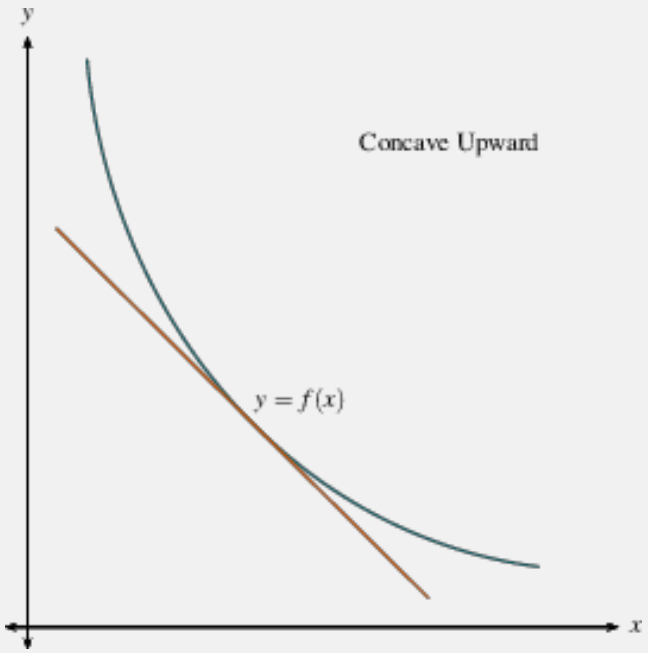
Вариационная нижняя оценка

Вариационной нижней оценки для функции $f(\theta)$ является функция $g(\theta, \nu)$, если

для любых θ, ν $g(\theta, \nu) \leq f(\theta)$ и для любого θ^* найдется μ^* такой, что $g(\theta^*, \nu^*) = f(\theta^*)$

То есть мы пытаемся снизу подпереть нашу функцию каким-то классом функций!

Пример



Источник: web.ma.utexas.edu/users/m408n/m408c/CurrentWeb/LM4-3-8.php

Вариационная нижняя оценка

Хотим решить задачу $\operatorname{argmax}_{\theta} f(\theta)$

Напрямую решить не получается, но мы смогли построить вариационную нижнюю оценку $g(\theta, \nu)$, для которой максимум по θ искать просто.

Тогда для решения задачи $\operatorname{argmax}_{\theta} f(\theta)$ можно итеративно находить максимумы для $g(\theta, \nu)$

1. Инициализировали θ_{old}, посчитали для него $\nu_{old} = \operatorname{argmax}_{\nu} g(\theta_{old}, \nu) = f(\theta_{old})$
2. $\theta_{new} = \operatorname{argmax}_{\theta} g(\theta, \nu_{old}), \nu_{new} = \operatorname{argmax}_{\nu} g(\theta_{new}, \nu)$

? Сходится ли такой метод?

Сходится ли метод

Да!

$$g(\theta_{old}, \mu_{old}) = f(\theta_{old}) < g(\theta_{new}, \mu_{old}) < g(\theta_{new}, \mu_{new}) = f(\theta_{new})$$

На каждой итерации мы увеличиваем значение нашей функции $f(\theta)$, но глобального оптимума никто не гарантирует — сходимся к локальному.

При чем тут машинное обучение? За что нам это?

EM алгоритм

$f(\theta)$ в нашей задачи это неполное правдоподобие

Наша задача построить для нее $g(\theta, \nu)$, которое бы являлась вариационной нижней оценкой. Очевидно, что в нее в каком-то виде войдут скрытые переменные.

EM (Expectation-maximization) - алгоритм, позволяющий находить оценку максимального правдоподобия в задачах со скрытыми переменными.

Схематично метод можно описать так:

1. Проинициализировали параметры
2. Максимизировать **по скрытым** переменным (E шаг)
3. Максимизировать **по параметрам** (M шаг)
4. Повторять 2 и 3 до сходимости

Напоминание из теории информации

Есть случайная величина. Какое число информации $h(x)$ мы получаем, когда наблюдаем определенную ее реализацию?

Можно показать, что $h(x) = -log_2 p(x)$

Получатель отправляет информацию о том, какая у него реализация случайной величины

Сколько в среднем бит ему надо передать?

$$H[x] = -\sum_x p(x) log_2 p(x)$$

Эта величина называется **энтропией** случайной величины x

?

Почему ее используют для дерева решений?

Напоминание из теории информации

Потому что она максимальна у равномерного распределения и равна нулю у вырожденного (когда значение принимается с вероятностью 1)

Для вещественных величин

$$H[x] = -\int p(x)\log p(x)dx$$

Для нормального распределения:

$$H[x] = \frac{1}{2}(1 + \log 2\pi\sigma^2)$$

Чем более «размазанное» распределение, тем больше у него энтропия!

Дивергенция Кульбака-Лейблера

Представим, что мы не знаем $p(x)$ и пытаемся восстановить его каким-то $q(x)$. Как померить **расстояние** между этими двумя распределениями, насколько мы хорошо приблизили?

Будем считать, насколько бит увеличится информация, если мы будем кодировать случайную величины из $p(x)$ с помощью $q(x)$

$$KL(q||p) = -\int p(x)\log q(x)dx - (-\int p(x)\log p(x)dx)$$

$$KL(q||p) = -\int p(x)\log \frac{q(x)}{p(x)}dx = \int p(x)\log \frac{p(x)}{q(x)}dx$$

? Это метрика?

Дивергенция Кульбака-Лейблера

Это не метрика!

Она не удовлетворяет ни условию симметричности, ни неравенству треугольника.

Но она **неотрицательна.** Докажем.

Неравенство Йенсена для выпуклых функций

$$\sum \lambda_i f(x_i) \geq f(\sum \lambda_i x_i)$$

$$KL(q||p) = -\int p(x) \log \frac{q(x)}{p(x)} dx \geq -\log \int p(x) \frac{q(x)}{p(x)} dx = -\log \int q(x) dx = 0$$

Равенство, когда $p(x) = q(x)$ в каждой точке.

То есть можно использовать, как функцию расстояния.

Дивергенция Кульбака-Лейблера

Пусть мы пытаемся восстановить вырожденное распределение для одного объекта

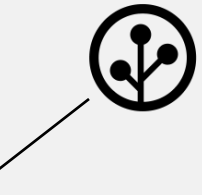
$$KL(q||p) = \sum p(y)\log\frac{p(y)}{q(y)}dx = \sum p(y)\log p(y) - \sum p(y)\log q(y) = -\sum p(y)\log q(y)$$

Будем считать что у объекта 2 класса:

$$-\sum p(y)\log q(y) = -[y=1]\log q(y=1) - [y=0]\log q(y=0)$$

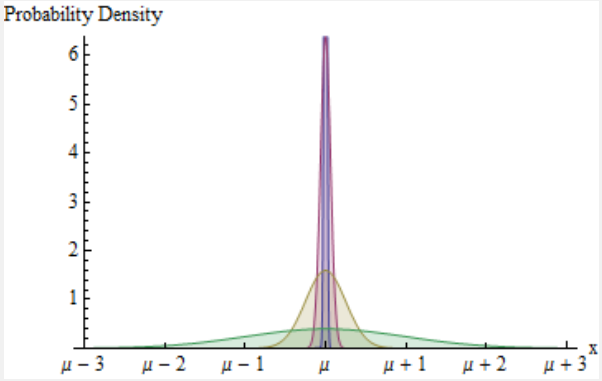
? Ничего не напоминает?

Дивергенция Кульбака-Лейблера



Logloss это просто дивергенция Кульбака-Лейблера.

Мы пытаемся восстановить для каждого объекта распредели над его классами такое, чтобы оно было максимально похоже на истинное врожденное распределение (то есть на такое, когда единственный класс имеет вероятность 1, а остальные классы 0)



Источник:

stats.stackexchange.com/questions/116517/degenerate-univariate-gaussian

Вывод EM алгоритма

Хотим максимизировать $\log P(X|\theta) \to \max\limits_{\theta}$

Пусть Z - скрытые переменные в нашей задаче

$$\log P(X|\theta) = \int q(Z) \log P(X|\theta) dZ = \int q(z) \log \frac{P(X,Z|\theta)}{P(Z|X,\theta)} dZ$$

(так как $P(X|\theta) = \frac{P(X,Z|\theta)}{P(Z|X,\theta)}$)

$$\log P(X|\theta) = \int q(z) \log \frac{P(X,Z|\theta) q(Z)}{P(Z|X,\theta) q(Z)} dZ =$$

$$= \int q(Z) \log \frac{P(X,Z|\theta)}{q(Z)} dZ + \int q(Z) \log \frac{q(Z)}{P(Z|X,\theta)} dZ$$

Где тут спряталась KL дивергенция?

Вывод EM алгоритма

Вот $KL(q||p) = \int q(Z)\log\frac{q(Z)}{P(Z|X,\theta)}dZ$

Первое слагаемое

$L(\theta, q) = \int q(Z)\log\frac{P(X,Z|\theta)}{q(Z)}dZ$ не дивергенция,

так как разный носитель!

Мы доказали, что $KL(q||p)$ неотрицательна.

$\log P(X|\theta) = L(\theta, q) + KL(q||p) \geq L(\theta, q)$

Для любого θ существует q, такое что неравенство перейдет в равенство (когда $KL(q||p) = 0$)

? Ну и что же дальше?

Вывод EM алгоритма

Это значит, что $L(q,\theta) = \int q(Z)\log\frac{P(X,Z|\theta)}{q(Z)}dZ$

является **вариационной нижней оценкой** для $f(\theta) = \log P(X|\theta)$, которую мы оптимизируем

Таким образом для решения $\log P(X|\theta) \to \max_{\theta}$

делаем итеративную оптимизацию $L(\theta, q)$

$\log L(\theta^*, q) \to \max_{q}$ (E шаг, expectation)

$\log L(\theta, q^*) \to \max_{\theta}$ (M шаг, maximization)

E шаг

Хотим оптимизировать нижнюю оценку по q

$$\log L(\theta^*, q) \to \max_q$$

q — это, вообще говоря, распределение. То есть нам нужно провести функциональную оптимизацию. Задача в общем случае непосильная, если бы не:

$$f(\theta) = \log P(X|\theta) = L(\theta, q) + KL(q||p)$$

То есть сумма от q не зависит! Значит, чтобы максимизировать $L(\theta, q)$ по q, надо минимизировать $KL(q||p)$ по q, ну а это то уже мы знаем как

?

Как минимизировать $KL(q||p)$ по q?

E шаг

$$KL(q||p) = \int q(Z)\log\frac{q(Z)}{P(Z|X,\theta)}dZ$$

$$\log KL(q||p) \to \min_{q}$$

По свойству KL дивергенции

$$q^*(Z) = P(Z|X,\theta)$$

Если аналитически вывод провести нельзя, то параметризуем наше q какими-то параметрами $q(\eta)$ и решаем задачу $\log KL(q(\eta)||p) \to \min_{\eta}$

M шаг

Хотим оптимизировать нижнюю оценку по θ

$$\log L(\theta, q^*) \to \max_\theta$$

$$L(\theta, q^*) = \int q^*(Z) \log \frac{P(X, Z \mid \theta)}{q(Z)} dZ = \int q^*(Z) \log P(X, Z \mid \theta) dZ - \int q(Z) \log q^*(Z) dZ$$

Второй член от θ не зависит, его можно отбросить

$$\int q^*(Z) \log P(X, Z \mid \theta) dZ \to \max_\theta$$

То есть на M шаге решаем задачу:

$$\mathop{\mathbb{E}}_Z \log P(X, Z \mid \theta) \to \max_\theta$$

?

Вместо исходной $\log P(X \mid \theta) \to \max_\theta$ стали решать

$\mathop{\mathbb{E}}_Z \log P(X, Z \mid \theta) \to \max_\theta$, в чем смысл?

M шаг

Смысл есть! Мы говорили, что $P(X|\theta)$ не лежит в экспоненциальном классе.

Мы можем подобрать скрытые переменные так, что $P(X,Z|\theta)$ уже будет в экспоненциальном классе, а значит $\log P(X,Z|\theta)$ будет вогнутой функции, а значит, что и $\mathop{\mathbb{E}}_{Z} \log P(X,Z|\theta)$ будет вогнутым!

Интеграл можем даже не считать аналитически, а например использовать методы Монте-Карло.

EM алгоритм

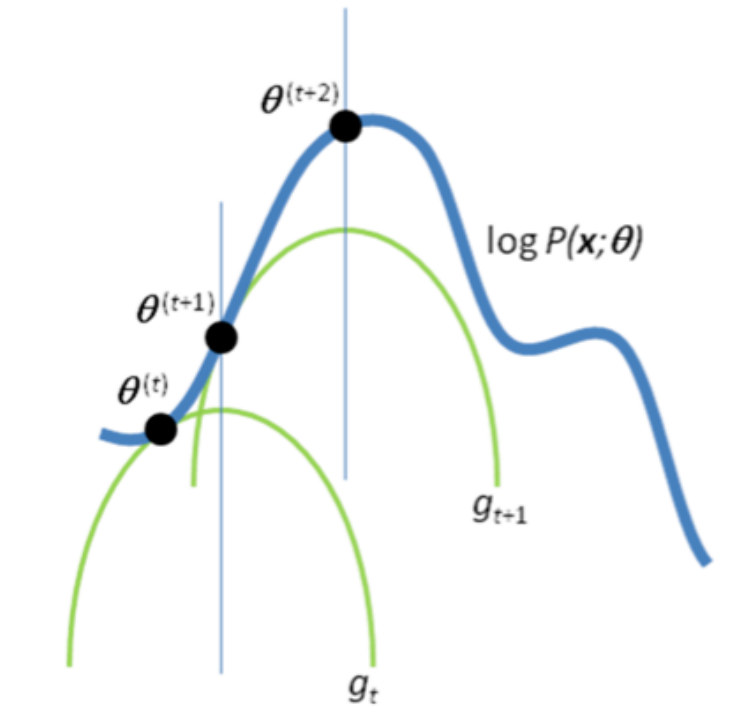
Инициализировали параметры θ_0

1. Оценили апостериорное распределение на скрытые переменные с весами с прошлого шага $q^*(Z) = P(Z|X, \theta_{old})$ (E шаг)
2. Нашли новые веса (M шаг)

$$\int q^*(Z) \log P(X, Z|\theta) dZ \to \max_{\theta}$$

3. Повторять до сходимости весов

EM алгоритм



Источник:

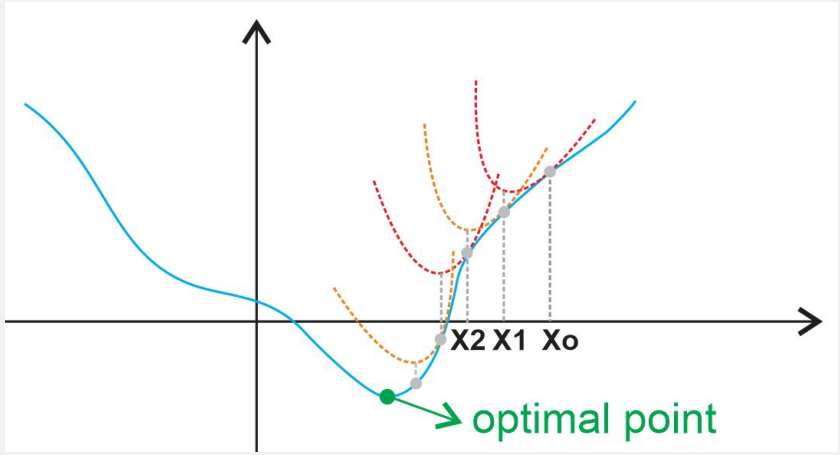
people.duke.edu/~ccc14/sta-663/EMAlgorithm.html

?

На какой метод оптимизации похоже?

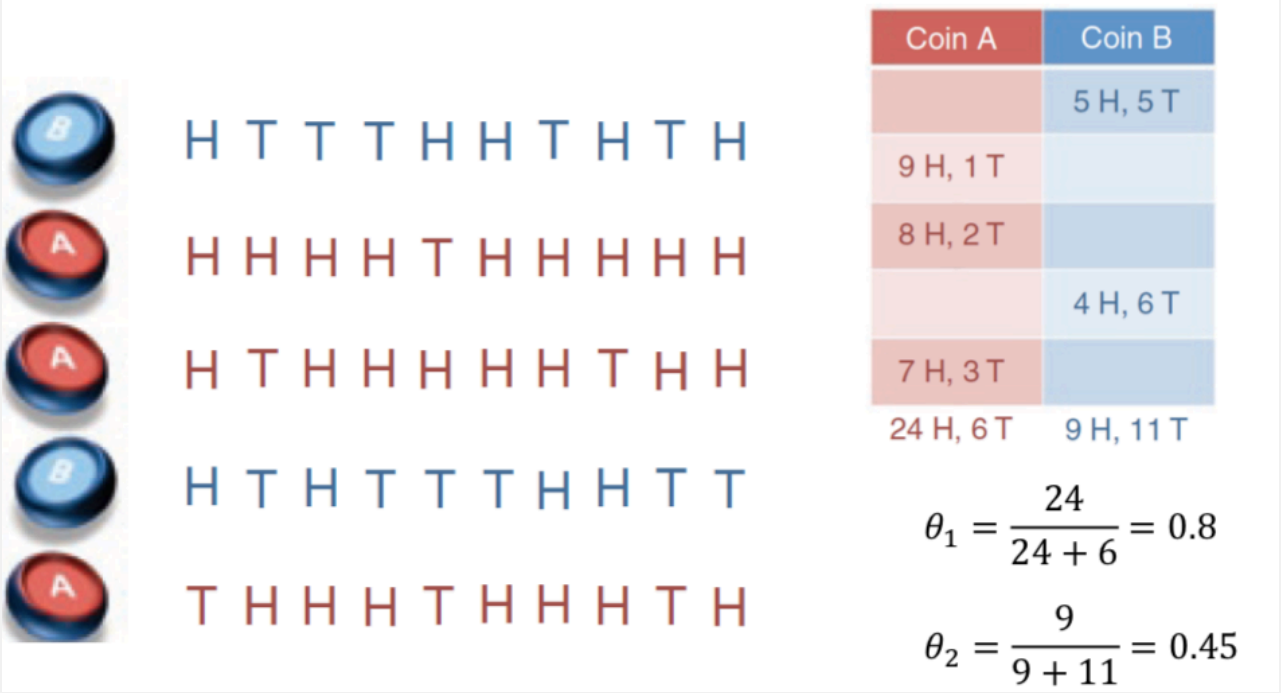
Метод Ньютона

$$f(x+dx) = f(x) + f'(x)dx + \frac{1}{2}f''(x)dx^2$$



Источник: ardianumam.wordpress.com/2017/09/27/newtons-method-optimization-derivation-and-how-it-works

Пример



Coin A	Coin B
	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	
24 H, 6 T	9 H, 11 T

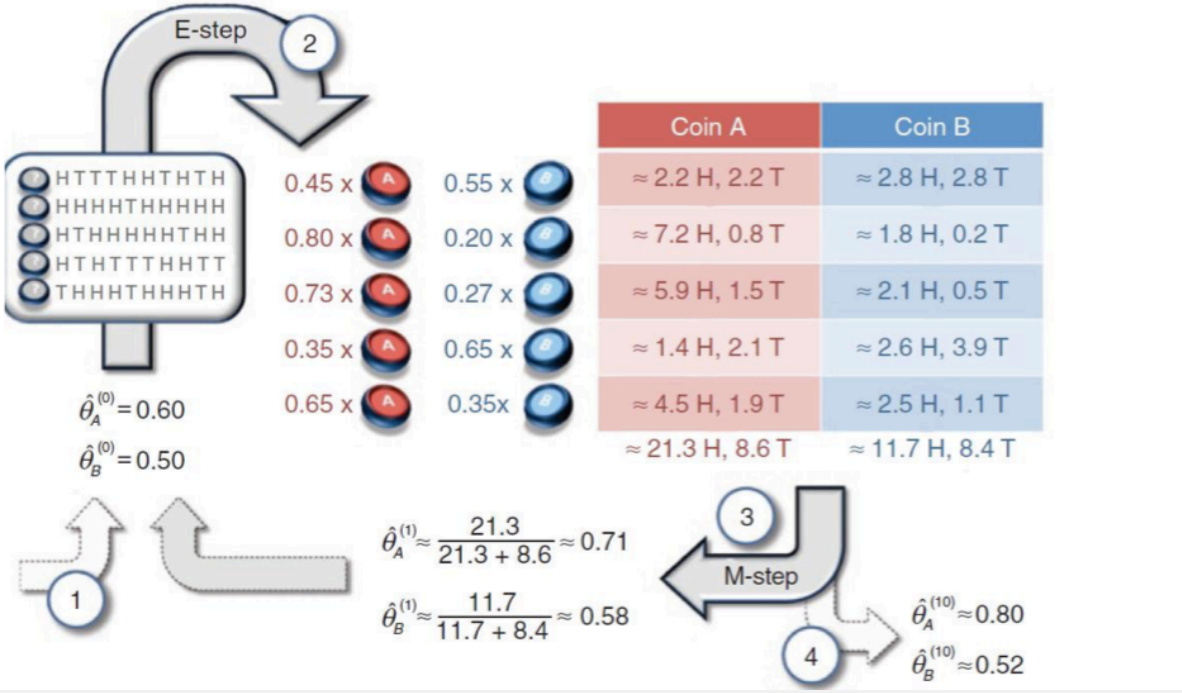
$$\theta_1 = \frac{24}{24+6} = 0.8$$

$$\theta_2 = \frac{9}{9+11} = 0.45$$

Источник: ibug.doc.ic.ac.uk/media/uploads/documents/expectation_maximization-1.pdf

? Что здесь параметры, а что скрытые переменные?

Пример



Источник: ibug.doc.ic.ac.uk/media/uploads/documents/expectation_maximization-1.pdf

Пример

$$\theta_A = 0.6, \theta_B = 0.5$$

$$L(\theta) = p^9(1-p)^{10-9}\ L(\theta_A) = 0.004, L(\theta_B) = 0.001$$

Для 2 броска вероятней, что это монетка A! А как оценить апостериорную вероятность?

$$P(Z|X,\theta^{old}) = \frac{P(X,\theta^{old}|Z)P(Z)}{P(X,\theta^{old})} = \frac{P(X,\theta^{old}|Z)P(Z)}{\sum_i P(Z_i)P(X,\theta^{old}|Z_i)}$$

$$P(Z=A|X,\theta^{old}) = \frac{0.004}{0.004+0.001} = 0.8, P(Z=B|X,\theta^{old}) = \frac{0.001}{0.004+0.001} = 0.2$$

Дальше считаем мат.ожидание. Дальше считаем θ_A, θ_B не через реальное число бросков, а через их мат.ожидания!

Пример

$$p(z=1\,|\,x,\theta^{old}) = \frac{p(x,\theta^{old}\,|\,z=1)p(z=1)}{P(z,\theta^{old})} = \frac{\theta_A^x(1-\theta_A)^{1-x}}{\theta_A^x(1-\theta_A)^{1-x} + \theta_B^x(1-\theta_B)^{1-x}}$$

$$\mathop{\mathbb{E}}_{Z} \log P(X,Z\,|\,\theta) = \mathbb{E}\log[(\theta_A^x(1-\theta_A)^{1-x})^z \cdot (\theta_B^x(1-\theta_B)^{1-x})^{1-z}] =$$

$$= \mathbb{E}[z](x\log\theta_A + (1-x)\log(1-\theta_A)) + \mathbb{E}[1-z](x\log\theta_B + (1-x)\log(1-\theta_B))$$

Дальше считаем производные по θ_A, θ_B подставим вместо $\mathbb{E}[z]$ вероятность $p(z=1\,|\,x,\theta^{old})$ (мат ожидание индикатора)

Ответ будет такой же, как для ММП, но число орлов и решек нужно домножить на $P(Z\,|\,X,\theta^{old})$

Для монетки A число орлов $9 * 0.8 = 7.2$, число решек $1 * 0.8 = 0.8$ Для монетки B число орлов $9 * 0.2 = 1.8$, число решек $1 * 0.2 = 0.2$

Резюме второй части

- Познакомились с итеративным методом оптимизации с помощью вариационной нижней оценки
- Узнали, что такое KL дивергенция и как она связана с Loglossom
- Вывели EM алгоритм
- Разобрали пример

Часть 3
Разделение смеси распределений

Напоминание

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**!

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

?

Как оцениваем каждый множитель?

Напоминание

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

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

?

Какими способами можно оценить $p(x \mid y)$

Напоминание

Методы оценки плотности:

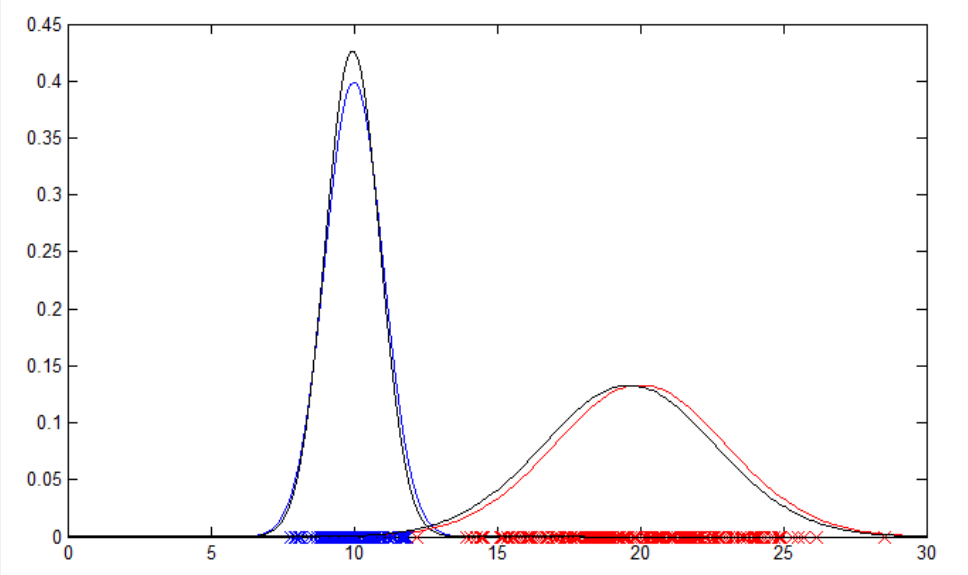
1. Непараметрический — смотрим, какой процент точек из обучающей выборки лежит в окрестности оцениваемого значения.
2. Параметрический — предполагаем, что наше распределение из параметрического семейства. Настраиваем параметры распределения по обучающей выборке.
3. Восстановление из смеси распределений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

Почему нельзя для смеси восстановить непараметрически или параметрически?

Напоминание

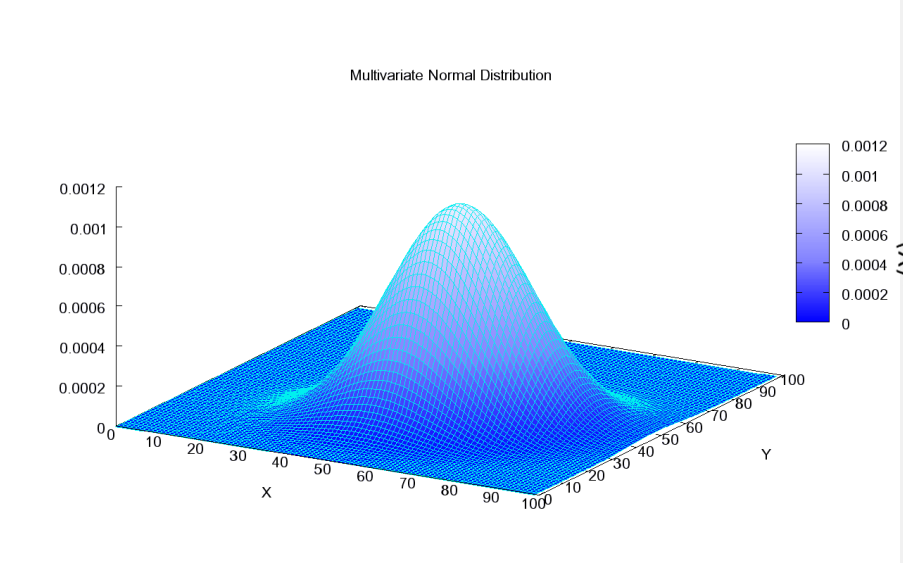
Непараметрически можно. Параметрически не получится.

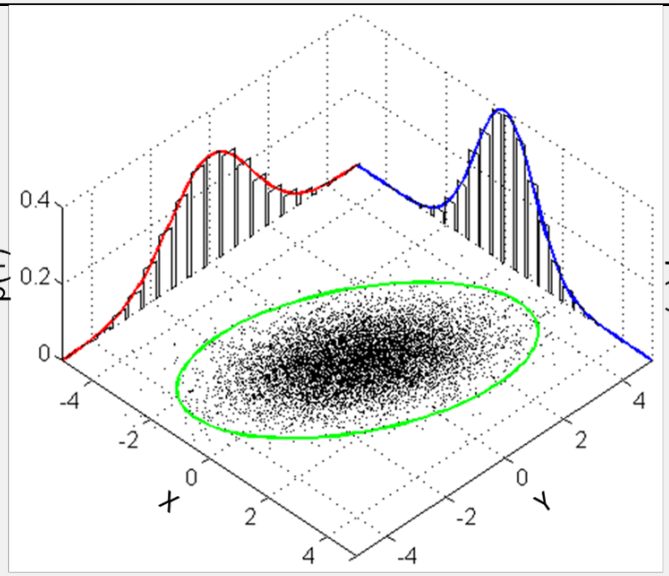
Потому что смесь распределений **не лежит в экспоненциальном классе.** Нужно вводить скрытые переменные.



Источник: chrisjmccormick.files.wordpress.com/2014/08/1d_example.png

Многомерное нормальное распределение



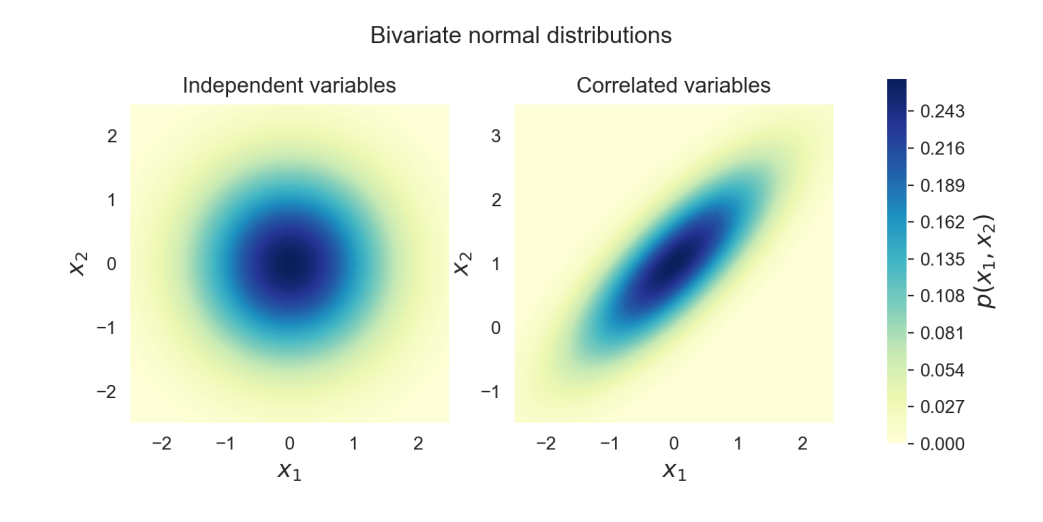


Источник: en.wikipedia.org/wiki/Multivariate_normal_distribution

$$N(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} \mid \Sigma \mid^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)},$$

Многомерное нормальное распределение

Для независимых случайных величин ковариационная матрица диагональна



Источник: peterroelants.github.io/posts/multivariate-normal-primer

Многомерное нормальное распределение

Оцениваем параметры с помощью ММП:

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

$$p(x_i|\theta) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)}$$

Покажите дома, что

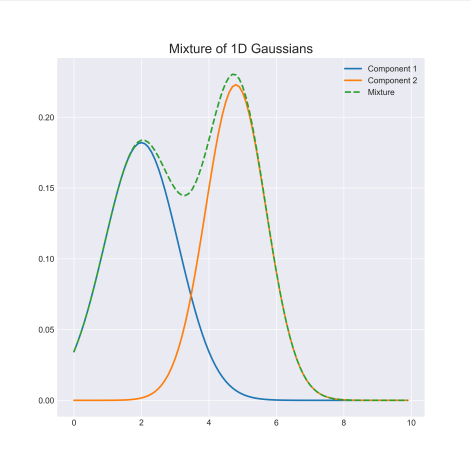
$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i, \Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^T(x_i-\mu)$$

Смесь распределений

$$p(x) = \sum_{k=1}^{K} \pi_k p_k(x), \sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0,$$

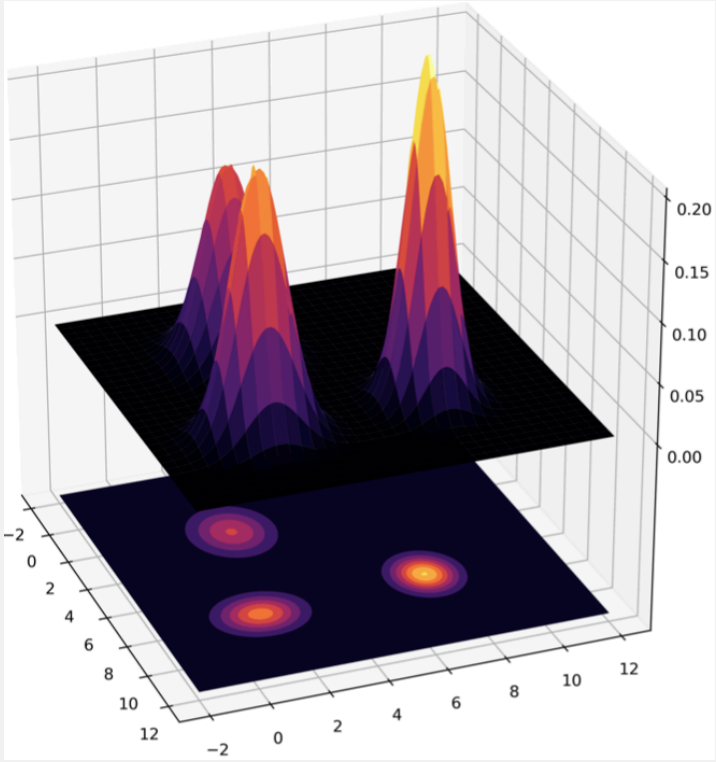
где K - число компонент смеси, $p_k(x)$ - распределение k компоненты, π_k - априорная вероятность k компоненты.

Мы будем говорить про смесь **многомерных нормальный** распределений.



Источник: angusturner.github.io/generative_models/2017/11/03/pytorch-gaussian-mixture-model.html

Смесь распределений



Источник: angusturner.github.io/generative_models/2017/11/03/pytorch-gaussian-mixture-model.html

?

Какие здесь скрытые переменные?

Правдоподобие

Параметризуем $p_k(x) = \phi(x \mid \Theta_k)$. Неполное правдоподобие выглядит плохо:

$$\log P(X \mid \Theta) = \log \prod_{i=1}^{N} p(x_i \mid \Theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k \phi(x_i \mid \Theta_k)$$

Полное правдоподобие:

$$\log P(X, Z \mid \Theta) = \log \prod_{i=1}^{N} p(x_i, Z \mid \Theta)$$

Введем скрытую переменную, которая будет отвечать за выбор компоненты.

z - k-мерный вектор, у которого одна компонента равна 1 , а остальные 0.

$$p(x_i, Z \mid \Theta) = \prod_{k=1}^{K} [\pi_k \phi(x_i \mid \Theta_k)]^{z_{i,k}}$$

$$\log P(X, Z \mid \Theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} z_{i,k} (\log \pi_k + \log \phi(x_i \mid \Theta_k))$$

Вот так уже намного приятнее!

Переход к нормальному распределению

Неполное правдоподобие:

$$\log P(X|\Theta) = \log \prod_{i=1}^{N} p(x_i|\Theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \Sigma_k)$$

Полное правдоподобие:

$$\log P(X, Z|\mu_k, \Sigma_k) = \sum_{i=1}^{N} \sum_{k=1}^{K} z_{i,k}(\log \pi_k + \log N(x_i|\mu_k, \Sigma_k))$$

Где $N(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{n/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}$

Такую модель называют Gaussian Mixture Models(GMM)

EM для смеси нормальных распределений

E шаг: считаем апостериорное распределение на скрытые переменные

$$p(z_{i,k}=1 \mid x_i, \Theta^{old}) = \frac{p(z_{i,k}=1)p(x_i \mid z_{i,k}=1, \Theta^{old})}{p(x_i \mid \Theta^{old})} = \frac{\pi_k^{old} N(x_i \mid \mu_k^{old}, \Sigma_k^{old})}{\sum_{j=1}^{K} \pi_j^{old} N(x_i \mid \mu_j^{old}, \Sigma_j^{old})} = g_{i,k}$$

Поэтому разделение смеси называют **мягкой кластеризацией**.

M шаг: максимизируем мат. ожидание логарифма полного правдоподобия

$$E_z \log P(X, Z \mid \mu_k, \Sigma_k) = E_z \sum_{i=1}^{N} \sum_{k=1}^{K} z_{i,k}(\log \pi_k + \log N(x_i \mid \mu_k, \Sigma_k)) =$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} g_{i,k}(\log \pi_k + \log N(x_i \mid \mu_k, \Sigma_k)) \to \max_{\mu_k, \Sigma_k, \pi_k}$$

при условии $\sum_{k=1}^{K} \pi_k = 1$

?

Как найти точку максимума?

EM для смеси нормальных распределений

Построив функцию Лагранжа и приравняв ее градиент к нулю. Тогда получим:

$$\pi_k = \frac{1}{N}\sum_{i}^{N} g_{i,k}$$

$$\mu_k = \frac{\sum_i^N g_{i,k} x_i}{\sum_i^N g_{i,k}} \; \Sigma_k = \frac{\sum_i^N g_{i,k}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i^N g_{i,k}}$$

ММП для нормального распределения давало вот что:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i, \Sigma = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^T (x_i - \mu)$$

То есть по сути такие же формулы, только взвешены с помощью $g_{i,k}$(насколько объект подходит под компоненту)

EM для смеси нормальных распределений

Алгоритм разделения смеси нормальных распределений

1. Выбрали начальные параметры π_k, μ_k, Σ_k
2. Оценили для каждого объекта его принадлежность к каждой гауссиане $g_{i,k}$
3. Обновили параметры π_k, μ_k, Σ_k по формулам
4. Повторять 2 3 до сходимости

? Зачем это может понадобиться, кроме как для восстановления плотности?

EM для смеси нормальных распределений

Это обучение без учителя, мягкая кластеризация.

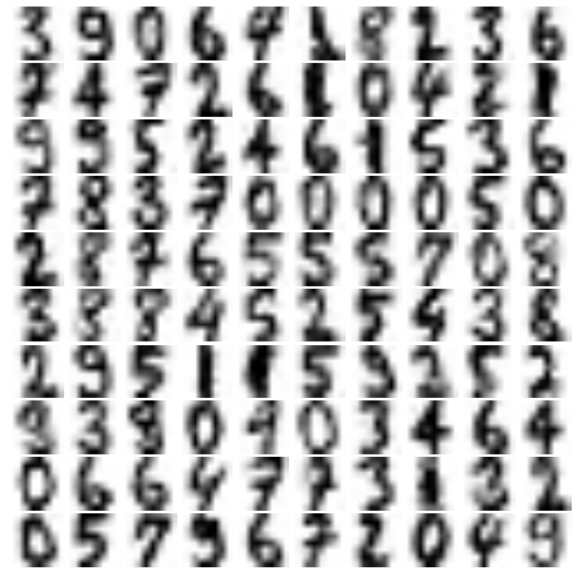
Вы можете применять ее на огромных массивах неразменных данных, находя такие мягкие кластера.

Дальше можно их использовать **как признаки** в задаче обучения с учителем.

?

GMM это генеративная модель?

GMM как генеративная модель



Источник: jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

? А какую генеративную модель мы проходили ранее?

Байесовский классификатор как генеративная модель

Байесовский классификатор:

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

Вы моделируете $p(x \mid y)$, а не просто $p(y \mid x)$.

С его помощью тоже можно генерировать новые объекты.

Предельный случай

Разделяем смесь нормальных распределений. Пусть $\Sigma = \sigma^2 I$, единичная матрица, σ^2 стремится к нулю, априорные вероятности кластеров равны.

$$p(z_{i,k} = 1 \,|\, x_i, \Theta^{old}) = \frac{\pi_k^{old} N(x_i \,|\, \mu_k^{old}, \Sigma_k^{old})}{\sum_{j=1}^{K} \pi_j^{old} N(x_i \,|\, \mu_j^{old}, \Sigma_j^{old})} = \frac{exp(-\frac{1}{2\sigma^2} ||x_i - \mu_k||^2)}{\sum_{j=1}^{K} exp(-\frac{1}{2\sigma^2} ||x_i - \mu_j||^2)} = g_{i,k}$$

Если σ^2 стремится к нулю, то $g_{i,k} = 1$ для самой близкой к объекту i гауссианы и $g_{i,k} = 0$ для всех остальных.

Дальше пересчитали единственный параметр:

$$\mu_k = \frac{\sum_i^N g_{i,k} x_i}{\sum_i^N g_{i,k}}$$

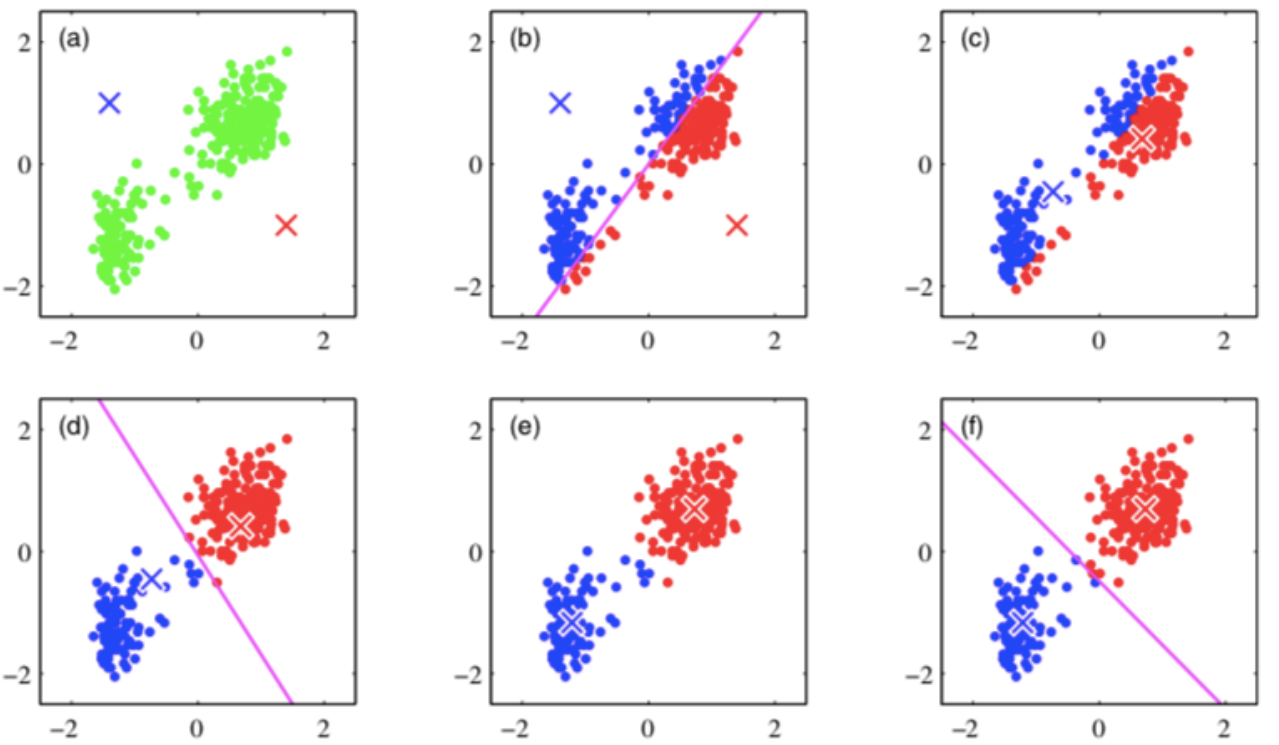
Ничего не напоминает?

K-means

Это и есть k-means! При устремлении гауссиан к дельта функциям мы переходим от мягкой кластеризации к жесткой.

1. Выбрали начальные центры кластеров
2. Каждый объект отнесли к ближайшему кластера
3. Обновили центры кластеров по формулам
4. Повторять 2 3 до сходимости

K-means



Источник: Bishop

Резюме третьей части

- Вспомнили, что такое задача восстановления плотности
- Познакомились с многомерным нормальным распределением
- С помощью EM алгоритма нашли параметры GMM
- Вспомнили, что такое генеративные модели
- Узнали связь k-means с GMM

ТЕХНОСФЕРА

Спасибо за внимание!