ФКН

Департамент больших данных и информационного поиска

Москва 2025

Лекция 8
ЕМ-алгоритм

Машинное обучение в цифровом продукте

Полякова И.Ю.

Пример

- Пусть преподавателю на проверку пришли работы из двух разных групп;
- При этом абсолютно все студенты забыли подписать свою работу;
- Преподавателю при этом нужно оценить уровень знаний студентов в каждой из групп

Для этого желательно бы знать номер группы студента (1 или 2), а также оценить параметры распределения в двух разных группах (средний уровень знаний и их разброс, например)

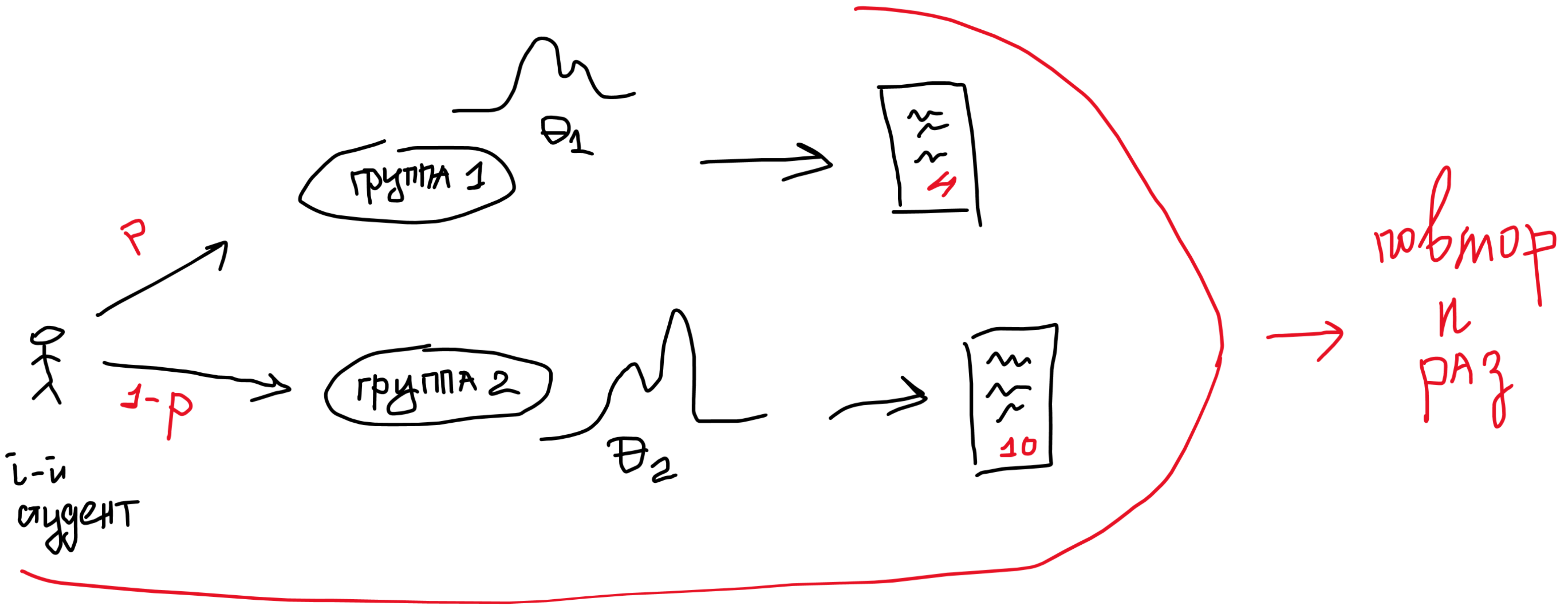
Как это сделать?

Пример

Попробуем придумать модель, описывающую «генерацию» каждой конкретной работы

1. «Природа» выбирает, в какой из групп учится студент (случайным образом);
2. Группы отличаются своим составом, силой знаний, преподавателями, поэтому работа конкретного студента «генерируется» из распределения группы n с параметрами θ_n

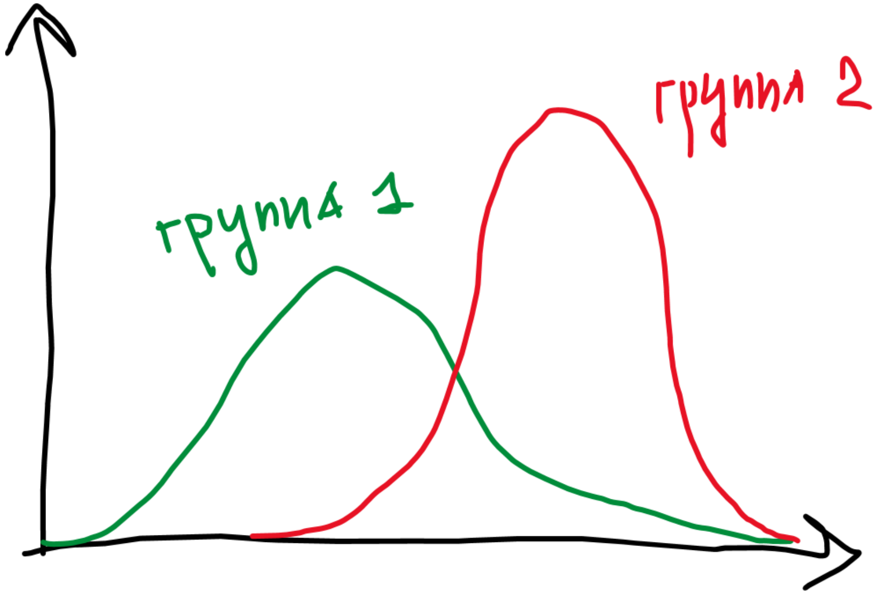
Пример



i-й студент
p
1-p
группа 1
группа 2
θ_1
θ_2
4
10
повтор n раз

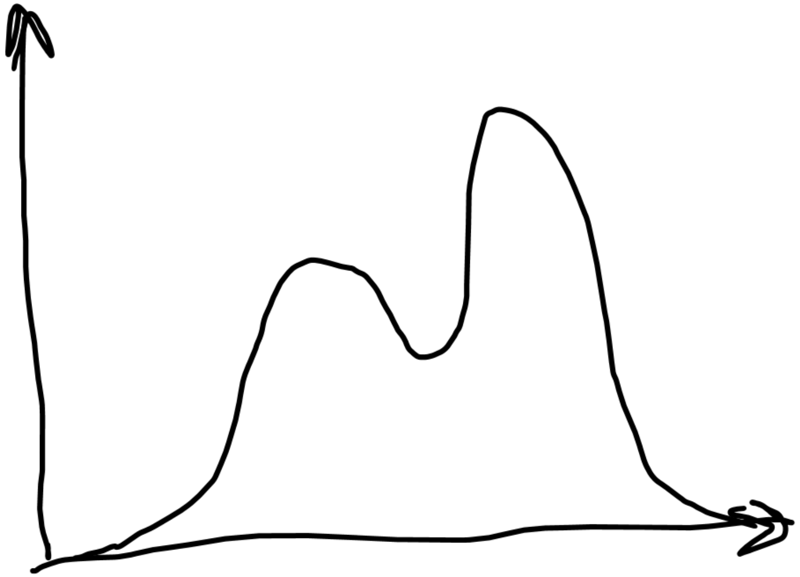
Пример

На выходе:



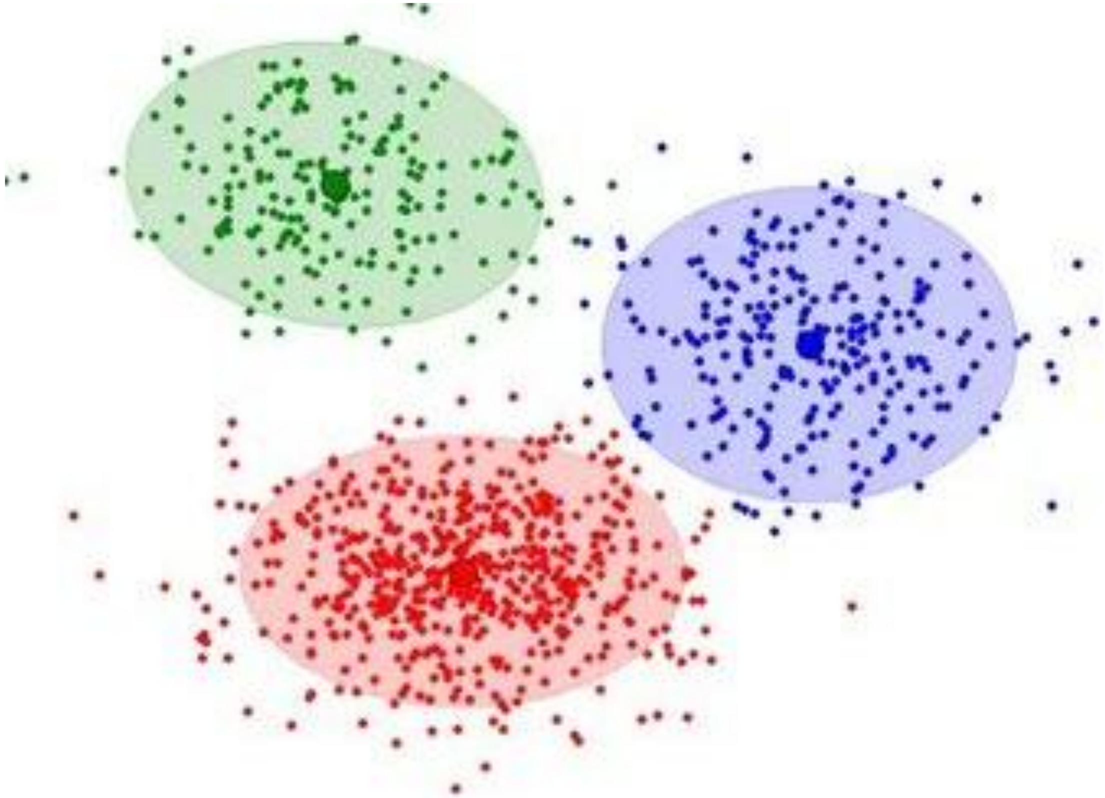
НО!

По факту видим:



Пример

Или в двумерном пространстве увидим что-то такое:



Модель смеси распределений

Mixture Model

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

K – число компонент в смеси
π_k – вероятность компоненты
$p_k(x)$ – распределение k-й компоненты смеси

Модель смеси распределений

Неполное правдоподобие или «ММП в лоб»

$$\sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \cdot \varphi(x \mid \theta_k) \to \max_{\pi_k, \theta_k}$$

$$p(x \mid z_k = 1) = \varphi(x \mid \theta_k)$$

Проблема:

- Функция вида «логарифм суммы» неприятна в оптимизации из-за огромного количества локальных максимумов в отличие от «суммы логарифмов»
- Работать с такой функцией правдоподобия не вариант!

Модель со скрытыми переменными

На примере модели смеси распределений

Введем скрытую переменную z, отвечающую за выбор номера компоненты в смеси

$$z \in \{0, 1\}^K \quad , \quad \sum_{k=1}^{K} z_k = 1$$

one-hot вектор

Где тогда будет фигурировать π_k из прошлой записи?

Можно сказать, что π_k - это в-ть того, что единице будет равна k-я компонента вектора скрытых переменных

$$p(z_k = 1) = \pi_k$$

Тогда распределение всего вектора:

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$$

Модель со скрытыми переменными

На примере модели смеси распределений

Если номер компоненты смеси известен, то СВ x имеет распределение:

$$p(x \mid z_k = 1) = \varphi(x \mid \theta_k)$$

$$p(x \mid z) = \prod_{k=1}^{K} [\varphi(x \mid \theta_k)]^{z_k}$$

Запишем совместное распределение x и z:

$$p(x, z) = p(z) \cdot p(x \mid z) = \prod_{k=1}^{K} [\pi_k \cdot \varphi(x \mid \theta_k)]^{z_k}$$

Полная функция правдоподобия:

$$\log p(X, Z \mid \theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \cdot [\log \pi_k + \log \varphi(x_i \mid \theta_k)]$$

Полное правдоподобие

$$\log p(X, Z \mid \theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \cdot \left[\log \pi_k + \log \varphi(x_i \mid \theta_k) \right]$$

- Немного преобразовали математическую модель исходной задачи;
- Функция полного правдоподобия имеет классический вид суммы логарифмов, что приятно для оптимизации;
- Проблема: переменные z скрыты от наблюдателя, их знает только «природа»

- **Как оценивать параметры?**

Примечание: «природой» в теории игр называется игрок «из вне», решения которого случайны и на которые невозможно повлиять. В этой лекции «природа» как бы выбирает номер компоненты в смеси (в общем случае генерирует скрытые переменные)

Модификация ММП

Идея: так как оптимизация одновременно и скрытых переменных, и «обычных» параметров не представляется возможной, попробуем оптимизировать их поочередно

Алгоритм:

1. Берем начальное приближение «обычных» параметров θ^{OLD};
2. Оцениваем скрытые переменные, зная X и θ^{OLD}, с помощью ММП;

$$Z^* = \arg\max_Z P(Z \mid X, \theta^{OLD}) = \arg\max_Z P(X, Z \mid \theta^{OLD})$$

3. Оценив скрытые переменные, обновляем «обычные» параметры;

$$\theta^* = \arg\max_\theta P(X, Z^* \mid \theta)$$

4. Повторяем до сходимости

Модификация ММП

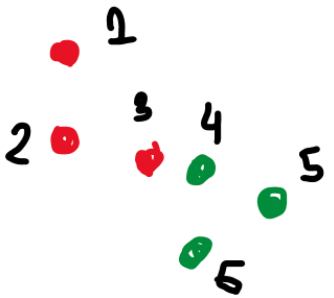
Проблема

- **Невозможно гарантировать «сходимость»** или что-то ее напоминающее для предложенного алгоритма;
- Плохо!
- Попробуем подойти к оценке скрытых переменных по-другому;
- А именно попробуем посчитать их **апостериорное распределение**, воспользовавшись формулой Байеса в случае модели смеси

априорное зн-е

природа

z_1	1	0
...		
z_3	1	0
z_4	0	1
...		
z_6	0	1



апостериорная оценка

z_1^*	0.95	0.05
...		
z_3^*	0.5	0.5
z_4^*	0.5	0.5
...		
z_6^*	0.05	0.95

наблюдатель

ЕМ-алгоритм

Алгоритм:

1. Берем начальное приближение «обычных» параметров θ^{OLD};
2. **Е-шаг:** вычисляем ***апостериорное распределение*** скрытых переменных, используя θ^{OLD};

$$p(z \mid x, \theta^{OLD})$$

3. **М-шаг:** усредняем логарифм полного правдоподобия по всем возможным значениям скрытых переменных с весами, равными апостериорным вероятностям этих значений (Q-функция). Максимизируем Q-функцию по θ;

$$\theta^{NEW} = \underset{\theta}{\operatorname{argmax}} \sum_z p(z \mid x, \theta^{OLD}) \log p(x, z \mid \theta)$$

4. Повторяем до сходимости

Апостериорное распределение

Общая формула: $$p(Z \mid X, \theta^{OLD}) = \frac{p(X, Z \mid \theta^{OLD})}{p(X \mid \theta^{OLD})}$$

Для смеси распределений: $$p(Z \mid X, \theta^{OLD}) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left[\pi_k^{OLD} \cdot \varphi(x_i \mid \theta^{OLD}) \right]^{z_{ik}}$$

Это распределение можно расписать как произведение распределений, соответствующих отдельным объектам

$$p(Z \mid X, \theta^{OLD}) = \prod_{i=1}^{n} p(z_i \mid x_i, \theta^{OLD})$$

Апостериорное распределение

- Скрытые переменные независимы при известной выборке объектов;
- Вектор скрытых переменных в случае модели смеси состоит из k значений (0 или 1);
- Запишем вероятности каждого из значений по формуле Байеса

Апостериорная в-ть принадлежности i-го объекта к k-му кластеру:

$$g_{ik} = p(z_{ik}=1 \mid x_i, \theta^{OLD}) = \frac{p(z_{ik}=1)\, p(x_i \mid z_{ik}=1, \theta^{OLD})}{\sum_{j=1}^{k} p(z_{ij}=1)\, p(x_i \mid z_{ij}=1, \theta^{OLD})} =$$

$$= \frac{\pi_k^{OLD} \cdot \varphi(x_i \mid \theta_k^{OLD})}{\sum_{j=1}^{k} \pi_j^{OLD} \cdot \varphi(x_i \mid \theta_j^{OLD})}$$

Формула Байеса

В «простой» записи

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Где:

P(A|B) – вероятность гипотезы A при наступлении события B;
P(A) – априорная вероятность гипотезы A
P(B|A) – вероятность наступления события B при истинности гипотезы A;
P(B) – полная вероятность наступления события B

Формула Байеса

Пример

Есть два завода, они производят одинаковые детали. Известно, что на первом произведено 6000 деталей, на втором 4000. Доля брака на первом составляет 10%, на втором – 20%.
Выбрали случайным образом деталь, она оказалась не бракованная. Какова вероятность, что она произведена на первом заводе? На втором заводе?

$N = 10000$

$n_1 = 6000$ $\quad P_{\text{не брак}} = 0.9$

$n_2 = 4000$ $\quad P_{\text{не брак}} = 0.8$

В-ть, что деталь произведена на станке 1: $\frac{6000}{10000} = 0,6$

В-ть, что деталь произведена на станке 2: $\frac{4000}{10000} = 0,4$

В-ть вытащить не бракованную деталь из всей совокупности:

$$0,6 \cdot 0,9 + 0,8 \cdot 0,4 = 0,86$$

В-ть, что не бракованная со станка 1: $\frac{0,6 \cdot 0,9}{0,86} \approx 0,63$

В-ть, что не бракованная со станка 2: $\frac{0,4 \cdot 0,8}{0,86} \approx 0,37$

EM-алгоритм

На примере модели гауссовой смеси

1. Инициализировали параметры;
2. На E-шаге оценили апостериорные вероятности g_{ik} по формуле Байеса;
3. Делаем M-шаг:

$$Q(\theta, \theta^{OLD}) = \sum_{i=1}^{n} \sum_{k=1}^{k} g_{ik} \left[\log \pi_k + \log N(x_i \mid \mu_k, \Sigma_k) \right] \to \max_{\pi_k, \mu_k, \Sigma_k}$$

Ищем частные производные по каждому из параметров (обратите внимание, что для поиска решения π_k придется использовать метод Лагранжа, так как $\sum \pi_k = 1$)

4. Получаем аналитические оценки искомых параметров:

$$\pi_k^{new} = \frac{1}{n} \sum_{i=1}^{n} g_{ik} \qquad \mu_k^{NEW} = \frac{1}{n\pi_k} \sum_{i=1}^{n} g_{ik} x_i \qquad \Sigma_k^{NEW} = \frac{1}{n\pi_k} \sum_{i=1}^{n} g_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

Подробнее: https://education.yandex.ru/handbook/ml/article/modeli-s-latentnymi-peremennymi

Что дает ЕМ-алгоритм?

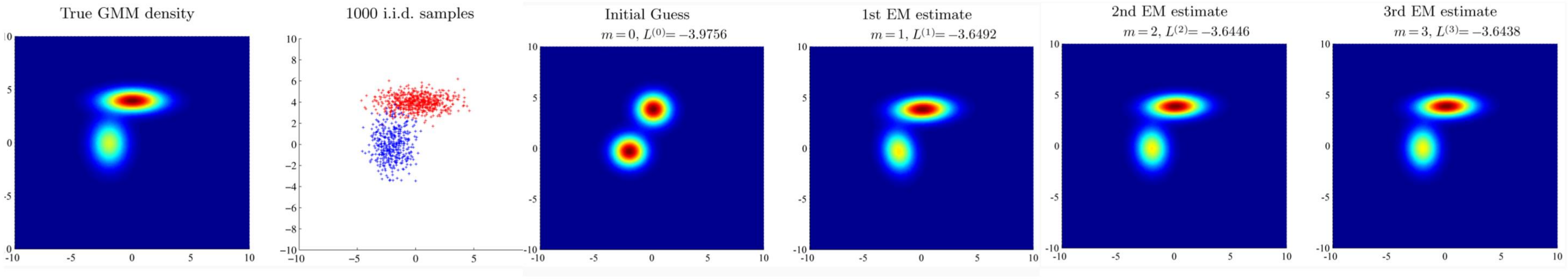
На прошлом слайде мы получили оценки параметров распределений, но что не менее важно, еще и g_{ik}, которые можно интерпретировать как принадлежность i-го объекта k-му кластеру, что дает возможность для:

- **Кластеризации**: при этом мы можем «играть» с видом распределений и их количеством кластеров (к сожалению, нам при запуске придется вручную задавать их кол-во);
- **Классификации**: однако, есть нюанс, так как в ЕМ-алгоритме мы имеем право на запуске «переставлять» номера кластеров, поэтому для задачи классификации обычно используют априорное знание о выборке;

Например, известно, что мальчиков в группе больше, чем девочек, поэтому, более часто прогнозируемому классу мы присвоим мужской пол

- Глобально ЕМ-алгоритм дает возможность применить оценки ММП к моделям со скрытыми переменными (и это касается далеко не только моделей смеси)

Пример работы кластеризации



Источник: https://logic.pdmi.ras.ru/~sergey/teaching/mlspsu17/15-em.pdf

Что дает ЕМ-алгоритм?

- Кластеризация с помощью модели гауссовой смеси (GMM), позволяет находить кластеры эллиптической формы и вычислять вероятности принадлежности точки к кластерам (что-то вроде продвинутого K-means);
- Заполнение пропущенных данных;
- Обучение скрытых марковских моделей (распознавание речи; анализ временных рядов);
- Тематическое моделирование (Topic Modeling): поиск скрытых тем в текстовых документах, определение ключевых тем в каждом из них

…

Подробнее:
1. https://scikit-learn.org/stable/modules/mixture.html
2. https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html
3. https://crowdsourcing-class.org/readings/downloads/ml/EM.pdf
4. http://www.machinelearning.ru/wiki/images/f/fb/Voron-ML-TopicModels.pdf

Про сходимость ЕМ-алгоритма

- Доказано, что правдоподобие с каждым шагом ЕМ-алгоритма не убывает;
- Оценки ЕМ-алгоритма сходятся к стационарной точке функции правдоподобия;
- Сходимость к локальному максимуму гарантируется только для некоторых семейств распределений (например, для экспоненциального)

За доказательствами сюда: https://github.com/esokolov/ml-course-hse/blob/master/2020-spring/lecture-notes/lecture15-em.pdf
Или сюда: https://education.yandex.ru/handbook/ml/article/modeli-s-latentnymi-peremennymi

Дополнительно

- Описание алгоритма на вики: https://clck.ru/3QZtb7
- Чуть подробнее есть здесь: https://education.yandex.ru/handbook/ml/article/modeli-s-latentnymi-peremennymi
- Конспект Евгения Соколова: https://github.com/esokolov/ml-course-hse/blob/master/2020-spring/lecture-notes/lecture15-em.pdf
- Про EM в Topic Modeling у Константина Воронцова (там есть и базовые выкладки, применимые не только в NLP): http://www.machinelearning.ru/wiki/images/f/fb/Voron-ML-TopicModels.pdf
- Немного про EM в Марковских цепях: https://logic.pdmi.ras.ru/~sergey/teaching/mlspsu17/15-em.pdf