**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра вычислительной техники**

отчет

**по дисциплине «Технология анализа и извлечения данных»**

Тема: Алгоритм EM

|  |  |  |
| --- | --- | --- |
| Студентка гр. 2308 |  | Рузанова Л.В. |
| Преподаватель |  | Холод И.И. |

Санкт-Петербург

2017

1. **Общие сведения об алгоритме**

Алгоритм EM был объяснен и получил свое название в классической статье 1977 года Артуром Демпстером, Нэном Лэрддом и Дональдом Рубином. Они указали, что метод был «много раз предложен в особых обстоятельствах» более ранними авторами. Очень подробное рассмотрение метода EM для экспоненциальных семейств было опубликовано Рофом Сандбергом в его диссертации и нескольких работах после его сотрудничества с Пер Мартином-Лёфом и Андерсом Мартином-Лёфом.

Опубликован в статье:

Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society, Series B. 39 (1): 1–38. JSTOR 2984875. MR 0501537

* 1. **Основная идея**

Данные в каждом кластере подчиняются нормальному закону распределению, поэтому для каждого кластера можно высчитать математическое ожидание и дисперсию, при которых функция правдоподобия (вероятность принадлежности объектов кластеру) максимальна.

Мы предполагаем, что любой объект принадлежит ко всем кластерам, но с разной вероятностью. Задача заключаться в «подгонке» совокупности распределений к данным, а затем в определении вероятностей принадлежности объекта к каждому кластеру. Объект должен быть отнесен к тому кластеру, для которого данная вероятность выше.

* 1. **Постановка задачи**

Имеется M многомерных векторов тестовых значений (исходные данные):

Xm=(xm1,xm2,...,xmN), m=1,...M, N – размерность вектора

K – количество кластеров (заранее известно, больше 1). Для каждого кластера задан вектор математических ожиданий μk и вектор дисперсии σ2k, где k=1,...K.

Целью алгоритма является построение вероятностей (gmk), что объект Xm принадлежит кластеру k.

1. **Реализация алгоритма**

Ниже представлена блок схема алгоритма.

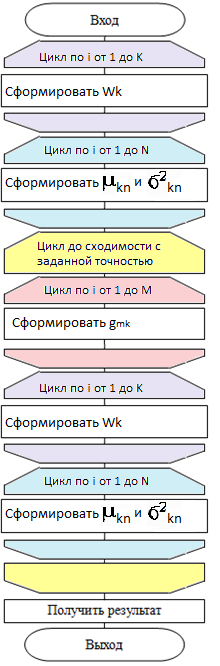


Рисунок 1. Блок-схема алгоритма.

1) Введем переменные wk= 1/K.

2) Инициализируем мат. ожидания для каждого кластера μkn=X1n,  k=1,...,K; n=1,...,N.

3) Инициализируем дисперсии для каждого кластера   
σ2kn=1/(MK)[(X1n-μkn)2+ (X2n-μkn)2+...+(XMn-μkn)2].

4) E-шаг gmk=(wkpk(Xm))/[w1p1(Xm)+w2p2(Xm)+...+ wKpK(Xm)],где pk(X) - плотность нормального распределения, соответствующее k-му кластеру.  pk(Xm)=1/(Σk(2π)N/2)\*exp[-((Xm1-μk1)2+...+(XmN-μkN)2)/(2Σ2k)], Σk=σk1\*σk2\*...\*σkN.

5) М-шаг wk=1/M[g1k+g2k+...+gMk], μkn=1/(Mwk)[g1kX1n+g2kX2n+...+gMkXMn],

σ2kn=1/(Mwk)[g1k(X1n-μkn)2+ g2k(X2n-μkn)2+...+gMk(XMn-μkn)2].

6) Шаги 4 и 5 повторять заданное количество раз.

После того как вероятности gmk построены, мы можем отнести объект Xm к кластеру с номером k', если gmk'=maxk{gmk}.

* 1. **Применение алгоритма**

Ниже представлены исходные данные (танки):

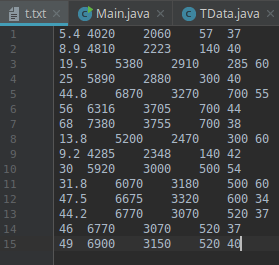
****

Рисунок 2. Файл с исходными данными.

Результат работы по распределению на 3 кластера (TemClust) представлен на рисунке 3.

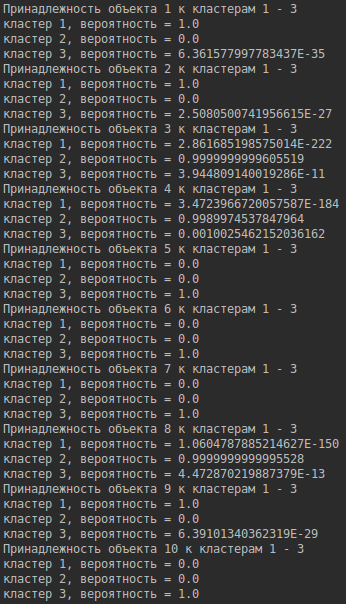
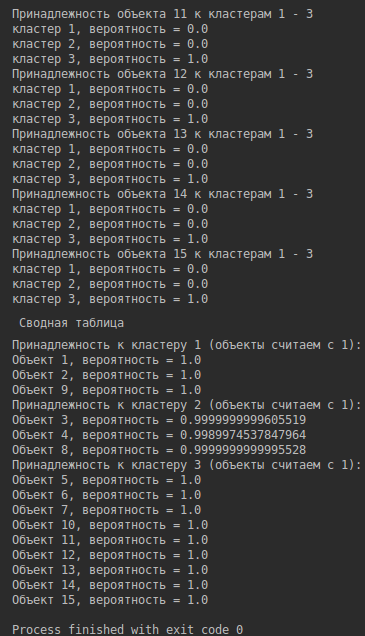
**** ****

Рисунок 3. Результат кластеризации танков.

Как видно на представленных рисунках результат работы алгоритма выявил принадлежность для каждого объекта к каждому кластеру. Соответственно с полученной информацией объект относится к тому кластеру, вероятность принадлежности к которому максимальна.

* 1. **Оценка алгоритма**

Сложность инициализации - O(k) для задания весов кластерам, O(N) для подсчета начальных значений мат. ожиданий и дисперсий.

Сложность E и M шагов - O(L\*d), где

* L отвечает за сходимость алгоритма, которая зависит от конкретного набора данных и изначальной инициализации. Она достигается (либо нет) с заранее заданной точностью.
* d = max(N, M), данный параметр влияет на внутренний цикл алгоритма, а именно прогон по параметрам объекта, либо, что вероятнее, по самим объектам.
* N – количество параметров объекта.
* M – количество объектов.

**Вывод:** в общем случае на время работы EM алгоритма влияет количество объектов и сходимость алгоритма.

Оценка потребления памяти

* O(M\*N) – выборка данных, N объектов с M полями.
* O(K\*N) – высчитывание промежуточных данных для каждого кластера из K по каждому объекту N.
  1. **Достоинства и недостатки**

**Достоинства:**

* Легок в реализации.
* Самообучающийся.
* Быстро сходится при удачной инициализации.
* *Возможность построения желаемого числа кластеров.*

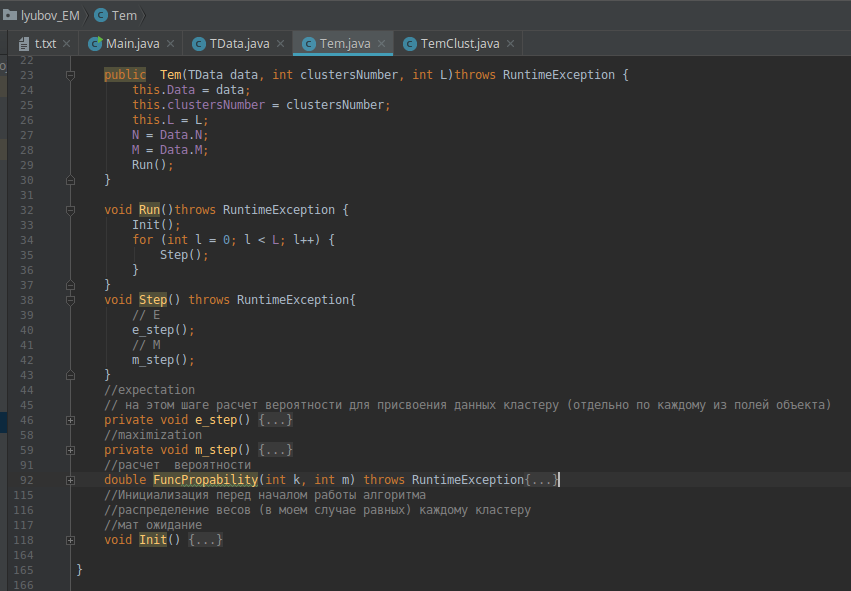
**Недостатки:**

* Предположение о нормальности всех измерений данных не всегда выполняется.
* При неудачной инициализации сходимость алгоритма может оказаться медленной или может быть получен некачественный результат при ее отсутсвии.
* Застревает в локальных оптимумах.
* *Подразумевает наличие априорных знаний о данных (изначальное задание количества разбиений).*

1. **Декомпозиция алгоритма**

Алгоритм легко декомпозируется в силу его природы:

1. Инициализация значений.
2. Главный цикл работы алгоритма состоящего из двух шагов E и M, плюс проверка сходимости.
3. Шаг E.
4. Шаг M.
5. Вне расчета алгоритма также в отдельную обработку выделен парсинг файла и наполнение данными структуры для дальнейшей работы.
6. Вывод результатов.
   1. **Блок работы EM алгоритма**



1) Tem - конструктор класса, передача основных параметров алгоритма (данных, количества кластеров, размерность данных и количество шагов). Вызов Run().

2) Run – вызов Init для инициализации значений (пункт А), запуск Step() главного цикла алгоритма (пункт B).

3) Step – вызов E-шага e\_step(), вызов M-шага m\_step().

4) E-шаг (пункт C).

5) M-шаг (пункт D).

6) FuncPropobility – расчет вероятности, довольно громоздкая функция, выделена в отдельный блок.

* 1. **Итоговая вычислительная сложность**

|  |  |
| --- | --- |
| Затраты по памяти | O(M\*N) |
| Сложность | O(L\*d)~ O(L\*K\*d) на практике |

* L отвечает за сходимость алгоритма, которая зависит от конкретного набора данных и изначальной инициализации. Она достигается (либо нет) с заранее заданной точностью.
* d = max(N, M), данный параметр влияет на внутренний цикл алгоритма, а именно прогон по параметрам объекта или по самим объектам.
* N – количество параметров объекта.
* M – количество объектов.
  1. **Особенности выявленные при реализации алгоритма**

Расширились ограничение на данные. Разработанный алгоритм позволяет определить вырожденность кластеризации по значениям полей объектов, после предлагает пользователю исключить данные поля из рассмотрения кластеризации.

1. **Выводы**

В ходе работы был изучен алгоритм кластеризации EM для числового типа данных. Была построена оптимальная модель алгоритма, оптимизирован сам алгоритм и реализован с помощью стандартных библиотек. Было проведено сравнение теоретических и практических результатов оценки работы алгоритма.