КЛАСТЕРИЗАЦИЯ И EM-АЛГОРИТМ

Сергей Николенко

СПбГУ — Санкт-Петербург

21 октября 2020 г.

Random facts:

- 21 октября в Великобритании — Apple Day; с 1990 года проводятся ярмарки с конкурсами вроде стрельбы из лука по яблокам или срезания самой длинной кожуры
- 21 октября 63 г. до н.э. избранный консулом Цицерон, получив сведения о намерениях Катилины совершить переворот, произнёс в сенате речь (то самое «Доколе же ты, Катилина, будешь злоупотреблять нашим терпением?») и планы Катилины сорвал
- 21 октября 1805 г. Пьер-Шарль Вильнёв, несмотря на возражения Антонио де Эсканьо, выстроил свои корабли в линию; увидев эскадру Горацио Нельсона, Вильнёв приказал сделать поворот фордевинд, но не успел выстроиться в кильватерный строй, и англичане победили превосходящие силы противника; сам Нельсон погиб, его тело для сохранности поместили в бочку с ромом, и с тех пор выдаваемый на кораблях ром английские моряки называли «адмиральской кровью»
- 21 октября 1824 г. Джозеф Аспдин запатентовал портлендский цемент, 21 октября 1832 г. Павел Шиллинг в своей петербургской квартире впервые продемонстрировал изобретённый им электромагнитный телеграф, а 21 октября 1879 г. Томас Эдисон испытал свою первую лампу накаливания с угольной нитью

Кластеризация

Суть лекции

- *Кластеризация* — типичная задача обучения без учителя: задача классификации объектов одной природы в несколько групп так, чтобы объекты в одной группе обладали одним и тем же свойством.
- Под свойством обычно понимается близость друг к другу относительно выбранной метрики.

Чуть более формально

- Есть набор тестовых примеров $X = \{x_1, \dots, x_n\}$ и функция расстояния между примерами ρ.
- Требуется разбить X на непересекающиеся подмножества (кластеры) так, чтобы каждое подмножество состояло из похожих объектов, а объекты разных подмножеств существенно различались.

Идея

- Есть точки $x_1, x_2, \dots, x_n$ в пространстве. Нужно кластеризовать.
- Считаем каждую точку кластером. Затем ближайшие точки объединяем, далее считаем единым кластером. Затем повторяем.
- Получается дерево.

Алгоритм

`HierarchyCluster`$(X = \{x_1, \dots, x_n\})$

- Инициализируем $C = X$, $G = X$.
- Пока в C больше одного элемента:
  - Выбираем два элемента C c_1 и c_2, расстояние между которыми минимально.
  - Добавляем в G вершину c_1c_2, соединяем её с вершинами c_1 и c_2.
  - $C := C \cup \{c_1c_2\} \quad \{c_1, c_2\}$.
- Выдаём G.

Результат

- В итоге получается дерево кластеров, из которого потом можно выбрать кластеризацию с требуемой степенью детализации (обрезать на том или ином максимальном расстоянии).
- Всё ли понятно?

Результат

- В итоге получается дерево кластеров, из которого потом можно выбрать кластеризацию с требуемой степенью детализации (обрезать на том или ином максимальном расстоянии).
- Всё ли понятно?
- Остаётся вопрос: как подсчитывать расстояние между кластерами?

Single-link vs. complete-link

- *Single-link* алгоритмы считают *минимум* из возможных расстояний между парами объектов, находящихся в кластере.
- *Complete-link* алгоритмы считают *максимум* из этих расстояний
- Какие особенности будут у single-link и complete-link алгоритмов? Чем они будут отличаться?

Очевидный алгоритм

- Нарисуем полный граф с весами, равными расстоянию между объектами.
- Выберем некий предопределённый порог расстояния r и выбросим все рёбра длиннее r.
- Компоненты связности полученного графа — это наши кластеры.

Минимальное остовное дерево

- Минимальное остовное дерево — дерево, содержащее все вершины (связного) графа и имеющее минимальный суммарный вес своих рёбер.
- Алгоритм Краскала (Kruskal): выбираем на каждом шаге ребро с минимальным весом, если оно соединяет два дерева, добавляем, если нет, пропускаем.
- Алгоритм Борувки (Boruvka).

Кластеризация

- Как использовать минимальное остовное дерево для кластеризации?

Кластеризация

- Как использовать минимальное остовное дерево для кластеризации?
- Построить минимальное остовное дерево, а потом выкидывать из него рёбра максимального веса.
- Сколько рёбер выбросим, столько кластеров получим.

DBSCAN

- Идея: кластер – это зона высокой плотности точек, отделённая от других кластеров зонами низкой плотности.
- Алгоритм: выделяем *core samples*, которые сэмплируются в зонах высокой плотности (т.е. есть по крайней мере n соседей, других точек на расстоянии $\leq \epsilon$).
- Затем последовательно объединяем core samples, которые оказываются соседями друг друга.
- Точки, которые не являются ничьими соседями, — это выбросы.

Birch

- Идея: строим дерево (CF-tree, от clustering feature), которое содержит краткие описания кластеров и поддерживает апдейты.
- $\mathrm{CF}_i = \{N_i, \mathrm{LS}_i, \mathrm{SS}_i\}$: число точек в кластере CF_i, $\mathrm{LS}_i = \sum_{\mathbf{x} \in \mathrm{CF}_i} x_i$ (linear sum), $\mathrm{SS}_i = \sum_{\mathbf{x} \in \mathrm{CF}_i} x_i^2$ (sum of squares).
- Этого достаточно для того, чтобы подсчитать разумные расстояния между кластерами.
- А также для того, чтобы слить два кластера: CF_i аддитивны.

Birch

- CF-дерево состоит из CF_i; оно похоже на B-дерево, сбалансировано по высоте. Кластеры – листья дерева, над ними “суперкластеры”.
- Добавляем новый кластер, рекурсивно вставляя его в дерево; если от этого число элементов в листе становится слишком большим (параметр), лист разбивается на два.
- А когда дерево построено, можно запустить ещё одну кластеризацию (любым другим методом) на полученных “мини-кластерах”.

Алгоритм EM

Постановка задачи

- Часто возникает ситуация, когда в имеющихся данных некоторые переменные присутствуют, а некоторые — отсутствуют.
- Даны результаты сэмплирования распределения вероятностей с несколькими параметрами, из которых известны не все.

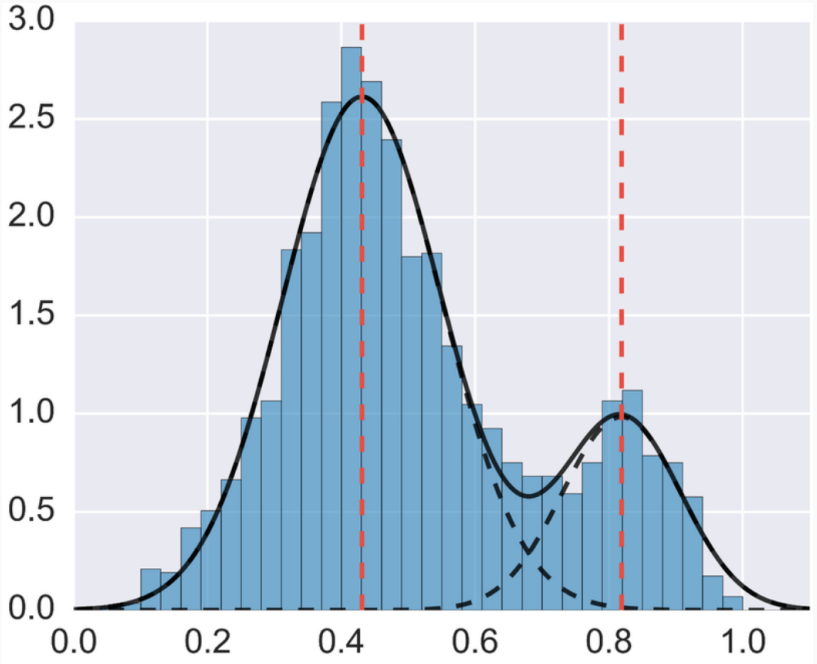
Постановка задачи

- Эти неизвестные параметры тоже расцениваются как случайные величины.
- Задача — найти наиболее вероятную гипотезу, то есть ту гипотезу h, которая максимизирует

$$E[\ln p(D|h)].$$

Частный случай

- Построим один из простейших примеров применения алгоритма EM. Пусть случайная переменная y сэмплируется из суммы двух нормальных распределений. Дисперсии даны (одинаковые), нужно найти только средние μ_1, μ_2.



- Какое тут правдоподобие? Как его оптимизировать?

Два распределения

- Нельзя понять, какие y_i были порождены каким распределением — классический пример *скрытых переменных*.
- Один тестовый пример полностью описывается как тройка $\langle y_i, z_{i1}, z_{i2} \rangle$, где $z_{ij} = 1$ iff y_i был сгенерирован j-м распределением.

Суть алгоритма EM

- Сгенерировать какую-нибудь гипотезу $h = (\mu_1, \mu_2)$.
- Пока не дойдем до локального максимума:
  - Вычислить ожидание $E(z_{ij})$ в предположении текущей гипотезы (E–шаг).
  - Вычислить новую гипотезу $h' = (\mu_1', \mu_2')$, предполагая, что z_{ij} принимают значения $E(z_{ij})$ (M–шаг).

В примере с гауссианами

- В примере с гауссианами:

$$E(z_{ij}) = \frac{p(y = y_i | \mu = \mu_j)}{p(y = y_i | \mu = \mu_1) + p(y = y_i | \mu = \mu_2)} = \\ = \frac{e^{-\frac{1}{2\sigma^2}(y_i - \mu_j)^2}}{e^{-\frac{1}{2\sigma^2}(y_i - \mu_1)^2} + e^{-\frac{1}{2\sigma^2}(y_i - \mu_2)^2}}.$$

- Мы подсчитываем эти ожидания, а потом подправляем гипотезу:

$$\mu_j \leftarrow \frac{1}{m} \sum_{i=1}^{m} E(z_{ij}) y_i.$$

- Звучит логично, но с какой стати это всё работает? Об этом потом. :)

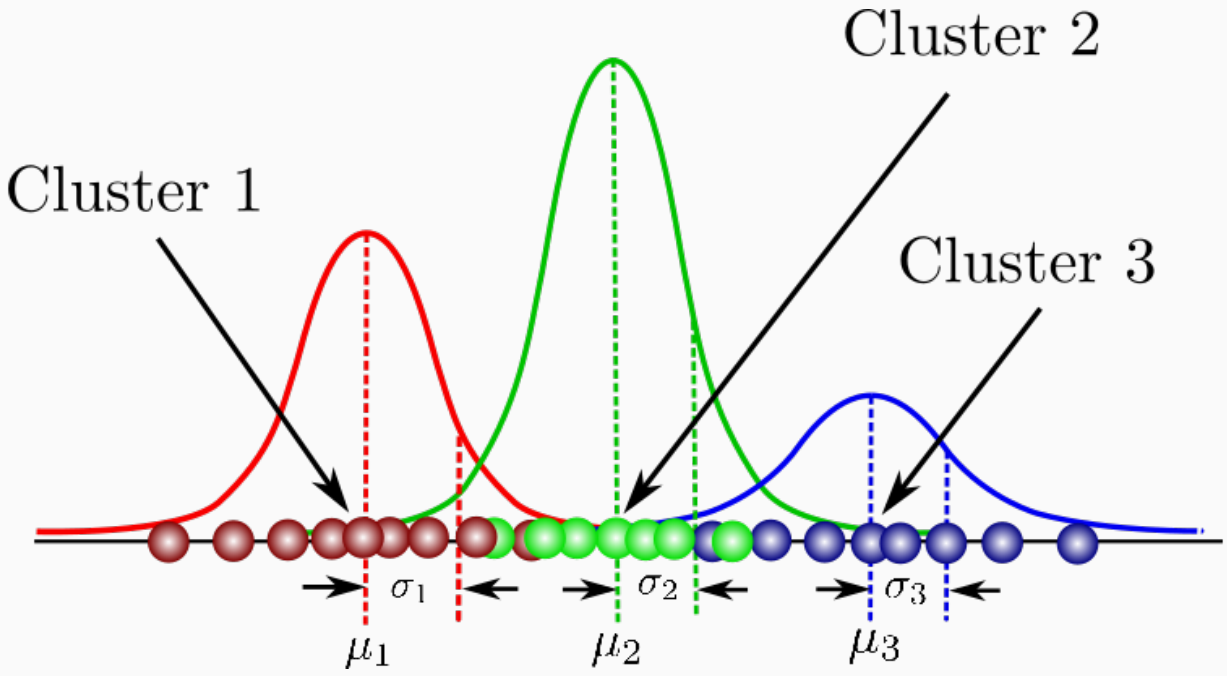
EM для кластеризации

Мысли?

- Какие есть мысли о применении алгоритма EM к задачам кластеризации?

Мысли?

- Какие есть мысли о применении алгоритма EM к задачам кластеризации?
- Кластеризацию можно формализовать как всё ту же задачу разделения смеси распределений:



Гипотезы

- Чтобы воспользоваться статистическим алгоритмом, нужно сформулировать гипотезы о распределении данных.
- *Гипотеза о природе данных*: тестовые примеры появляются случайно и независимо, согласно вероятностному распределению, равному смеси распределений кластеров

$$p(\mathbf{y}) = \sum_{c \in C} w_c p_c(\mathbf{y}), \quad \sum_{c \in C} w_c = 1,$$

где w_c — вероятность появления объектов из кластера c, p_c — плотность распределения кластера c.

Гипотезы cont'd

- Остается вопрос: какими предположить распределения p_c?

Гипотезы cont'd

- Остается вопрос: какими предположить распределения p_c?
- Часто берут сферические гауссианы, но это не слишком гибкий вариант: кластер может быть вытянут в ту или иную сторону.

Гипотезы cont'd

- Остается вопрос: какими предположить распределения p_c?
- Часто берут сферические гауссианы, но это не слишком гибкий вариант: кластер может быть вытянут в ту или иную сторону.
- Можно взять, например, эллиптические гауссианы.
- *Гипотеза* 2: Каждый кластер c описывается d–мерной гауссовской плотностью с центром $\mu_c = \{\mu_{c1}, \dots, \mu_{cd}\}$ и диагональной матрицей ковариаций $\Sigma_c = \mathrm{diag}(\sigma_{c1}^2, \dots, \sigma_{c2}^2)$ (т.е. по каждой координате своя дисперсия).

Постановка задачи и общий вид алгоритма

- В этих предположениях получается в точности задача разделения смеси вероятностных распределений. Для этого и нужен EM–алгоритм.
- Каждый тестовый пример описывается своими координатами $(\mathbf{y}_1, \dots, \mathbf{y}_N)$.
- Скрытые переменные $\mathbf{z}_n$ в данном случае — это one-hot вектор $\mathbf{z}_n = (z_{nc})$ того, какому кластеру принадлежит $\mathbf{y}_n$.
- Обозначим вероятности того, что объект $\mathbf{y}_n$ принадлежит кластеру $c \in C$, через g_{nc}.

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{nc}:

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{nc}:

$$g_{nc} = \frac{w_c p_c(\mathbf{y}_n)}{\sum_{c' \in C} w_{c'} p_{c'}(\mathbf{y}_n)}.$$

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{nc}:

$$g_{nc} = \frac{w_c p_c(\mathbf{y}_n)}{\sum_{c' \in C} w_{c'} p_{c'}(\mathbf{y}_n)}.$$

- M–шаг: с использованием g_{nc} уточняются параметры кластеров w, μ, σ:

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{nc}:

$$g_{nc} = \frac{w_c p_c(\mathbf{y}_n)}{\sum_{c' \in C} w_{c'} p_{c'}(\mathbf{y}_n)}.$$

- M–шаг: с использованием g_{nc} уточняются параметры кластеров w, μ, σ:

$$w_c = \frac{1}{N} \sum_{n=1}^{N} g_{nc}, \quad \mu_c = \frac{1}{n w_c} \sum_{n=1}^{N} g_{nc} \mathbf{y}_n,$$

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{nc}:

$$g_{nc} = \frac{w_c p_c(\mathbf{y}_n)}{\sum_{c' \in C} w_{c'} p_{c'}(\mathbf{y}_n)}.$$

- M–шаг: с использованием g_{nc} уточняются параметры кластеров w, μ, σ:

$$w_c = \frac{1}{N} \sum_{n=1}^{N} g_{nc}, \quad \mu_c = \frac{1}{n w_c} \sum_{n=1}^{N} g_{nc} \mathbf{y}_n,$$

$$\sigma_{cj}^2 = \frac{1}{n w_c} \sum_{n=1}^{N} g_{nc} \left(y_{nj} - \mu_{cj}\right)^2.$$

Алгоритм

`EMCluster`$(Y, |C|)$:

- Инициализировать $|C|$ кластеров; начальное приближение: $w_c := 1/|C|$, $\mu_c :=$ случайный $\mathbf{y}_n$, $\sigma^2_{cj} := \frac{1}{n|C|}\sum_{i=1}^{N}\left(y_{nj} - \mu_{cj}\right)^2$.
- Пока принадлежность кластерам не перестанет изменяться:
  - E–шаг: $g_{nc} := \frac{w_c p_c(\mathbf{y}_n)}{\sum_{c' \in C} w_{c'} p_{c'}(\mathbf{y}_n)}$.
  - M–шаг: $w_c = \frac{1}{N}\sum_{i=1}^{N} g_{nc}$, $\mu_{cj} = \frac{1}{nw_c}\sum_{i=1}^{N} g_{nc} y_{nj}$,

$$\sigma^2_{cj} = \frac{1}{nw_c}\sum_{i=1}^{N} g_{nc}\left(y_{nj} - \mu_{cj}\right)^2.$$

- После сходимости определить принадлежность x_i к кластерам:

$$\mathrm{clust}_i := \arg\max_{c \in C} g_{nc}.$$

Почему так?

- Как доказать, что E-шаг и M-шаг действительно в данном случае так выглядят?
- На E-шаге для параметров $\theta = (w, \mu, \sigma)$:

$$Q(\theta \mid \theta^{(m)}) = \mathbb{E}_{Z|Y,\theta^{(m)}} \left[\log p(Y, Z \mid \theta)\right] =$$

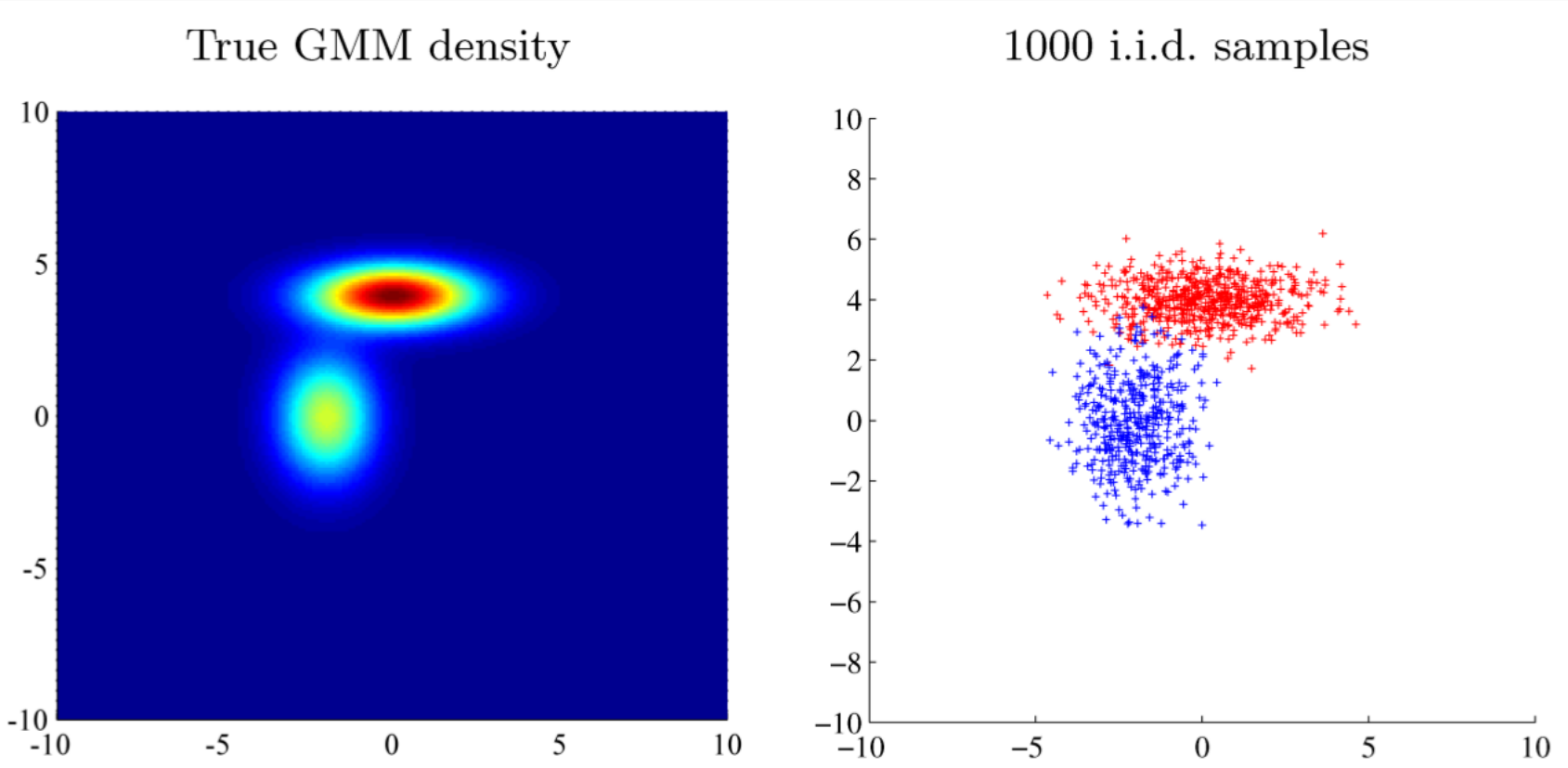
$$= \mathbb{E}_{\theta^{(m)}} \left[\log \prod_{n=1}^{N} \prod_{c=1}^{C} p(\mathbf{y}_n, z_{nc} \mid \theta)^{z_{nc}}\right] =$$

$$= \mathbb{E}_{\theta^{(m)}} \left[\sum_{n=1}^{N} \sum_{c=1}^{C} z_{nc} \left(\log p(z_{nc} \mid w_c) + \log p(\mathbf{y}_n \mid \mu_c, \sigma_c)\right)\right] =$$

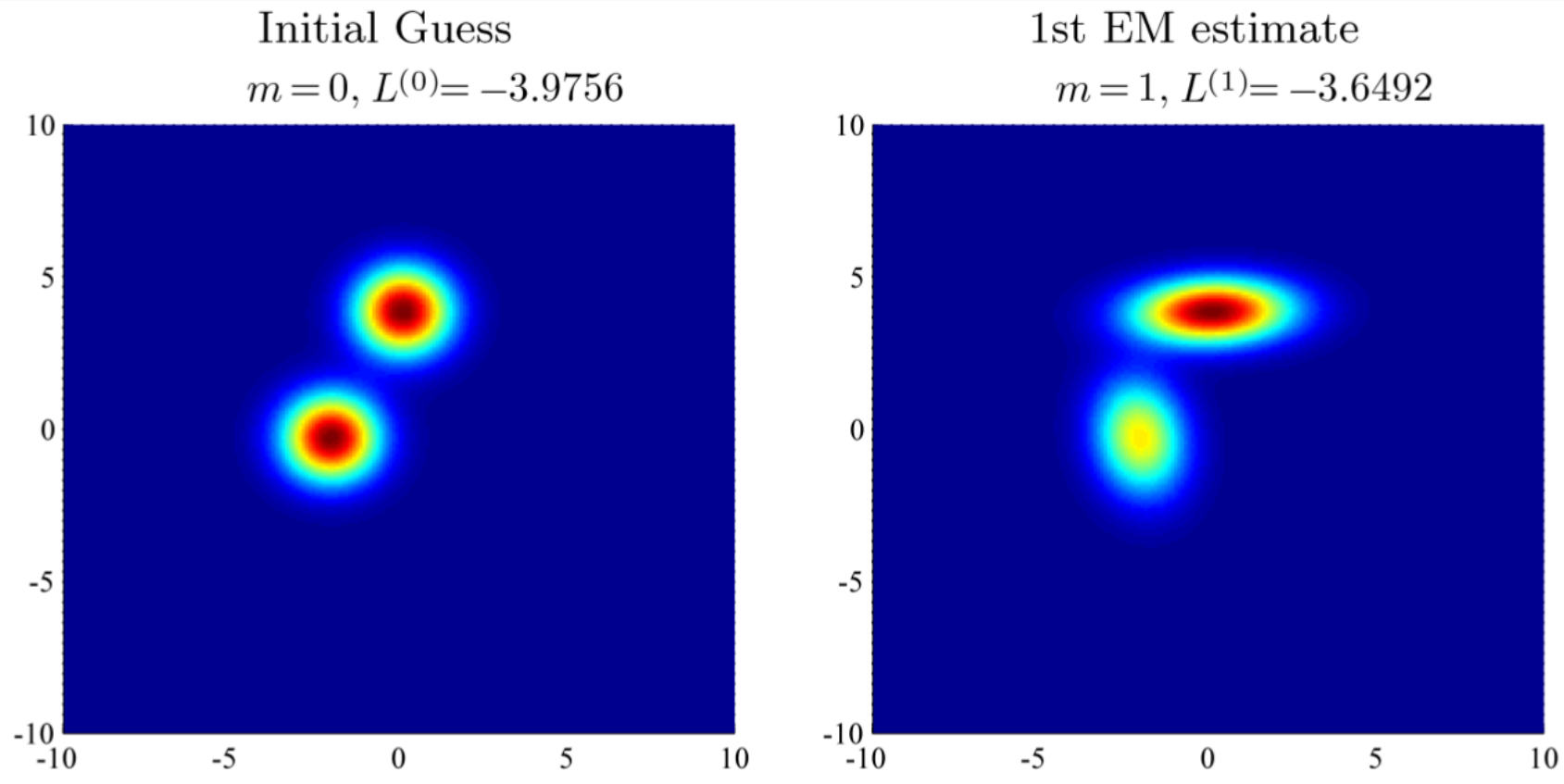
$$= \sum_{n=1}^{N} \sum_{c=1}^{C} \left(\mathbb{E}_{\theta^{(m)}} \left[z_{nc}\right] \log w_c + \mathbb{E}_{\theta^{(m)}} \left[z_{nc}\right] \log p(\mathbf{y}_n \mid \mu_c, \sigma_c)\right).$$

- Отсюда и получается обучение каждого гауссиана независимо, но с весами $g_{nc} = \mathbb{E}_{\theta^{(m)}} \left[z_{nc}\right]$.

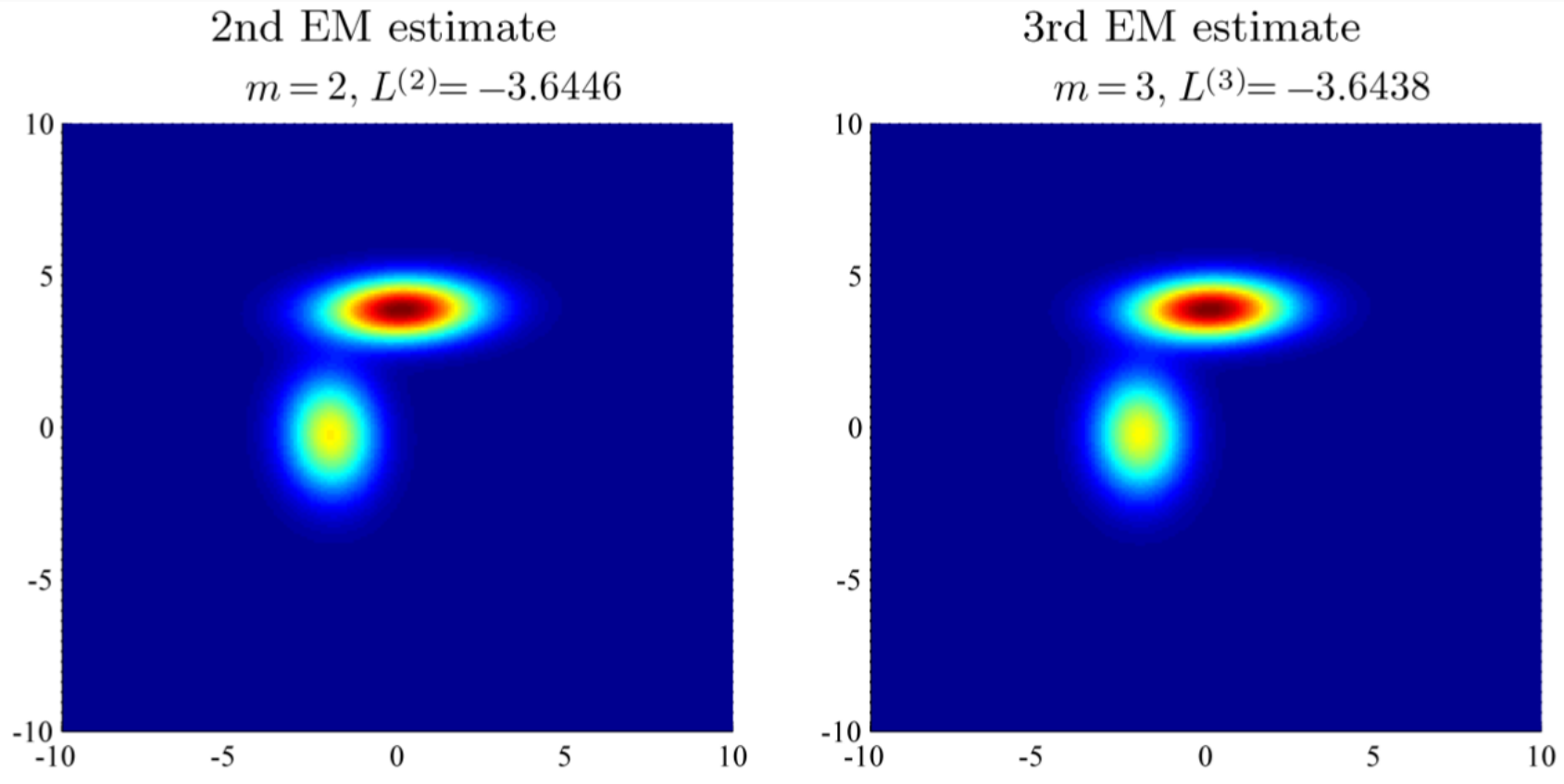
Пример


True GMM density
1000 i.i.d. samples
10
5
0
-5
-10
-10
-5
0
5
10
10
8
6
4
2
0
−2
−4
−6
−8
−10
−10
−5
0
5
10

Пример


Initial Guess
$m = 0$, $L^{(0)} = -3.9756$
1st EM estimate
$m = 1$, $L^{(1)} = -3.6492$
10
5
0
-5
-10
-10
-5
0
5
10

Пример


2nd EM estimate
$m = 2$, $L^{(2)} = -3.6446$
3rd EM estimate
$m = 3$, $L^{(3)} = -3.6438$
10
5
0
-5
-10
-10
-5
0
5
10

Проблема

- Остается проблема: нужно задавать количество кластеров.
- Как её решать?

Свойства и простые расширения EM

Суть алгоритма k--средних

- Один из самых известных алгоритмов кластеризации – алгоритм k-средних – это фактически упрощение алгоритма EM.
- Формальная цель алгоритма k–средних — минимизировать меру ошибки

$$E(Y, C) = \sum_{n=1}^{n} ||\mathbf{y}_n - \mu_i||^2,$$

  где μ_i — ближайший к $\mathbf{y}_n$ центр кластера.
- Т.е. мы не относим точки к кластерам, а двигаем центры, а принадлежность точек определяется автоматически.

Алгоритм неформально

- Идея та же, что в EM:
  - Проинициализировать.
  - Классифицировать точки по ближайшему к ним центру кластера.
  - Перевычислить каждый из центров.
  - Если ничего не изменилось, остановиться, если изменилось — повторить.

Алгоритм

kMeans$(Y, |C|)$:

- Инициализировать центры $|C|$ кластеров $\mu_1, \dots, \mu_{|C|}$.
- Пока принадлежность кластерам не перестанет изменяться:
  - Определить принадлежность $\mathbf{y}_n$ к кластерам:

$$\mathrm{clust}_n := \arg\min_{c \in C} \rho(\mathbf{y}_n, \mu_c).$$

  - Определить новое положение центров кластеров:

$$\mu_c := \frac{\sum_{\mathrm{clust}_n = c} \mathbf{y}_n}{\sum_{\mathrm{clust}_n = c} 1}.$$

И чем же это от EM отличается?

Point-estimate variant of EM

- Разница в том, что мы не считаем вероятности принадлежности кластерам, а жестко приписываем каждый объект одному кластеру.
- Это на самом деле вариант EM, в котором вместо полного распределения $p(X \mid \theta)$, которое в Q-функции используется, мы берём просто точку максимального правдоподобия
- Point-estimate variant of EM, или Classification EM:

$$\mathbf{z}^{(m)} = \arg\max_{\mathbf{z}} p(\mathbf{z} \mid \mathbf{y}, \theta^{(m)}),$$
$$\theta^{(m+1)} = \arg\max_{\theta} p(\mathbf{z}^{(m)} \mid \theta^{(m)}).$$

Спасибо!

Спасибо за внимание!