

«Машинное обучение и анализ данных»

ЕМ-алгоритм

Александр Дьяконов

14 апреля 2021 года

План

- **Gaussian Mixture Model (GMM)**
- **ЕМ-алгоритм**
- **Генеративные модели**
- **Обоснование ЕМ**

Gaussian Mixture Model (GMM)

распределение данных $p(x) = \sum_{t=1}^k \pi_t \text{norm}(x \mid \mu_t, \Sigma_t)$

$$\sum_{t=1}^k \pi_t = 1, \pi_t \geq 0$$

цель – определить $\{\pi_t, \mu_t, \Sigma_t\}_{t=1}^k$

Понятно, какая генерация точек соответствует такому распределению

Проблема ММП:

$$\sum_{i=1}^m \log \left(\sum_{t=1}^k \pi_t \text{norm}(x_i \mid \mu_t, \Sigma_t) \right) \rightarrow \max$$

всё сокращается только при k=1

Gaussian Mixture Model (GMM)

GMM – универсальный аппроксиматор плотности
(если можно делать много гауссиан)

Решаем задачу нечёткой кластеризации в частном случае
«разделение смеси гауссиан»

проблемы оптимизации

- невыпуклость (как всегда)
- должна быть инвариантность к перестановкам

можно применять SGD, трюк: $\Sigma_t = M_t M_t^T$

чтобы матрица была положительно определённой
но всё равно м.б. проблемы <https://arxiv.org/pdf/1506.07677.pdf>

Обучение GMM

$$\sum_{i=1}^m \log \left(\sum_{t=1}^k \pi_t \text{norm}(x_i | \mu_t, \Sigma_t) \right) \rightarrow \max$$

0) случайная инициализация параметров: $\{\pi_t, \mu_t, \Sigma_t\}_{t=1}^k$

1) Повторять до сходимости

1.1 – **Е-шаг**) по текущим параметрам вычислить:

$$\gamma_{it} = \frac{\pi_t \text{norm}(x_i | \mu_t, \Sigma_t)}{\sum_j \pi_j \text{norm}(x_i | \mu_j, \Sigma_j)}$$

~ вероятность i-й объект в t-м кластере

1.2 – **М-шаг**) по γ_{it} пересчитать параметры кластеров

$$m_t = \sum_{i=1}^m \gamma_{it}$$

объёмы

$$\mu_t = \frac{1}{m_t} \sum_{i=1}^m \gamma_{it} x_i$$

центры

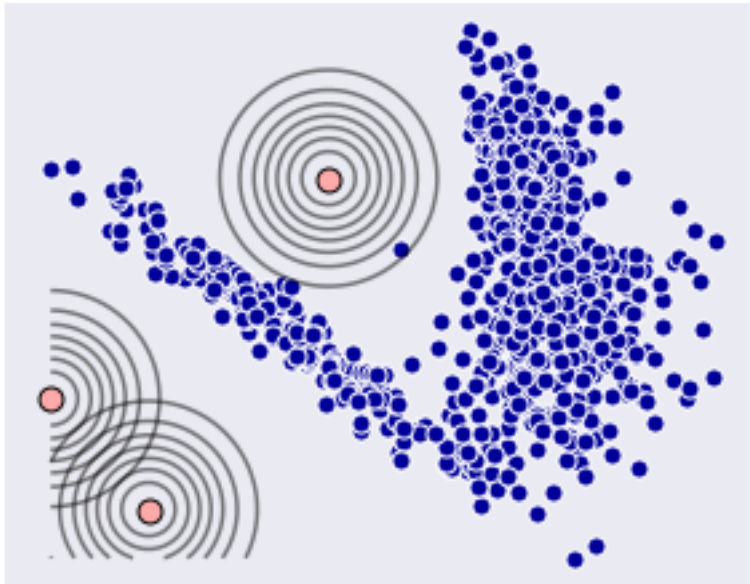
$$\Sigma_t = \frac{1}{m_t} \sum_{i=1}^m \gamma_{it} (x_i - \mu_t)(x_i - \mu_t)^T$$

формы

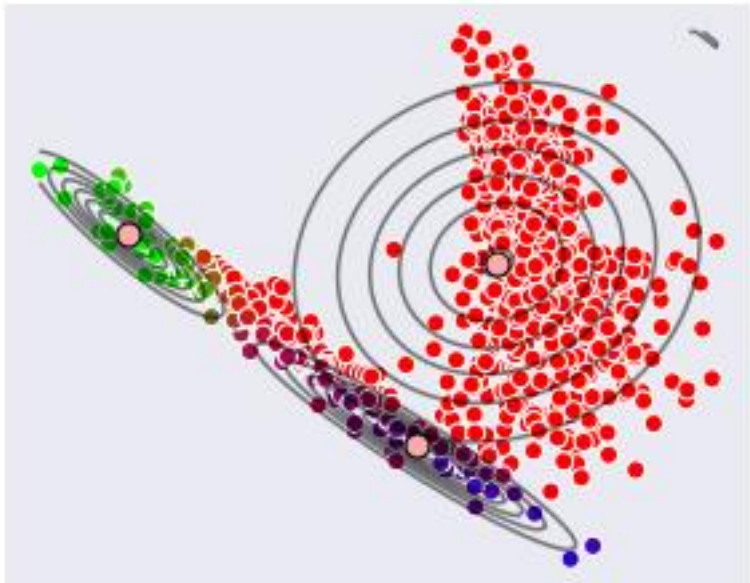
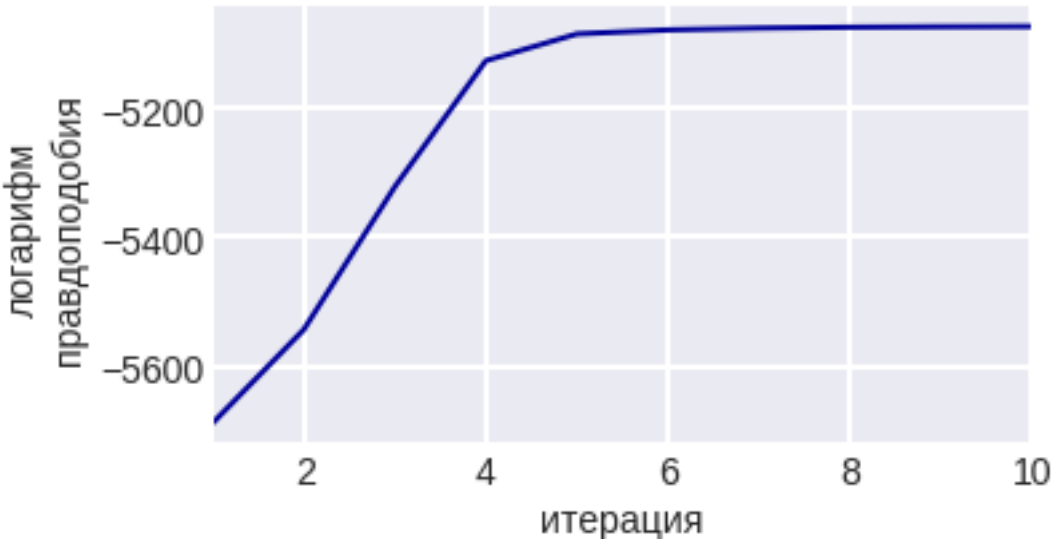
$$\pi_t = \frac{m_t}{m}$$

вероятности

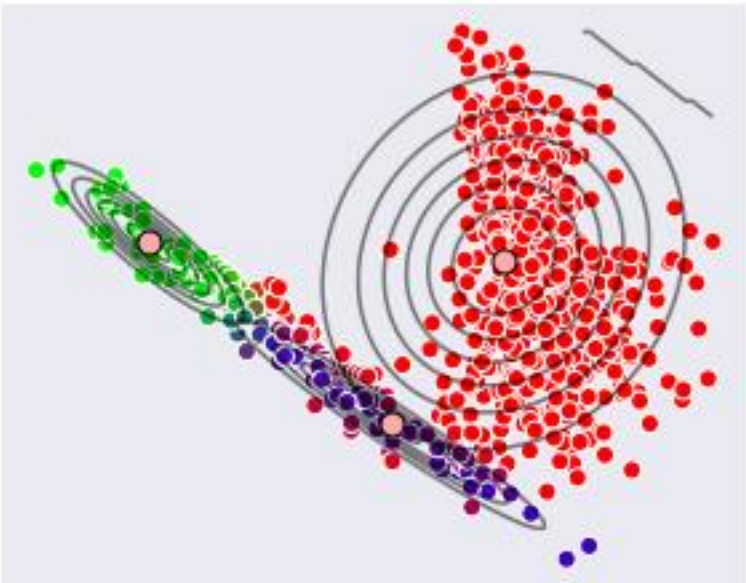
ЕМ: эксперименты



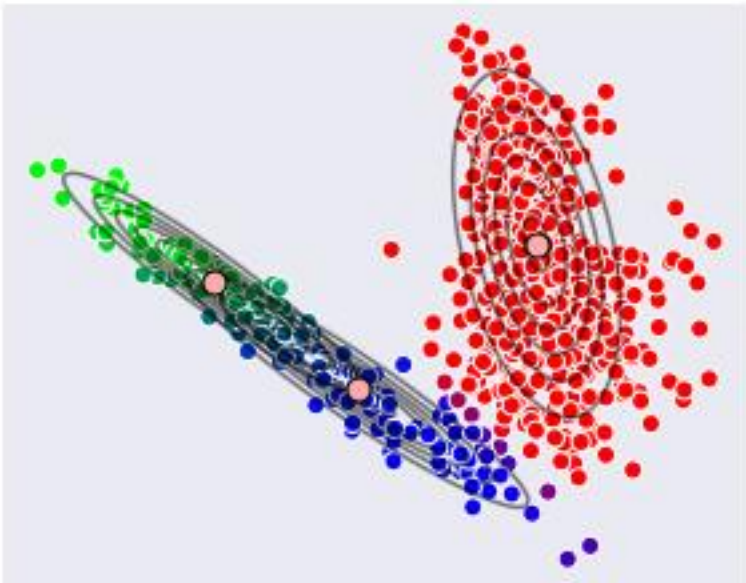
начальное приближение



итерация 1



итерация 2

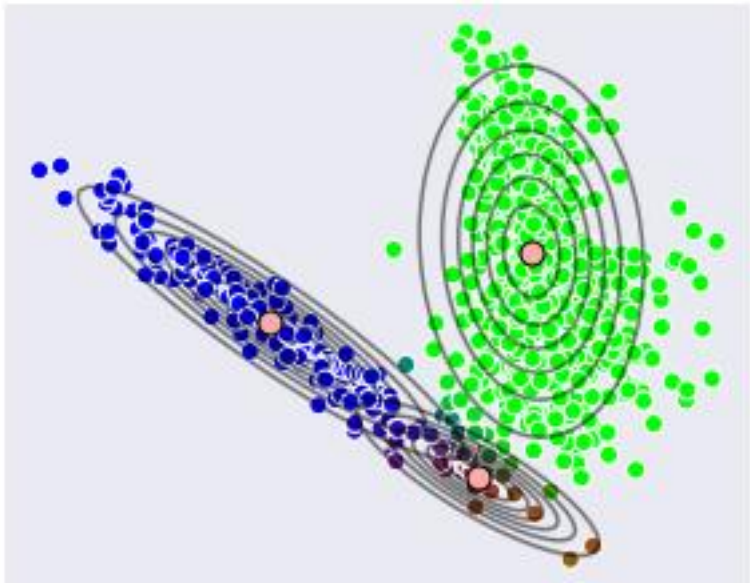


итерация 10

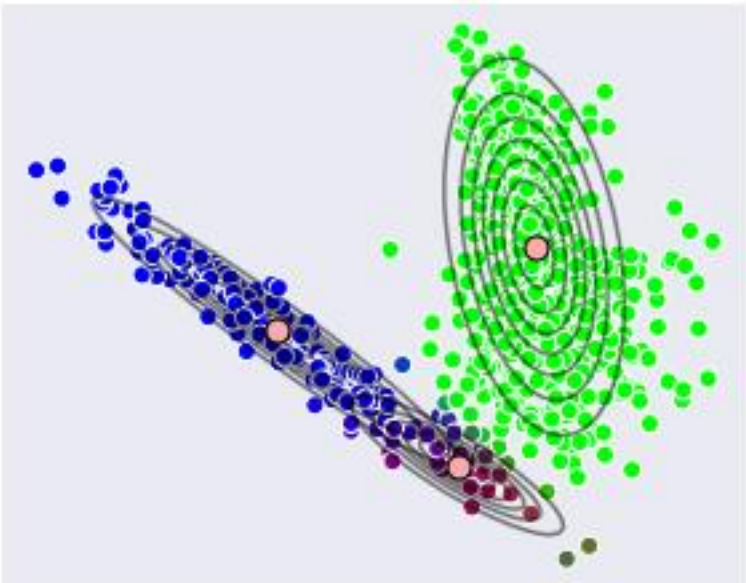
ЕМ: эксперименты



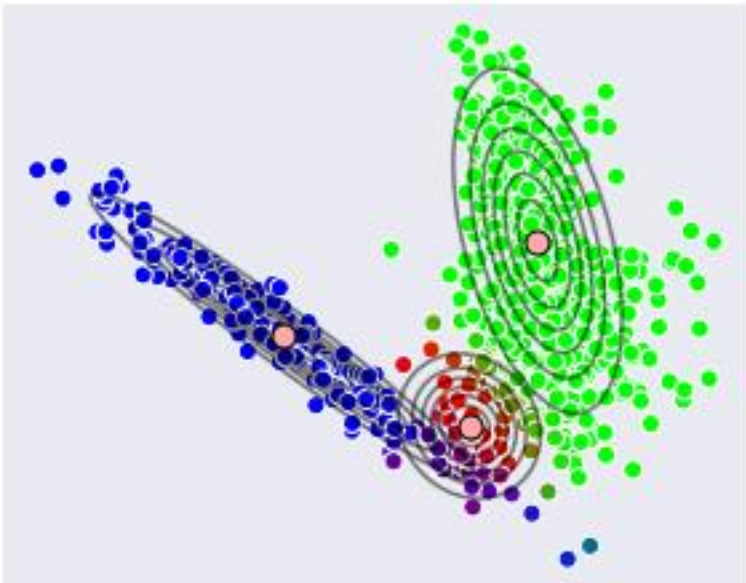
начальное приближение



итерация 1

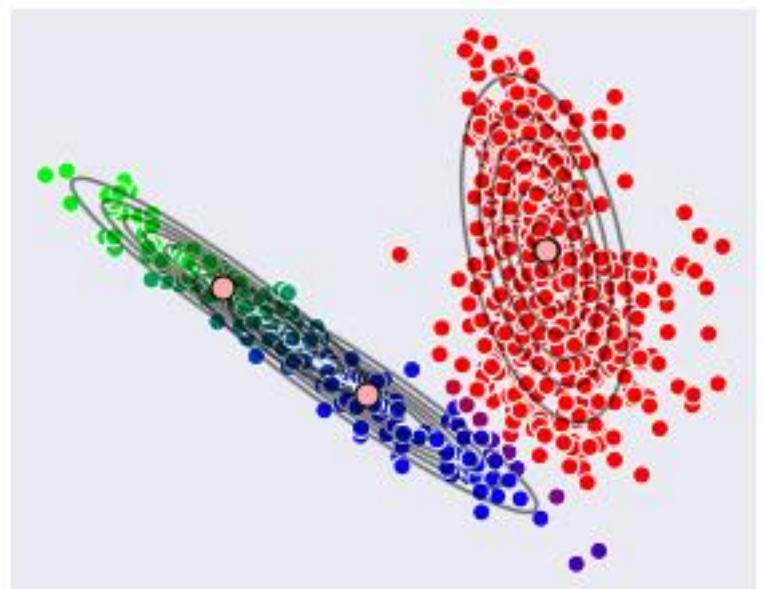
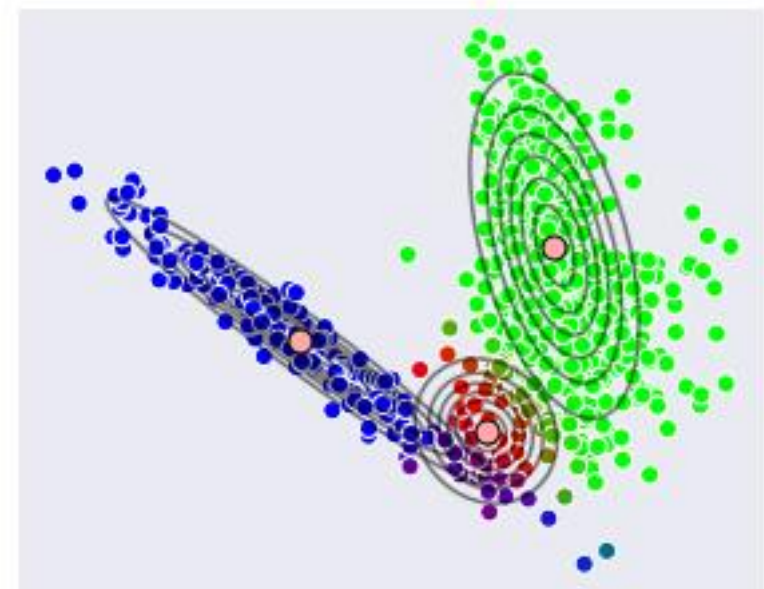
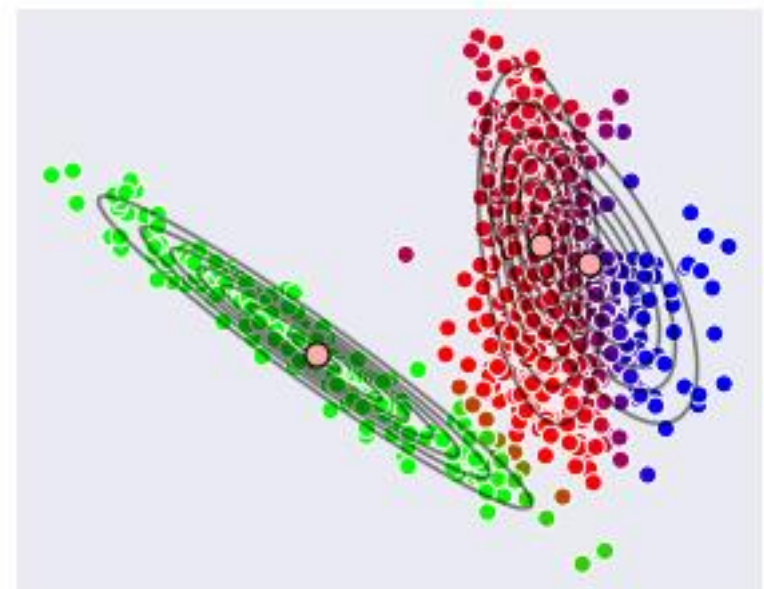
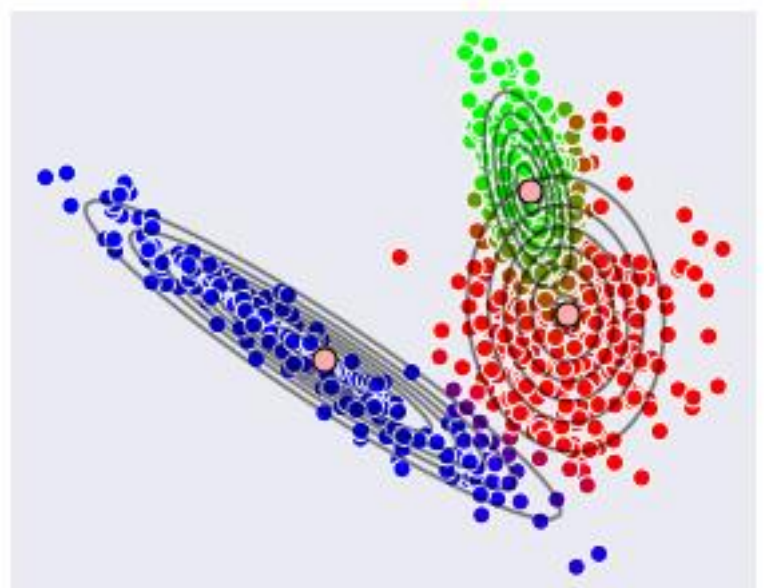
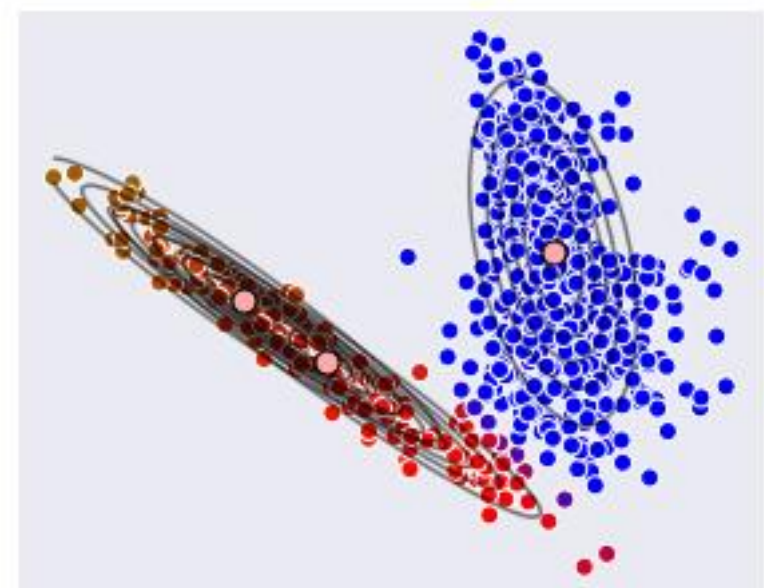
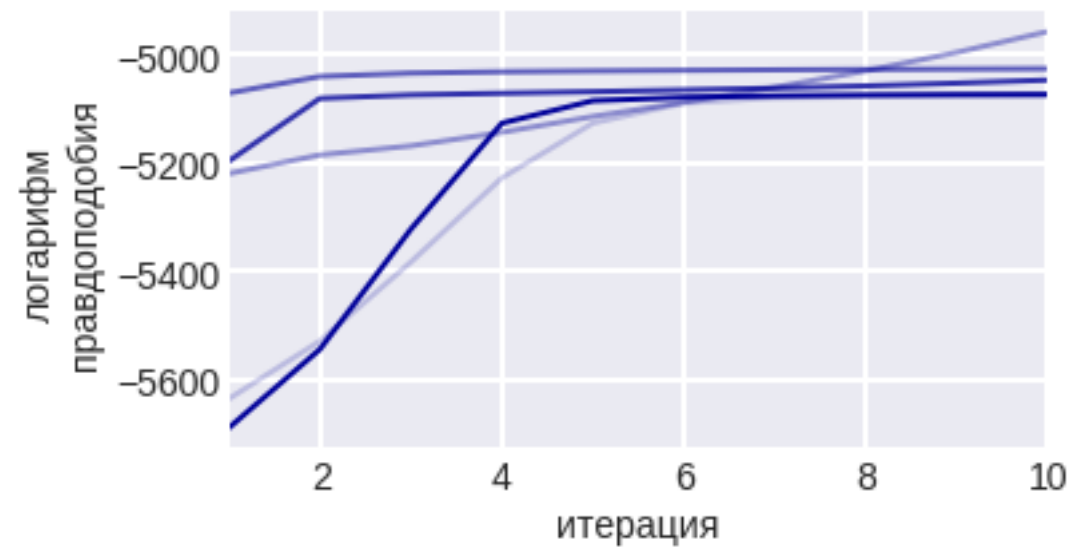


итерация 2

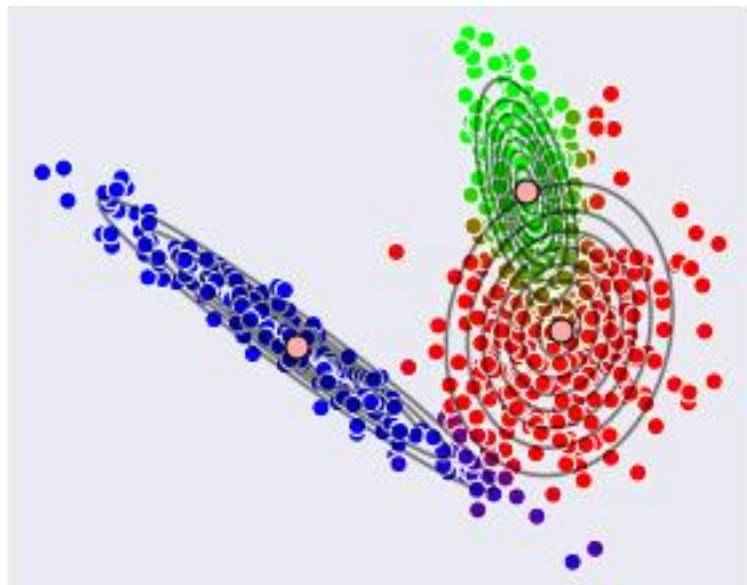


итерация 10

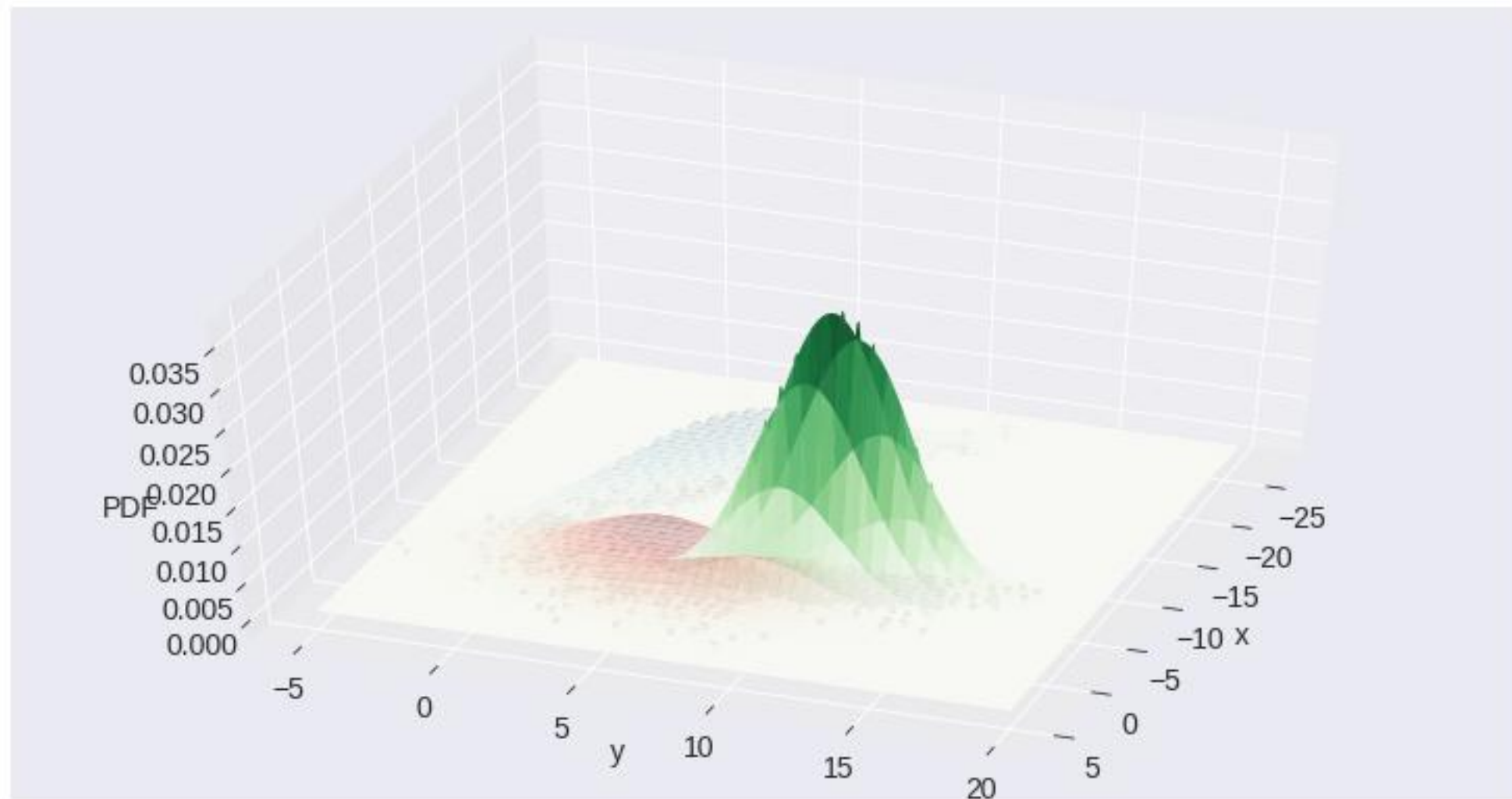
ЕМ: разные результаты работы алгоритма



ЕМ: результат



лучшее правдоподобие



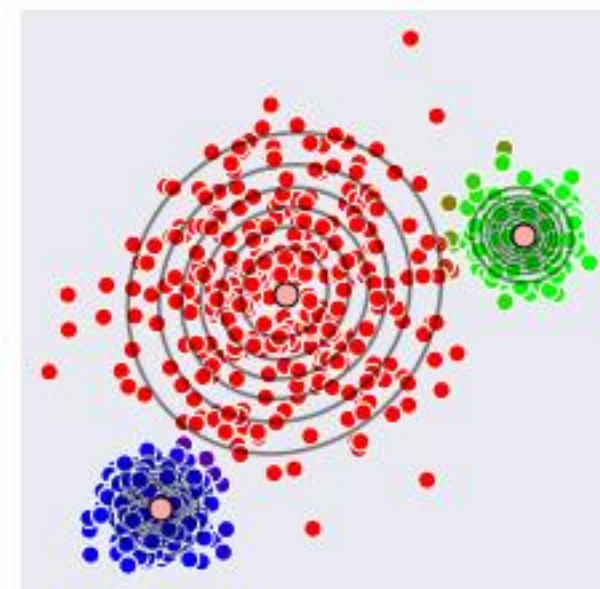
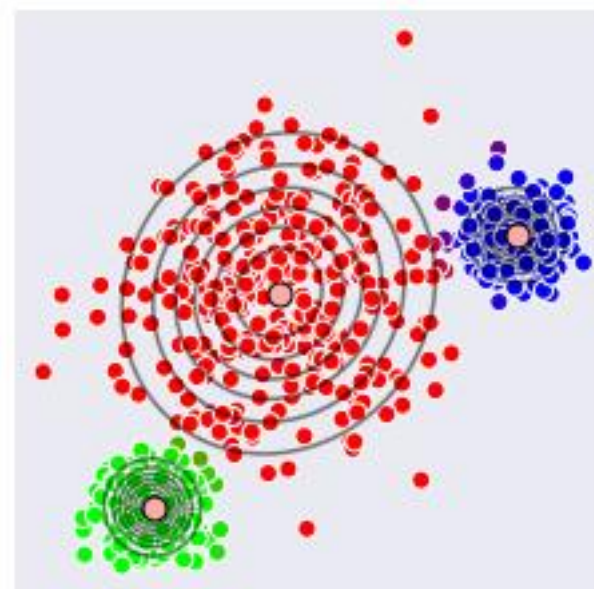
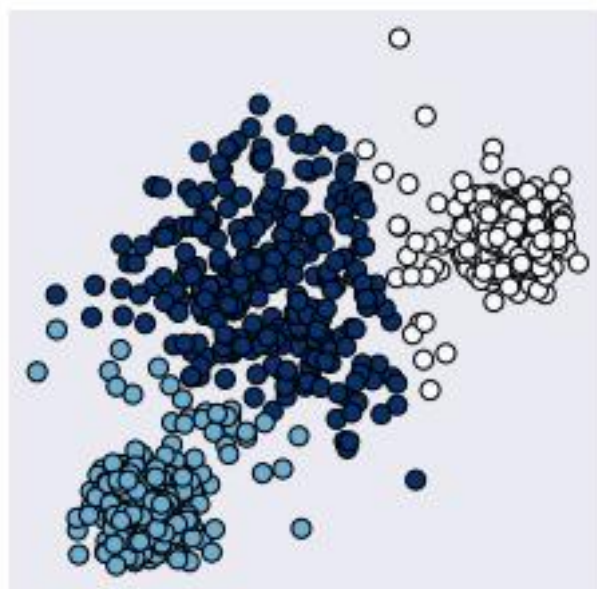
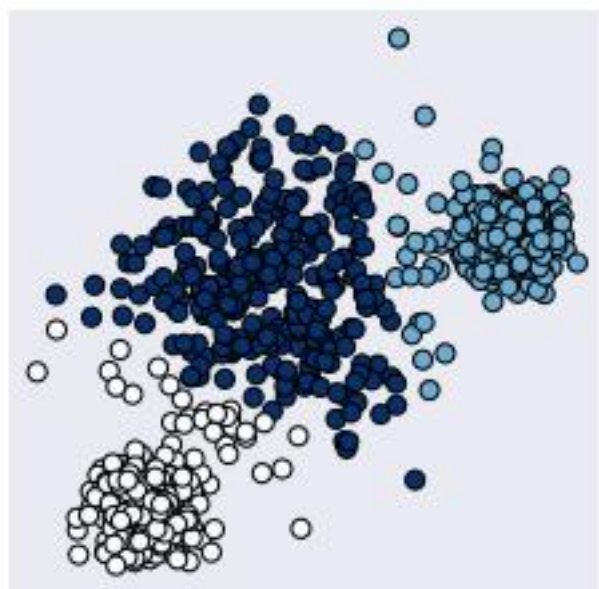
– можно не угадать с числом гауссиан

Связь GMM и k-means

наш ЕМ-алгоритм превращается в **soft-k-means**,
неявно возникает расстояние Махаланобиса, когда

- **распределения нормальные**
 - **кластеры равновероятны (equal priors)**
 - **ковариационные матрицы $\Sigma_t = \varepsilon I$**
- **если чёткая кластеризация (на Е-шаге), то в k-means**

слева – k-means (плохо с разными по размерам кластерами), справа – GMM



Минутка кода

`sklearn.mixture.GaussianMixture`

`n_components` – **число компонент (1)**

`covariance_type` – **формы**, `full` – у каждой компоненты своя ковариационная матрица,
`tied` – **одна матрица на всех**, `diag` – у каждой компоненты своя диагональная матрица,
`spherical` – у каждой компоненты своя дисперсия

`tol` – **порог для остановки**

`reg_covar` – **добавка к диагоналям матриц ковариаций**

`max_iter` – **число итераций**

`n_init` – **число инициализаций (рестартов 1)**

`init_params` – **как делать инициализацию** `kmeans` или `random`

`weights_init` – **ручная инициализация (веса объектам по компонентам)**

`means_init` – **ручная инициализация средних**

`precisions_init` – **ручная инициализация обратных матриц ковариации**

`random_state` –

`warm_start` –

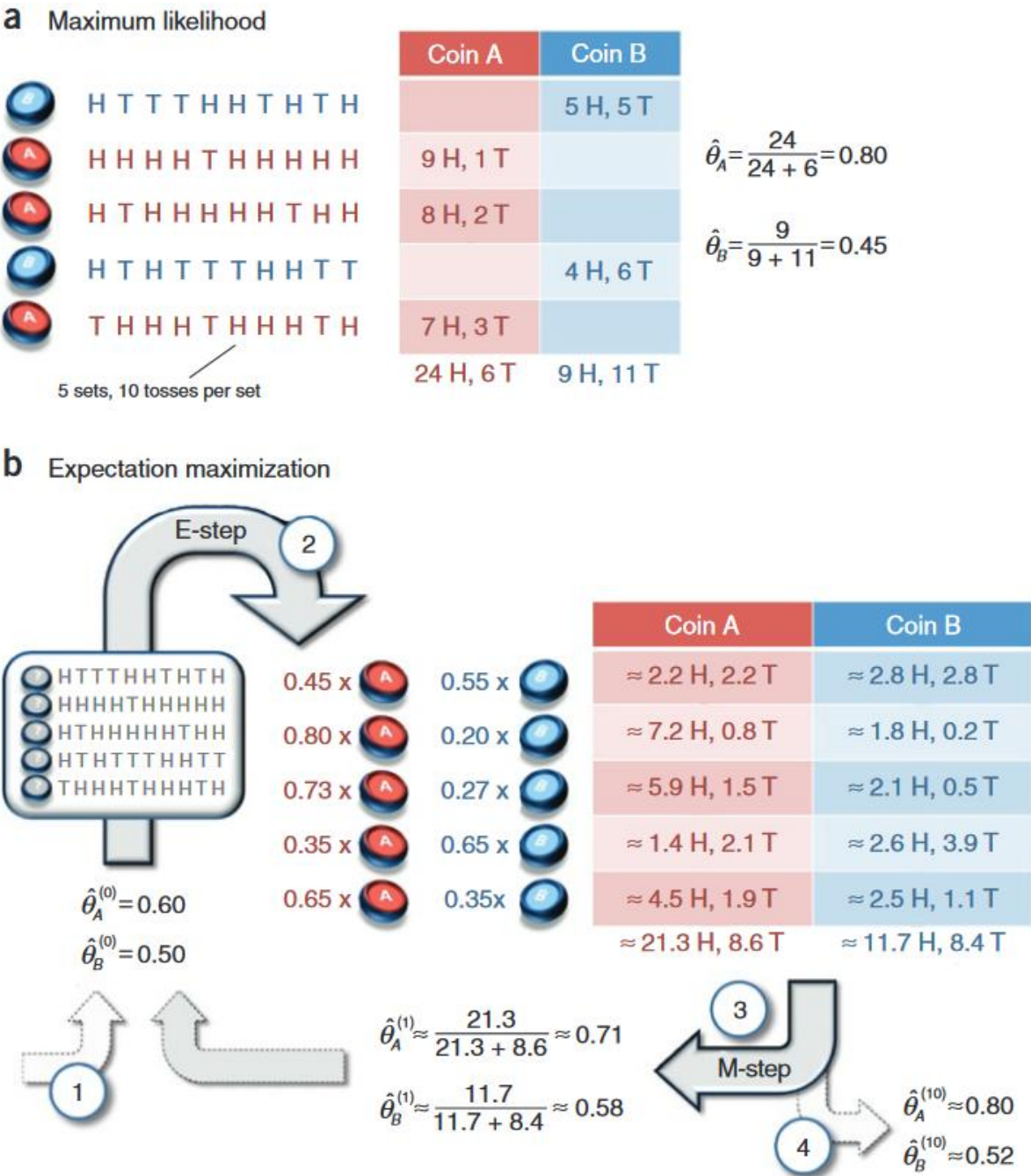
`verbose` –

`verbose_interval` –

GMM – итоги

- + быстрый алгоритм**
- + понятная геометрия**
- + естественное обобщение k-means**
- более универсальный**
- конкретный вид распределений**
- в своих популярных реализациях**
- число компонент задаётся вручную**
- можно проверить адекватность на отложенной выборке**

Пример ЕМ-алгоритма в простой модельной задаче



Пример ЕМ-алгоритма в простой модельной задаче

**Есть две нечестные монеты, 5 экспериментов с ними:
выбирается одна монета ($P=0.5$), подбрасывается 10 раз
Задача: оценить вероятности выпадения орлов у монет**

Пусть сначала оценки: $p_A = 0.6$, $p_B = 0.55$

Е-шаг

**Рассмотрим серию $S = \text{«НТТТННТНТН»} = 5H + 5T$
для монеты **A** вероятность серии $\sim p_A^5(1-p_A)^5 \approx 0.0008$
для монеты **B** вероятность серии $\sim p_B^5(1-p_B)^5 \approx 0.001$**

вероятность, что первая серия получена монетой A

$$P(S \leftarrow A) = \frac{p_A^5(1-p_A)^5}{p_A^5(1-p_A)^5 + p_B^5(1-p_B)^5} \approx 0.44$$

Пример ЕМ-алгоритма в простой модельной задаче

М-шаг – пересчитываем вероятности выпадения орла:

$$p_A = \frac{P(S_1 \leftarrow A) \cdot \#H_1 + P(S_2 \leftarrow A) \cdot \#H_2 + \dots}{P(S_1 \leftarrow A) \cdot \#(H + T)_1 + P(S_2 \leftarrow A) \cdot \#(H + T)_2 + \dots}$$

$$p_A = \frac{0.44 \cdot 5 + \dots}{0.44 \cdot 10 + \dots}$$

$$p_B = \frac{P(S_1 \leftarrow B) \cdot \#H_1 + P(S_2 \leftarrow B) \cdot \#H_2 + \dots}{P(S_1 \leftarrow B) \cdot \#(H + T)_1 + P(S_2 \leftarrow B) \cdot \#(H + T)_2 + \dots}$$

– взвешенная модификация MLE для уточнения параметров

здесь $\#(H + T)_t$ – число бросков в t -й серии S_t

$\#H_t$ – число орлов в ней

взято из... <https://www.nature.com/articles/nbt1406?pagewanted=all>

Генеративные модели

Пусть данные $\{x_1, \dots, x_m\}$ порождаются следующим образом:

- 1) генерируется $z_t \sim p(z \mid \varphi)$
- 2) генерируется $x_t \sim p(x \mid z_t, \theta)$

x – наблюдаемая переменная (observed variable)

z – латентная переменная (hidden variable)

Latent Variable Model – вероятностная модель, в которой не все переменные наблюдаются

Ненаблюдаемые переменные:
латентные / скрытые (latent / hidden) variables

Генеративные модели

$$z_t \sim p(z | \varphi) \rightarrow x_t \sim p(x | z_t, \theta)$$

Learning problem – найти параметры распределений

Inference problem – **использовать**

что такое в генеративных моделях плотность:

$$p(x) = \sum_z p(x, z) = \sum_z p(x | z) p(z)$$

напоминает нам GMM

$$p(x | \theta, \varphi) = \sum_z p(x | z, \theta) p(z | \varphi)$$

при выборе параметров $\varphi = \pi$, $\theta = (\mu, \Sigma)$

Генеративные модели

логарифм правдоподобия

$$\sum_x \log p(x | \theta, \varphi) = \sum_x \log \left(\sum_z p(x | z, \theta) p(z | \varphi) \right)$$

трудность применения MLE/MAP – максимизировать правдоподобие затруднительно:

Поэтому метод – ЕМ, его идея:

вместо суммирования по всем z пытаемся

для каждого x_t угадать, какое z_t ему соответствует,

если бы знали

$$\sum_t \log p(x_t | z_t, \theta) p(z_t | \varphi) = \sum_t \log p(x_t | z_t, \theta) + \sum_t \log p(z_t | \varphi) \rightarrow \max$$

находим параметры θ, φ , потом используем их чтобы снова угадать z_t и так по циклу

тонкость – дальше – взвешенное правдоподобие

Обоснование ЕМ (в общем виде)

Смесь произвольных распределений

$$p(x) = \sum_j \pi_j p(x | \Theta_j), \sum_{t=1}^k \pi_t = 1, \pi_t \geq 0$$

**чтобы не работать с правдоподобием,
вводим скрытые переменные и выписываем полное правдоподобие**

$$\prod_x \prod_z p(x, z | \dots) = \prod_i \prod_j \pi_j^{z_{ij}} p(x_i | \Theta_j)^{z_{ij}}$$

здесь бинарные скрытые переменные описывают принадлежность к компонентам

$$z_{ij} = I[x_i \sim p(x | \Theta_j)]$$

для каждого x определён $z = (z_1, \dots, z_k)$

$$\underbrace{\mathbf{P}(z_{ij} = 1)}_{\gamma_{ij}} = \mathbf{P}(z_j = 1 | x_i) = \frac{\mathbf{P}(z_j = 1) p(x_i | z_j = 1)}{\sum_t \mathbf{P}(z_t = 1) p(x_i | z_t = 1)} = \frac{\pi_j p(x_i | \Theta_j)}{\sum_t \pi_t p(x_i | \Theta_t)}$$

Обоснование ЕМ

Чем лучше полное правдоподобие...

$$\log \prod_i \prod_j \pi_j^{z_{ij}} p(x_i | \Theta_j)^{z_{ij}} = \sum_i \sum_j z_{ij} (\log \pi_j + \log p(x_i | \Theta_j))$$

здесь логарифм и сумма поменялись местами

теперь возьмём матожидание

$$\mathbf{E}_z \sum_i \sum_j z_{ij} (\log \pi_j + \log p(x_i | \Theta_j)) = \sum_i \sum_j \gamma_{ij} (\log \pi_j + \log p(x_i | \Theta_j))$$

в распределении Бернулли матожидание совпадает с вероятностью

теперь будем оптимизировать полученное взвешенное правдоподобие

Обоснование ЕМ что выяснили...

$$\gamma_{ij} = \frac{\pi_j p(x_i | \Theta_j)}{\sum_t \pi_t p(x_i | \Theta_t)}$$

оценка принадлежности

$$J = \sum_i \sum_j \gamma_{ij} (\log \pi_j + \log p(x_i | \Theta_j)) \rightarrow \max$$

оптимизация взвешенного правдоподобия

Оптимизация условная:

$$\frac{\partial}{\partial \pi_t} \left[J - \lambda \left(\sum_j \pi_j - 1 \right) \right] = \sum_i \frac{\gamma_{ij}}{\pi_j} - \lambda = 0$$

$$\pi_j = \sum_i \frac{\gamma_{ij}}{\lambda}$$

учитывая условия нормировки:

$$\pi_j = \frac{\sum_i \gamma_{ij}}{\sum_j \sum_i \gamma_{ij}} = \frac{1}{m} \sum_i \gamma_{ij} = \frac{m_t}{m}$$

$\equiv m_t$

до сих пор не было предположений относительно распределений

Обоснование ЕМ – для гауссиан

$$J = \sum_i \sum_j \gamma_{ij} \left(\log \pi_j + \underbrace{\log p(x_i | \Theta_j)}_{\text{norm}(x_i | \mu_j, \Sigma_j)} \right) \rightarrow \max$$
$$J \propto \sum_i \sum_j \gamma_{ij} (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)$$

дифференцируем по параметру (тут безусловная оптимизация) и =0

$$\sum_i \gamma_{ij} \Sigma_j^{-1} (x_i - \mu_j) = 0$$

$$\sum_i \gamma_{ij} x_i = \sum_i \gamma_{ij} \mu_j$$

$$\mu_j = \frac{\sum_i \gamma_{ij} x_i}{\sum_i \gamma_{ij}} = \frac{1}{m_t} \sum_i \gamma_{ij} x_i$$

в рамках обосновали все формулы пересчёта в ЕМ
(аналогично с матрицами ковариаций – не будем)

Обоснование ЕМ – для распределения Бернулли

$$J = \sum_i \sum_j \gamma_{ij} \left(\log \pi_j + \log \underbrace{p(x_i | \Theta_j)}_{\text{Bernoulli}(x_i | \mu_j)} \right) \rightarrow \max$$

$$J \propto \sum_i \sum_j \gamma_{ij} \log(\mu_j^{x_i} (1 - \mu_j)^{1-x_i}) = \sum_i \sum_j \gamma_{ij} (x_i \log \mu_j + (1 - x_i) \log(1 - \mu_j))$$

дифференцируем по параметру (тут безусловная оптимизация) и =0

$$\sum_i \left(\gamma_{ij} \frac{x_i}{\mu_j} - \gamma_{ij} \frac{(1 - x_i)}{(1 - \mu_j)} \right) = 0$$

$$\frac{\sum_i \gamma_{ij} x_i}{\mu_j} = \frac{\sum_i \gamma_{ij} x_i (1 - x_i)}{1 - \mu_j}$$

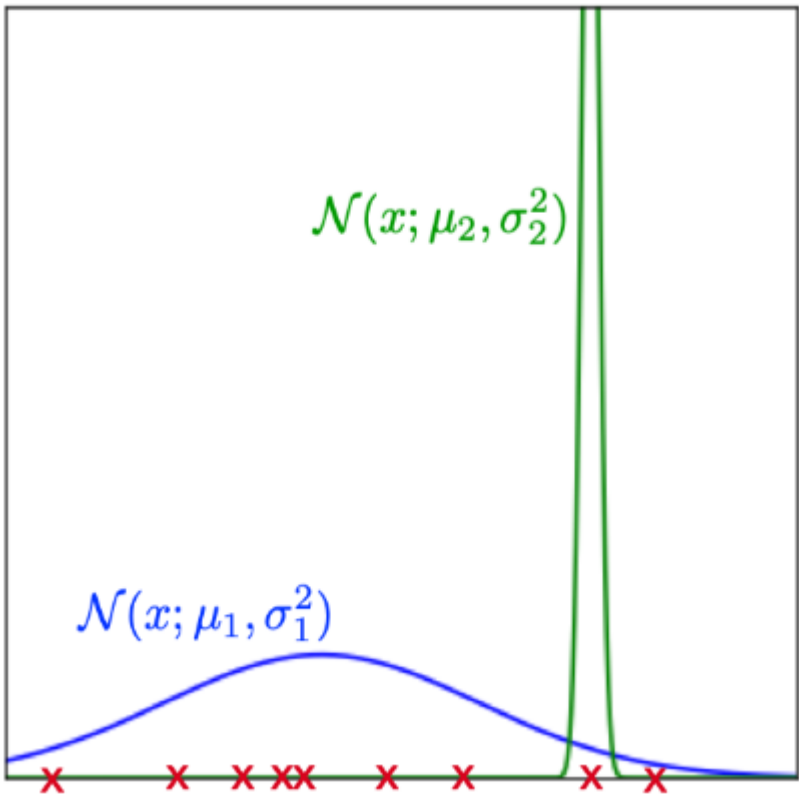
$$\mu_j = \frac{\sum_i \gamma_{ij} x_i}{\sum_i \gamma_{ij}} = \frac{1}{m_t} \sum_i \gamma_{ij} x_i$$

такая же формула

ЕМ-алгоритм

- + для любых смесей распределений, а дальше – общий подход – универсальная идея**
- но надо их знать априорно**
- + получаем оценки вероятностей (интерпретация)**
- + можно брать инициализацию из k-means**
- в отличии от него – произвольная форма и вероятность кластеров**
- не всегда задача корректна (кластер с маленькой дисперсией в центре точки выборки сколь угодно увеличивает правдоподобие)**
- это называется сингулярность – см кн. Бишопа**
- + шаги организованы так, что не уменьшается правдоподобие**
- по свойствам похож на k-means**
- тоже сильно зависит от инициализации**
- тоже задаётся число компонент (можно по значению правдоподобия настраивать)**
- не всегда практичный**

Сингулярность



Дальше дополнительный материал

Нижняя оценка Marginal Log-Likelihood (Evidence)

стандартный приём в ML:

$$\sum_i \log p(x_i | \theta, \varphi) = \sum_i \log \left(\sum_j \frac{p(x_i, z = j | \Theta) q_j}{q_j} \right) \geq \sum_i \sum_j q_j \log \frac{p(x_i, z = j | \Theta)}{q_j}$$

здесь z принимает дискретные значения

\geq – воспользовались неравенством Йенсена (Jensen) для выпуклых функций

$$f(\mathbf{E}[x]) \geq \mathbf{E}[f(x)]$$

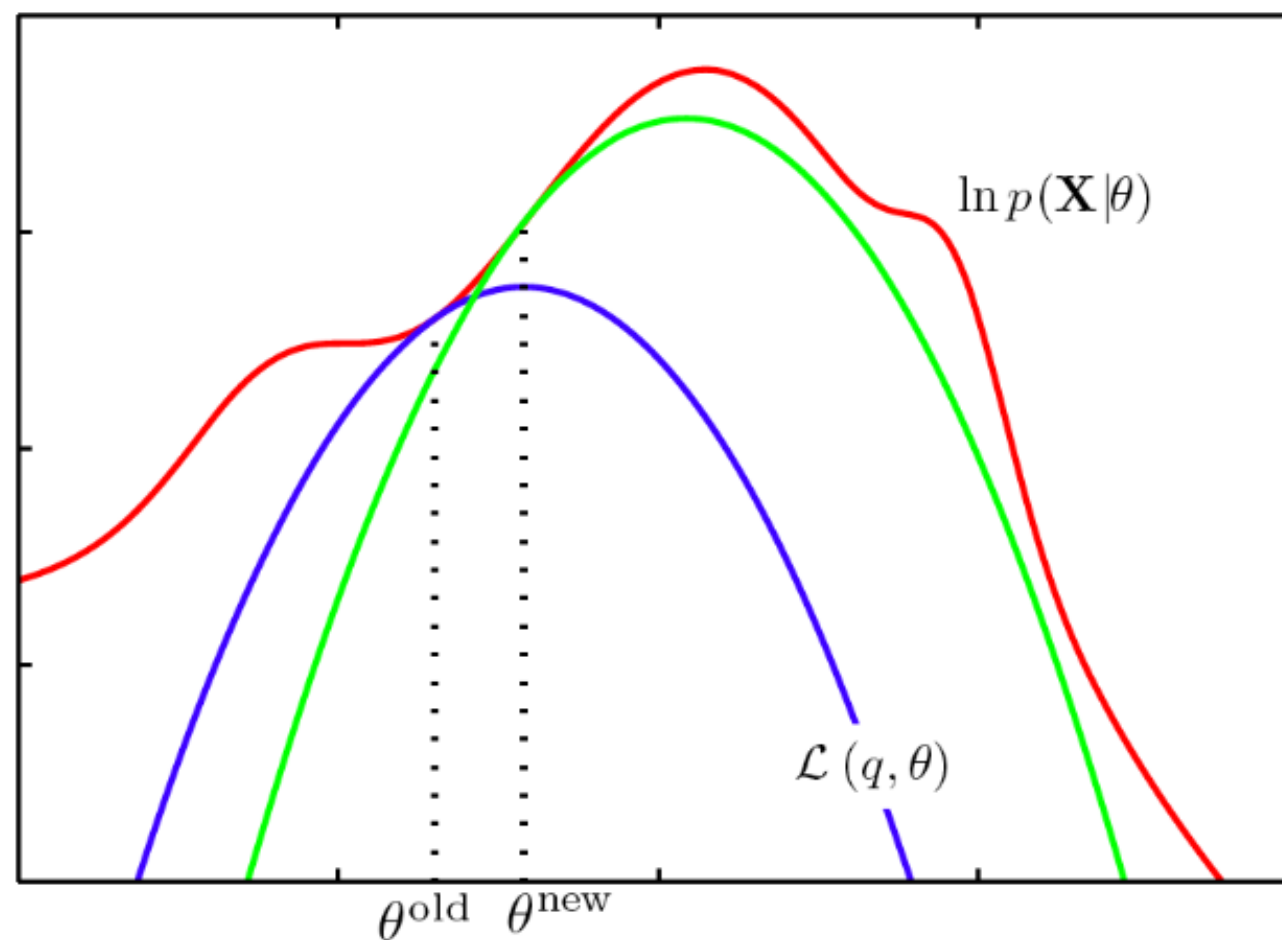
q – произвольное распределение на значениях z

можно максимизировать правую часть – ELBO – evidence lower bound – оценку левой

$$\text{MLE: } \hat{\Theta}_{\text{MLE}} = \arg \max \sum_i \log p(x_i | \Theta)$$

$$\text{EM: } \hat{\Theta}_{\text{EM}} = \arg \max \sum_i \sum_j q_j \log \frac{p(x_i, z = j | \Theta)}{q_j}$$

Нижняя оценка Marginal Log-Likelihood (Evidence)



**ЕМ вычисляет ELBO для текущих параметров
и максимизирует её для получения новых параметров**

Bishop Pattern recognition and machine learning, Figure 9.14

ЕМ «на верхнем уровне»

0. Инициализация параметров Θ

1. Повторять до сходимости

1.1. Выбор

$$q \leftarrow \arg \max_q \sum_i \sum_j q_j \log \frac{p(x_i, z = j | \Theta)}{q_j}$$

доказывается (см. дальше), что для i $q = p(z | x_i, \Theta)$

обратим внимание, что q – распределение значений z

1.2. Выбор

$$\Theta \leftarrow \arg \max_{\Theta} \sum_i \sum_j q_j \log \frac{p(x_i, z = j | \Theta)}{q_j}$$

Есть теорема о сходимости при необременительных условиях

Florin Vaida «Parameter Convergence for EM and MM Algorithms» // Statistica Sinica, 2005,
<http://www3.stat.sinica.edu.tw/statistica/oldpdf/a15n316.pdf>

Кстати, связь с KL-дивергенцией

$$\begin{aligned}\sum_z q(z) \log \frac{p(x, z | \Theta)}{q(z)} &= \sum_z q(z) \log \frac{p(z | x, \Theta) p(x | \Theta)}{q(z)} = \\ &= \sum_z q(z) \log \frac{p(z | x, \Theta)}{q(z)} + \sum_z q(z) \log p(x | \Theta) = \\ &= -\text{KL}[q(z), p(z | x, \Theta)] + \log p(x | \Theta)\end{aligned}$$

вот почему там оптимальный выбор $q = p(z | x_i, \Theta)$

Ссылки

Bishop C. M. Pattern recognition and machine learning. – Springer, 2006

Неплохие курсы с объяснением ЕМ-алгоритма

<http://www.cs.toronto.edu/~rgrosse/teaching.html>