Работа № 3.8. Кластеризация: алгоритмы K-means и EM

1. Теоретические сведения. Метод K-means

Был изобретён в 1950-х годах математиком Гуго Штейнгаузом и почти одновременно Стюартом Ллойдом. Особую популярность приобрёл после работы Маккуина.

- Изобретён в 1950-х годах
- Целевая функция - минимум суммы квадратов расстояний от точек до центров соответствующих им кластеров
- Смешанная (дискретно-непрерывная) задача оптимизации
- K-means – **эвристика**!
- Итерация алгоритма состоит из 2-х этапов
- Количество кластеров задаётся заранее

Описание алгоритма K-means

- **Инициализация**: алгоритм инициализируется центрами кластеров:
  - Первые k точек
  - Случайные k точек
  - Заранее определенные k точек
  - K-means++
  - Другиеалгоритмы (Random Partitioning, Build Algorithm…)
- **Шаг назначения**: известны центры кластеров. Распределение точек по ближайшим кластерам.
- **Шаг обновления**: пересчёт центров кластеров.

При инициализации точки как правило выбираются из набора данных. В некоторых случаях, точки могут быть заданы и не из набора данных, а как некоторые заранее определенные значения.

Если расстояние эвклидово – то среднее

Если Манхеттена – то медиана

K-means++ - каждый следующий центроид выбирается по вероятности – чем дальше расположена точка от текущих известных центров, тем выше вероятность ее выбора.

Назначение:

$$S_i^{t+1} = \{x_p : (x_p - \mu_i^t)^2 \leq (x_p - \mu_j^t)^2, \forall j \neq i\}$$

S_i– кластер с номером i, x– точки для кластеризации, μ– центры кластеров, t– номер шага.

Обновление средних:

$$\mu_i^{t+1} = \frac{1}{|S_i^{t+1}|} \sum_{x_j \in S_i^{t+1}} x_j$$

Оба шага таким образом уменьшают сумму квадратов расстояний.

Действительно, после шага 1 суммарное расстояние уменьшиться поскольку точки могли быть распределены в более дальние кластеры, таким образом их вклад может только уменьшиться. На втором шаге находится среднее значение всех точек, входящих в новый кластер, известно, что именно среднее значение даёт минимальную сумму квадратов расстояний от него до всех значений точек (минимум дисперсии достигается в мат. ожидании).

Алгоритм останавливается после определенного количества итераций либо по достижении сходимости (центры и распределение точек по кластерам не изменилось за итерацию)

K-means

Как правило нужно запускать алгоритм с разным k чтобы найти оптимальное разбиение

Например, вместо нахождения 3 кластеров (1 большой, 2 поменьше, но все явно отделены друг от друга), алгоритм может разбить большой кластер на 2 поменьше, а в третий поместить 2 маленьких.

Для улучшения результатов можно ограниченное число раз перезапустить алгоритм для других начальных центроидов

- Особенности метода:
  - Количество кластеров необходимо определять самостоятельно
  - Теоретически обеспечена сходимость к локальному минимуму
  - На практике локальные минимумы могут не давать логичный результат для кластеризации
  - Результат зависит от выбора начальных центроидов, которых может быть бесконечно много.
  - Старается создавать кластеры примерно одного размера / разброса, выделяет эллиптические(шарообразные) кластеры

1. Задача

- Набор данных –Quake (землетрясения)
- Задача – определить точки сейсмической активности
- Способ решения – кластеризация с выделением центров кластеров
- Методы кластеризации – K-means, EM-алгоритм

http://sci2s.ugr.es/keel/category.php?cat=uns

Набор данных Quakes (землетрясения)

- Объекты – информация о землетрясения
- 2178 объектов
- 4 признака:
  - Глубина точки гипоцентра землетрясения
  - Широта и долгота точки землетрясения
  - Сила землетрясения по шкале Рихтера
- Какую информацию можно найти в этой базе с помощью кластеризации?

```
from sklearnimport cluster
model = cluster.KMeans(n_clusters=n,init='random', algorithm='full',
max_iter=10000)
#Заданиепараметров
clusterobj = model.fit(dataset)
#Нахождениекластеров
print(clusterobj.cluster_centers_)
print(clusterobj.inertia_)
#Распечатка центров полученных кластеров и целевой функции
```

EM-алгоритм

Имя алгоритму было дано в 1977 году в статье авторов Arthur Dempster, Nan Laird, Donald Rubin. До этого метод использовался различными учёными для разных задач без какого-то определения алгоритма. Однако именно эта статья в *Journal of the Royal Statistical Society* дала жизнь EM-методу как инструменту статистического анализа.

- Данные представляются как выборка из смеси распределений (например, нормальных)
- Целевая функция – функция правдоподобия для предложенного набора данных и заданного количества компонент смеси.
- Итерация алгоритма состоит из 2-х этапов
- Количество компонент задаётся заранее

Описание EM-алгоритма

Далее предполагается что распределения нормальные (гауссовы). Под параметрами тогда понимаются вектор средних и матрица ковариаций каждой компоненты смеси, а также априорные вероятности того, из какой компоненты получены данные. В качестве скрытых переменных могут быть взяты переменные z_{ij}, которые представляют из себя апостериорные вероятности получения объекта I из компоненты j. При инициализации, их берут такими, что если объект i получен из компоненты j, то $z_{ij} = 1$, иначе = 0.

Тогда, на шаге E данные апостериорные вероятности могут быть получены через формулу Байеса –f это функции плотности, альфа – априорные вероятности.

На шаге M параметры пересчитываются с помощью весов W довольно очевидным образом.

- Предполагается, что данные X получены из смеси распределений с параметрами ϑ, также предполагается наличие скрытых переменных Z. Параметры перед началом алгоритма инициализируются.
- **Estimation (E-step):** На основе данных и известных параметров вычисляется матрица Z весов или апостериорных вероятностей:

$$w_{ij} = \frac{f\left(x_i \middle| z_{ij}, \theta_j^{s-1}\right) * \alpha_j^{s-1}}{\sum_{r=1}^{k} f(x_i | z_{ir}, \theta_r^{s-1}) * \alpha_r^{s-1}}$$

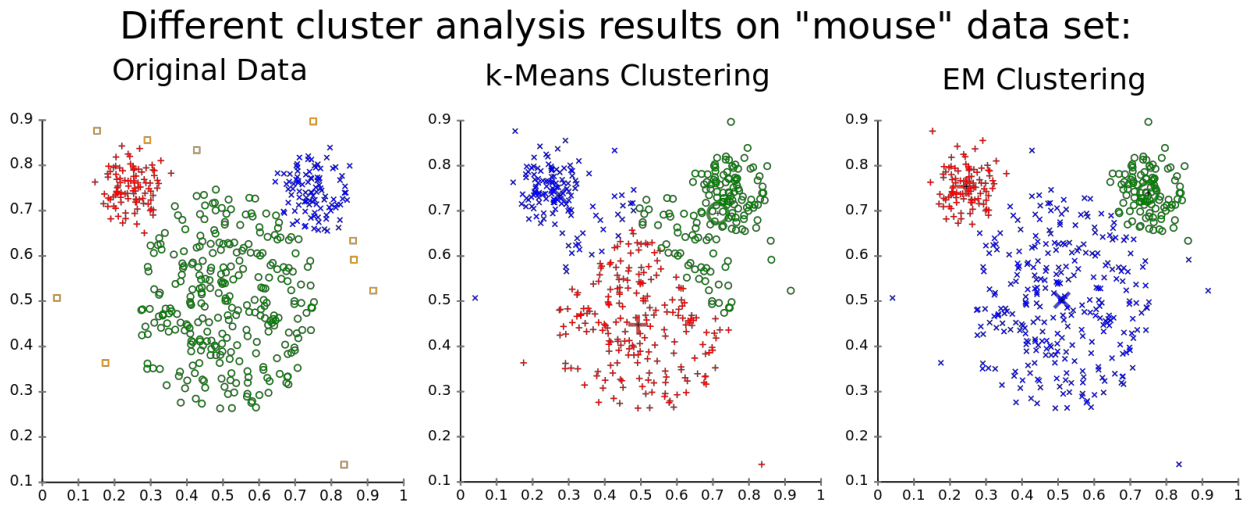
- **Maximization (M-step):** Пересчитываются параметры (средние значения, априорные вероятности, матрицы ковариации)

Особенности EM - алгоритма

Для большей устойчивости алгоритм можно несколько раз перезапускать с разными начальными условиями.

- Количество компонент необходимо определять самостоятельно
- Вектор скрытых переменных вводится таким образом, чтобы:
  - Его было легко найти при известных параметрах
  - Поиск максимума правдоподобия упрощается если известен вектор скрытых переменных
- Теоретически обеспечена сходимость к локальному минимуму
- Локальный минимум сильно зависит от начальной инициализации параметров (неустойчивость по начальным данным)

Картинка внизу хорошо подходит для K-means. Сверху можно увидеть пример не самого хорошего набора данных для K-means кластеризации – видно, что кластеры пытаются быть примерно одного размера



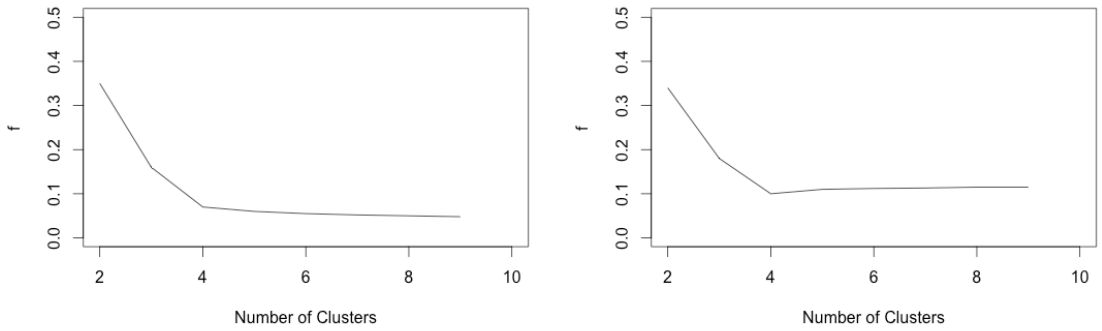
Алгоритмы определения количества кластеров

Вместо суммы внутрикластнерных расстояний можно использовать отношение среднего внутрикластерного расстояния к среднему межкластерному расстоянию.

Индексы как правило обладают некоторым набором значений, при котором можно сделать вывод о высокой степени кластеризации. Наличие соответствующих значений для некоторого выбранного количества кластеров может являться причиной оконгчательного выбора данного числа кластеров.

- Каменистая осыпь: анализируя график суммы внутрикластерных расстояний, эмпирически находится место, где увеличение количества кластеров перестаёт сильно влиять на изменение этой суммы.
- Различныеиндексы– Davies-Bouldin index, Dunn index, Silhouette coefficient.
- Другие эмпирические наблюдения (по количеству объектов в кластерах, по визуальным данным и т.д.).

Слева пример графика функции внутрикластерных расстояний, справа – отношения внутрикластерных расстояний к межкластерным. Можно заметить, что в районе 4 кластеров оба графика «останавливаются». Это может служить эмпирической причиной выбора 4 кластеров в этом конкретном случае




```
from sklearnimport mixture
model = mixture.GaussianMixture(n_components=50, max_iter=10000)
#Задание параметров
model.fit(dataset)
#Нахождение кластеров
print(model.means_)
#Распечатка центров полученных кластеров
```

Практическое задание

1) Построить графики показателей (сумма квадратов расстояний, отношение среднего внутрикластерного расстояния к внекластерному) для различного количества кластеров для набора данных Quake. Определить оптимальное число кластеров, анализируя эмпирическую информацию распределения «очагов».

Как правило нужно запускать алгоритм с разным k чтобы найти оптимальное разбиение

Например, вместо нахождения 3 кластеров (1 большой, 2 поменьше, но все явно отделены друг от друга), алгоритм может разбить большой кластер на 2 поменьше, а в третий поместить 2 маленьких.

Для улучшения результатов можно ограниченное число раз перезапустить алгоритм для других начальных центроидов