Сравнение kmeans и EM алгоритма

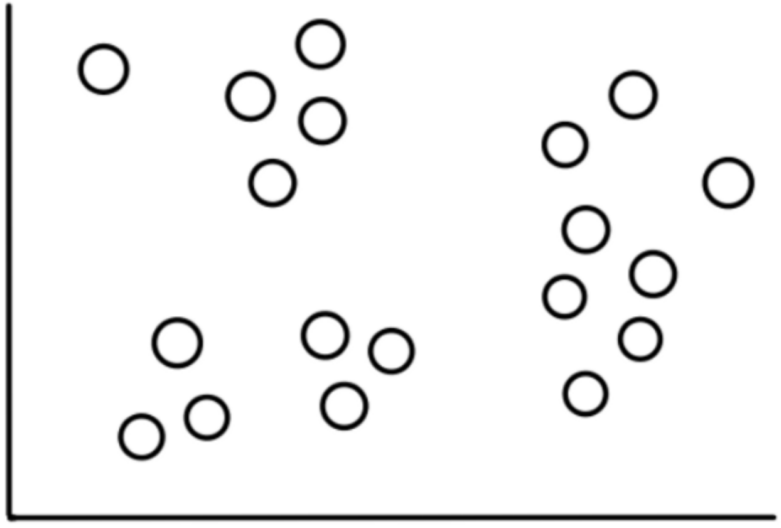
Руденко Данил 2 курс ФПМИ

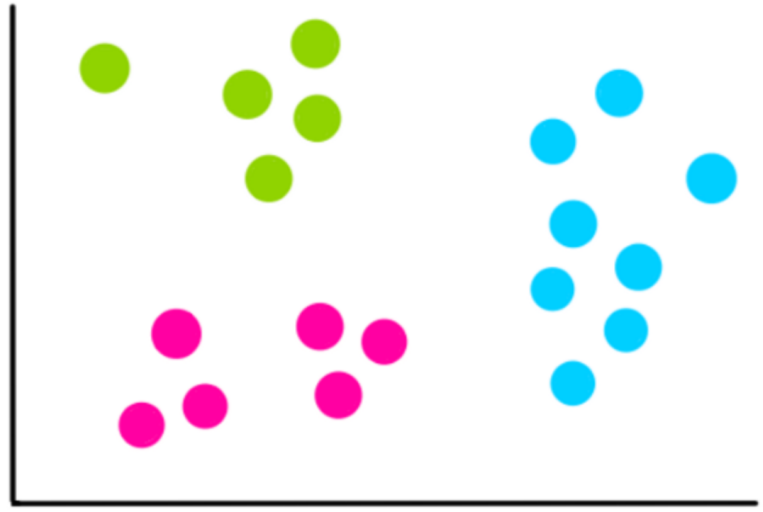
rudenko.dv@phystech.edu

7 апреля 2024 г.

Постановка задачи

- Пусть в результате нескольких экспериментов получили некоторый смешанный набор данных.
- Требуется разделить на куски, похожие друг на друга, найти какую-то структуру
- Идеальный вариант - разделить на отдельные эксперименты





[]

Основные методы кластеризации данных

- Kmeans algorithm (метод k средних)
- EM algorithm (алгоритм ожидания и максимизации)
- Hierarchical algorithm
- Кластеризация с помощью минимального остовного дерева

В этой работе остановимся только на первых двух - Kmeans, EM algorithm

План презентации

Kmeans algorithm

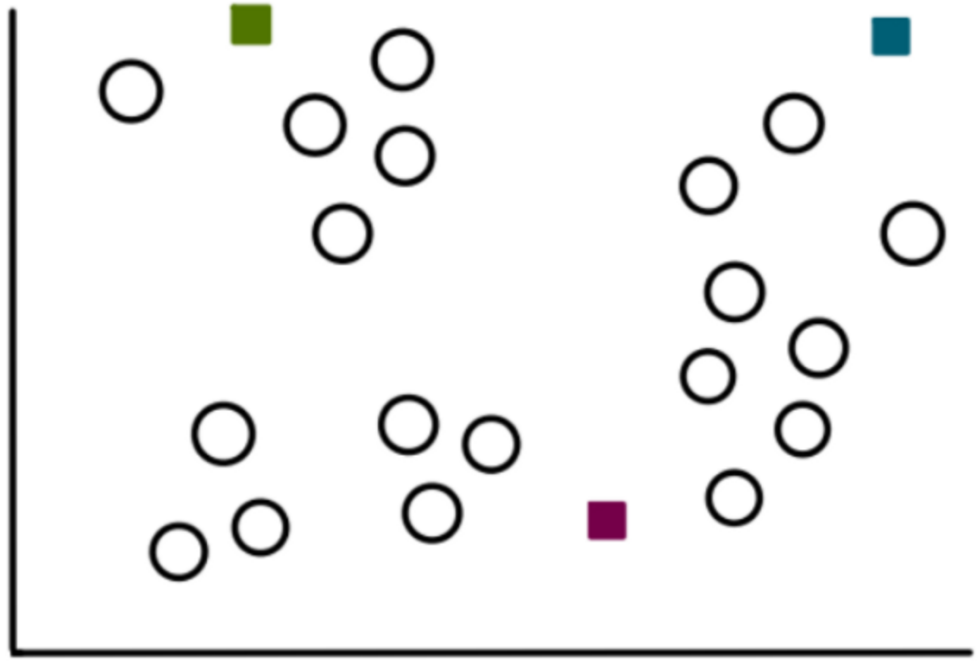
Kmeans algorithm - Алгоритм k средних

- Простой алгоритм, наиболее популярный метод кластеризации
- Состоит из двух повторяющихся этапов:
  1. Обновление параметров кластера
  2. Обновление разделения данных на кластеры
- Завершается, когда не происходит изменения при обновлениях

Рассмотрим шаги поподробнее

- Пусть известно, что данные нужно разделить на k кластеров.
- Пусть также данные описываются набором векторов $\{x_i\}_{i=1}^{N}$.

Тогда нулевой шаг - выбор k случайных векторов $\{\mu\}_{i=1}^{k}$ - центры кластеров. Центры кластеров - единственные их параметры



Первый Шаг

Для каждой точки данных найдем ближайший центр кластера:

$$z_i = \arg\min_c(||x_i - \mu_c||^2)$$

Здесь z_i - скрытый (latent) параметр, принадлежности точки к i кластеру

Второй Шаг

Обновим центры кластеров, так чтобы они были средним от выбранных точек в кластере.

$$\mu_c = \frac{1}{|S|} \sum_{i \in S_c} x_i, S_c = \{i | z_i = c\}$$

Теперь повторяем шаги 1-2, пока центры меняются

2

3

4

5

План презентации

EM - алгоритм нежесткой кластеризации

Алгоритм EM - Итеративный алгоритм поиска оценок максимума правдоподобия модели, в ситуации, когда она зависит от скрытых (ненаблюдаемых) переменных.
Алгоритм ищет параметры модели итеративно, каждая итерация состоит из двух шагов:

1. E (Expectation) шаг — поиск наиболее вероятных значений скрытых переменных.
2. M (Maximization) шаг — поиск наиболее вероятных значений параметров, для полученных на шаге E значений скрытых переменных.

Постановка задачи

- Данные задаются набором векторов $X = \{x_i\}_{i=1}^{N} \subseteq \mathbb{R}^d$
- Считаем, что данные следует разделить на k кластеров (например, было проведено k различных экспериментов, либо подобрали нужное k)
- Плотность распределения смеси:

$$p(x) = \sum_{j=1}^{k} \omega_j p_j(x)$$

где $(\sum_{j=1}^{k} \omega_j = 1)$ ω_j - априорная вероятность j-ой компоненты, $p_j(x) = \psi(x, \theta_j)$ - плотность распределения j-ой смеси

- Хотим максимизировать логарифм правдоподобия

$$\ln(\prod_{i=1}^{N} p_i) = \sum_{i=1}^{N} \ln(\sum_{i=1}^{k} w_i p_i) \to \max_{\Theta}$$

Первый шаг

E (Expectation) шаг — поиск наиболее вероятных значений скрытых переменных. При неизменных значениях обычных переменных

- Пусть $H = (h_{ij})_{m \times k}$, где $h_{ij} = \mathbb{P}(\theta_j | x)$
- По формуле Байеса можно получить

$$h_{ij} = \frac{\omega_j p_j(x_i)}{p(x_i)} = \frac{\omega_j p_j(x_i)}{\sum_{t=1}^{k} \omega_t p_t(x_i)}$$

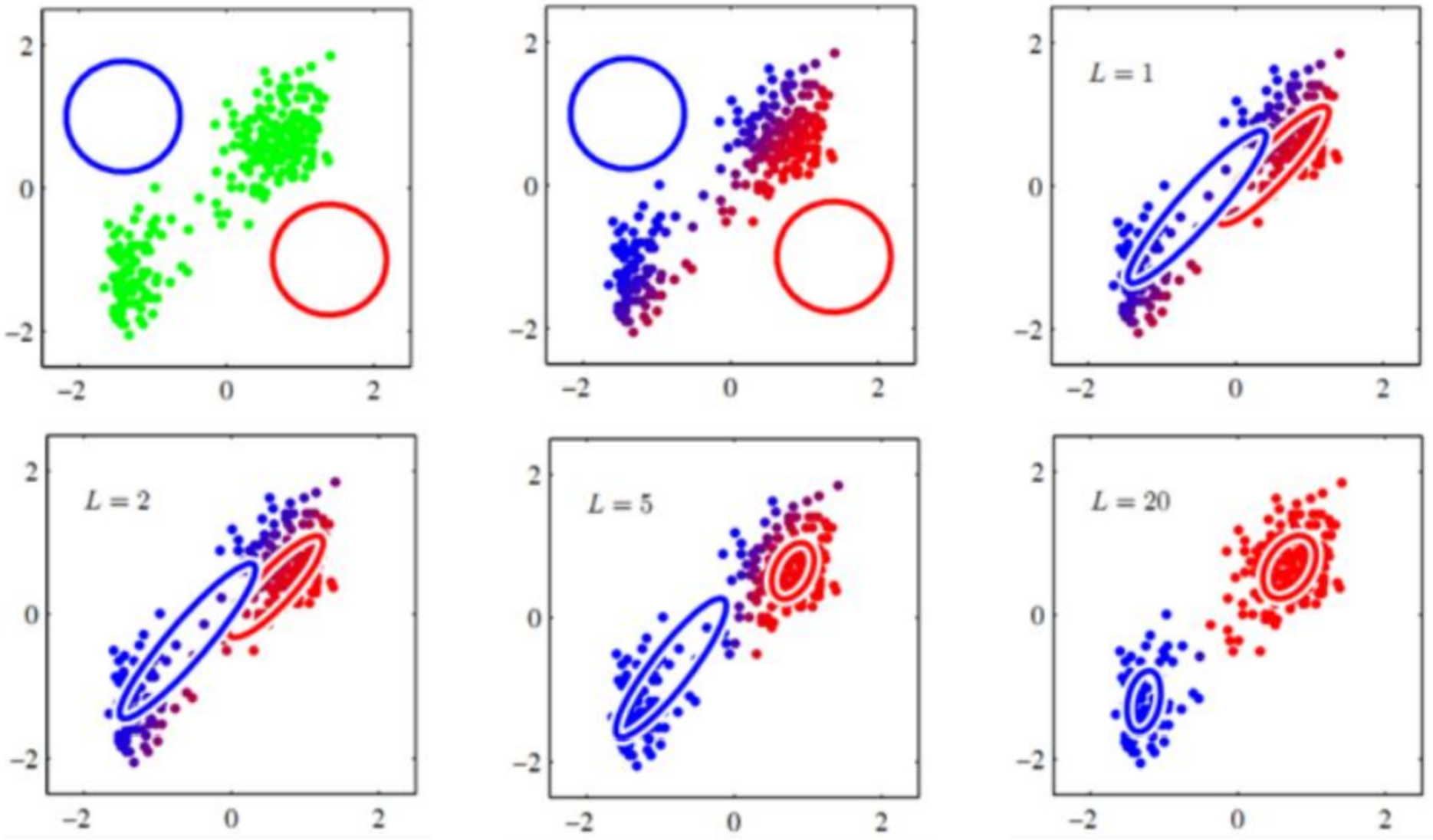
То есть зная значения вектора параметров $\Theta = (\omega_1, \ldots, \omega_k, \theta_1, \ldots, \theta_k)$ можно найти значения скрытых переменных

Второй шаг

M (Maximization) шаг — поиск параметров, которые максимизируют логорифм правдоподобия, при зафиксированных скрытых переменных (с предыдущего пункта)

$$\theta_{j+1} = \arg\max_{\theta} \mathbb{E}_{\mathbb{P}(H|X,\Theta_j)}(\log(\mathbb{P}(X,H|\Theta))) = \arg\max_{\Theta} \sum_{i=1}^{N} h_{ij} \ln(\psi(x,\theta_j))$$

Визуализация



План презентации

Новые параметры

- Предположим, что данные - результат k экспериментов, каждый из которых распределен нормально

$$p_j(x) = \mathcal{N}(x, \mu_j, \Sigma_j)$$

- Тогда явные параметры кластеров - их матожидание (центр распределения), и матрица ковариации

$$\Theta = (\omega_1, \ldots, \omega_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k)$$

Шаги алгоритма

1. E-шаг:

$$h_{ij} = \frac{\omega_j \mathcal{N}(x_i, \mu_j, \Sigma_j)}{\sum_{j=1}^{k} \omega_j \mathcal{N}(x_i, \mu_j, \Sigma_j)}$$

2. M-шаг:

$$\omega_j = \frac{1}{N} \sum_{j=1}^{N} h_{ij}, \quad \mu_j = \frac{1}{N\omega_j} \sum_{i=1}^{N} h_{ij} x_i$$

$$\Sigma_j = \frac{1}{N\omega_j} \sum_{i=1}^{N} h_{ij} (x_n - \mu_j)^T (x_n - \mu_j)$$

План презентации

Посмотрим на данные

Как справился Kmeans

Как справился EM алгоритм

2 нормальных и одно Лапласовое распределение

Теперь немного поменяем задачу - рассмотрим различные распределения в одной смеси. Формулы останутся те же, но нужно поменять $\mathcal{N}(x, \mu, \Sigma)$ на $\mathcal{L}ap(x, \mu, \Sigma)$


Кластеризация 2х нормальных и одного
Лапласссового распределений с помощью EM
0
1
2
8
6
4
2
0
−2
−4
−6
−8
−6
−4
−2
0
2
4
6

План презентации

Преимущества EM

- Правильно работает со смесями распределений
- Лучше справляется с нахождением кластеров по плотности

Недостатки EM

- Сходится к локальному минимуму
- Чувствительна к начальным параметрам
- Нужно заранее знать/подобрать число кластеров

Спасибо за внимание

Рис.: Ссылка на github с моим кодом