**数据分析与处理部分实践作业 3#**

# KMeans聚类实现

## 聚类算法介绍

### 1.1.1聚类任务

在“无监督学习”中，训练样本的标记信息是未知的，目标是通过对无标记训练样本的学习来揭示数据的内在性质及规律，为进一步数据分析提供基础。聚类的目的是试图将数据集中的样本划分为若干个通常不相交的子集，每个子集称为一个“簇”。

聚类既能是一个单独过程，用于寻找数据内在的分布结构，也可作为分类等其他学习任务的前驱过程。

### 1.1.2性能度量

聚类性能指标大致两类，一是将聚类结果与某个“参考模型”进行比较，称为“外部指标”，二是直接考察聚类结果而不利用任何参考模型，称为“内部指标”。

聚类性能的度量的外部指标为：

Jaccard系数：

FMI指数：

Rand指数：

性能度量结果值均在[0,1]区间，值越大越好。

聚类性能的度量的内部指标为：

DB指数：

Dunn指数：

DBI的值越小越好，而DI的值越大越好。

### 1.1.3 距离计算

给定样本与，最常用的是闵可夫斯基距离：

其中当p=1时为曼哈顿距离，p=2时为欧式距离。

此外，对于无序属性，使用VDM距离：

其中，表示属性u上取值为a的样本数，表示在第i个样本簇中在属性u上取值为a的样本数，k为样本簇数。

明显的，闵可夫斯基距离和VDM结合可处理混合属性，当样本空间中不同属性重要性不同时，可使用加权距离，赋予权重。

## k均值算法(KMeans聚类)

### 1.2.1 算法原理

K-means 聚类算法是要确定类的数目k 为聚类数据集的一个先决条件，通常簇的数目被事先确定，K-means聚类算法是算法的最广泛使用的算法之一，对于大型数据处理效率比较高，特别是当样本分布呈现更明显的团聚现象时效果会更好[10,11]。K-means 聚类算法的基本思想是给定的随机初始k个簇中心，按照训练数据的最近点的原则进行分类分配给每个群集。然后根据平均的方法来计算每个群集的质心，从而确定新的集群簇心。一直迭代，直到簇中心的移动距离小于某个给定的值后结束算法。

算法核心思想：首先选定簇数，随机选择样本作为初始均值向量，依次考察每个样本与当前均值向量均值的距离，选定距离最近的簇，归于该簇。

所有样本考察结束一轮以后，分别更新，每个簇的新的均值向量，不断重复上述过程，当均值向量未更新后得到最终的簇划分。为了防止运行时间过长，设置一个最大运行轮数或者最小调整幅度阈值。

### 1.2.2 算法实现步骤

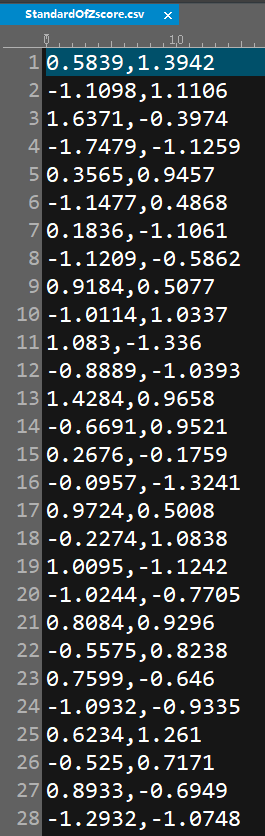
初始随机给定K个簇中心，按照最邻近原则把待分类样本点分到各个簇。然后按平均法重新计算各个簇的质心，从而确定新的簇心。一直迭代，直到簇中心移动距离小于某个给定的值。具体来说，通过输入聚类个数k，以及包含 n个数据对象的数据库，输出满足方差最小标准的k个聚类，算法描述如下：

|  |
| --- |
| **Algorithm 1** K-means聚类算法 |
| **输入:** 样本集;  聚类簇数k  **过程**：   1. 从D中随机选择k个样本作为初始均值向量 2. **repeat** 3. 令 4. **for** j=1,2,…,m **do** 5. 计算样本与各均值向量的距离：; 6. 根据距离最近的均值向量确定的簇标记：; 7. 将样本划入相应的簇：; 8. **end for** 9. **for** i=1,2,…,k **do** 10. 计算新均值向量：; 11. **if**  then 12. 将当前均值向量更新为 13. **else** 14. 保证当前均值向量不变 15. **end if** 16. **end for** 17. **Until** 当前均值向量均未更新   **输出：**簇划分 | |

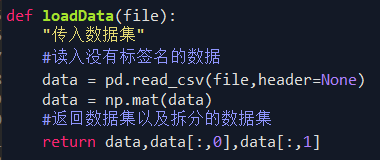
## k均值算法实现过程

### 1.3.1 导入数据

1、数据集的选择格式，80行2列的csv格式的数据：



2、导入数据集：



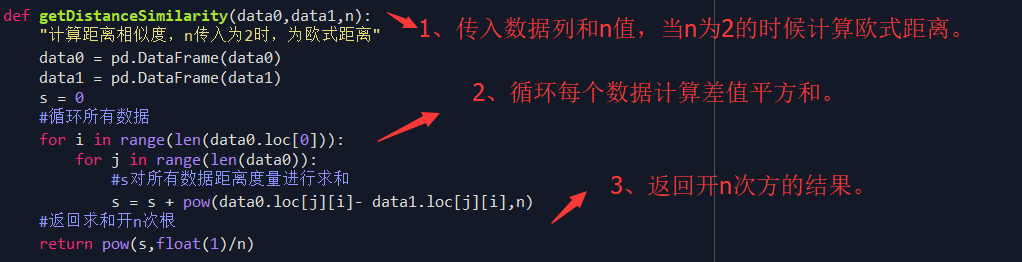
通过pandas中的read\_csv读入数据，转为矩阵形式，并将两列拆开，目的是为了测试后面的欧式距离函数的计算。

3、数据读入结果：



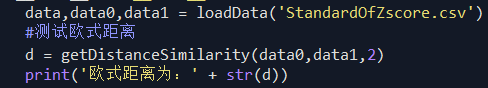
### 1.3.2 计算距离相似度

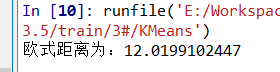
1、函数构造：



2、根据距离度量公式来构造函数，传入n值为2时，计算数据的欧式距离。

3、调用函数计算欧式距离：





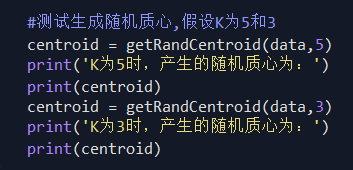
### 1.3.3 生成初始的随机簇中心

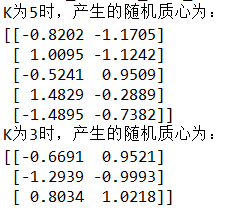
1、构造函数：



2、过程：首先在数据集dataframe最后添加一列，作为产生随机数列，并且保证随机生成必须不同；然后按rand列进行升序排序，选择排列在前k个数据作为初始化随机簇中心，保证每次选择的随机性。

3、调用函数，测试结果：





4、缺点，由于函数是自己写的随机选点，因为要产生不同随机数，时间效率上耗费很大，可以考虑调用random.sample函数来产生不同的随机数，效率高很多。

### 1.3.4 KMeans聚类函数

1、KMeans聚类函数构造：



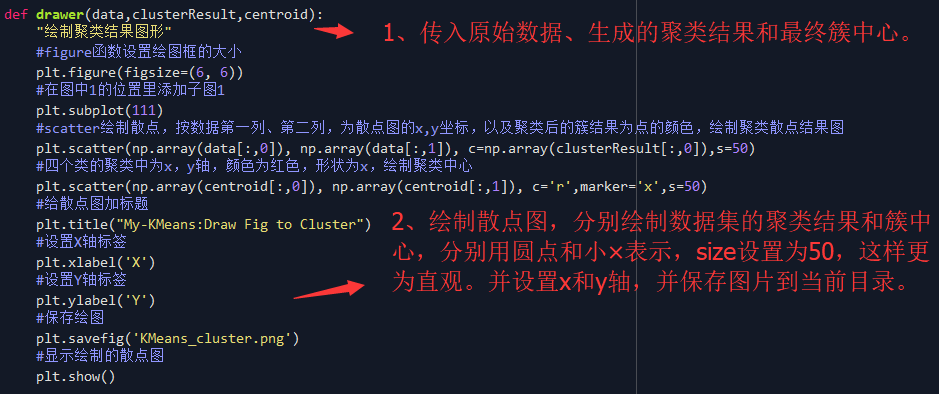
2、要点：

（1）要对簇中心进行拷贝，否则比较上一轮簇中心和当前簇中心是否一样就没有意义了，因为会一直改变；

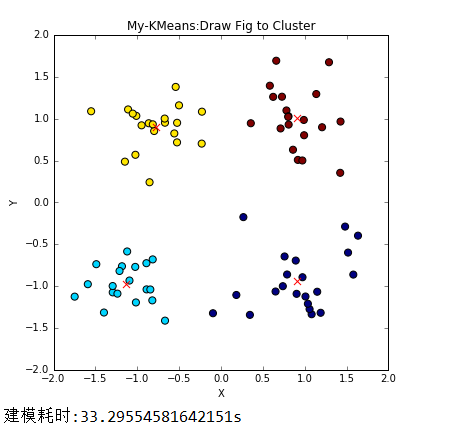
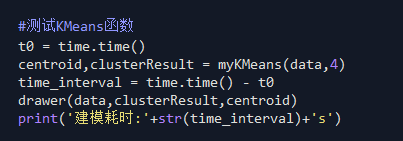
（2）更新簇中心的时候，要在dataframe类型的数据集后面添加一列来标记当前是属于哪一类，取出那一类，求均值作为当前那一类的新的簇中心。当是在求均值的时候，又要将最后一列删除，这里耗费时间性能特别明显，特别是要操作很多次，导致时间性能特别差。

（3）判断簇中心是否变化时，应该比较矩阵中的每一个值都已经没有变化，我写进了循环里面，这里也有些耗时间。

3、绘图函数的构造：



4、调用KMeans聚类函数与绘图函数：



由于种种原因导致时间性能并不好，这里调用了Python的time库，打印了具体建模的时间，可以看见时间性能很差，聚类效果相对不错。

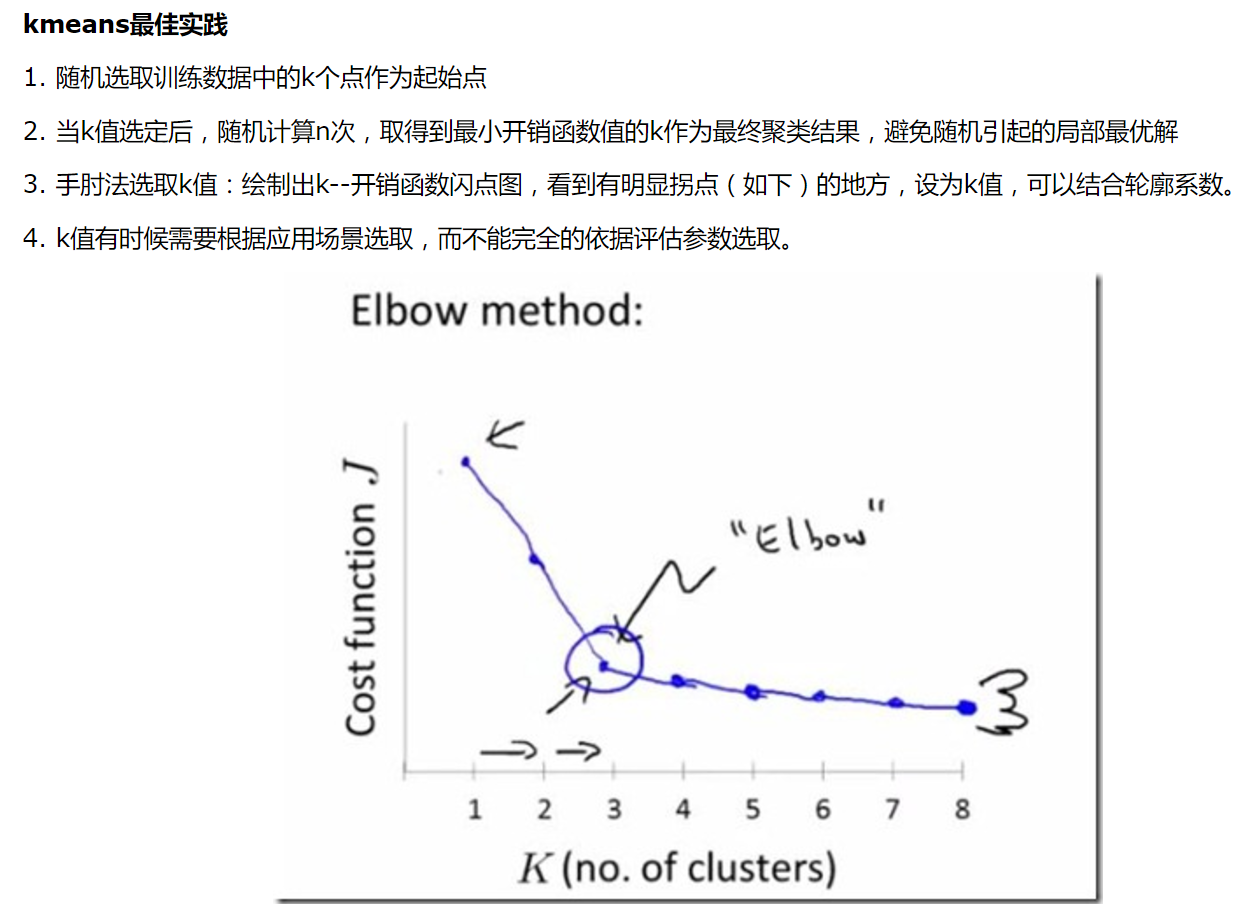
### 1.3.5 用肘方法确定K值

#### 1.3.5.1肘方法介绍

一般来说聚类不会超过8类，肘方法就是绘制K值关于误差平方和的折线图，当K越大时，误差平方和会越来越小，但也不是取K越大越好。

这里给一个博客链接，讲解肘方法确定K值：<http://www.cnblogs.com/bourneli/p/3645049.html>

**当K为3的时候，是最明显的拐点的地方，因此K值取3。**

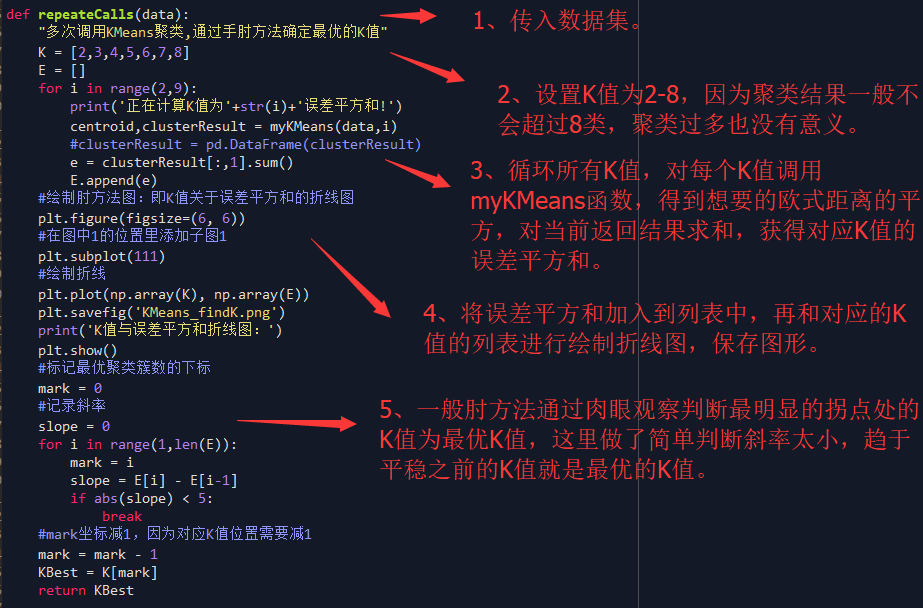


#### 1.3.5.2误差平方和计算

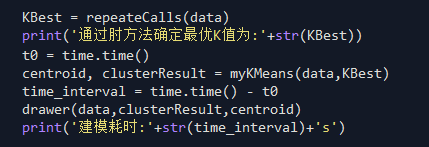
1、获取对应K值聚类结果的欧式距离：

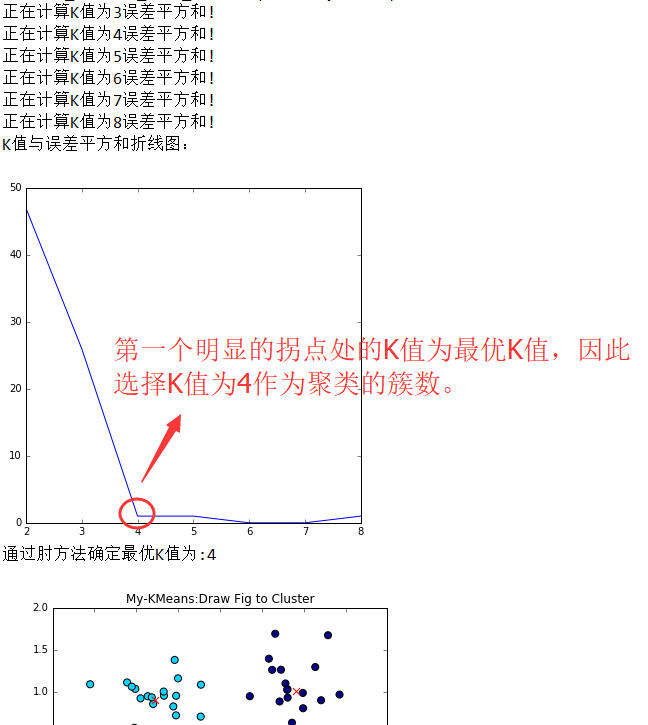


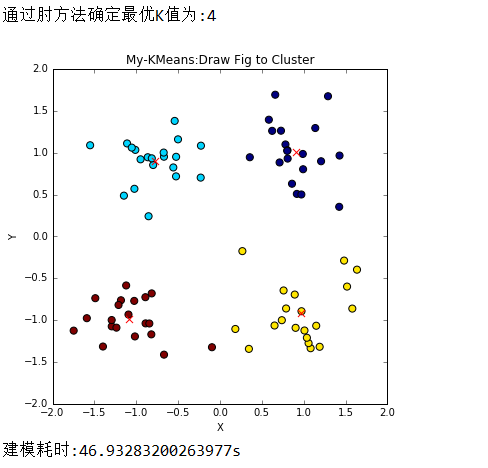
2、对应K值获取误差平方和并绘图：



3、调用函数，得到结果：







4、不足就是时间性能差，一般在20s到50s左右，如果使用numpy库中的计算欧式距离以及产生随机数的sample函数，时间性能会好很多。并且在操作dataframe类型的时候，更新簇中心的时候，大量循环进行了del操作，导致时间性能很差，自己实现的时候没有考虑时间性能。

# EM算法的实现

EM, ExpectationMaximization Algorithm, 期望最大化算法。一种迭代算法，用于含有隐变量(hidden variable)的概率参数模型的最大似然估计或极大后验概率估计，其概率模型依赖于无法观测的隐变量。经常用在ML与计算机视觉的数据聚类领域。

**EM原理：**

最大期望个算法经过两个步骤交替进行计算:

step1: 计算期望E，利用对隐藏变量的现有估计值，计算其最大似然估计值。

step2: 最大化M，最大化在E步上求得的最大似然值来计算参数值。

（M步上找到的参数估计值被用于下一个E步计算中，这个过程不断交替进行）

**EM流程：**

1. 初始化分布参数

2. 重复直到收敛：

E步骤：估计位置参数的期望值，给出当前的参数估计

M步骤：重新估计分布参数，以使得数据似然性最大，给出位置变量的期望估计。

EM是一种解决存在隐含变量优化问题的有效方法，既然不能直接最大化L(o)，可以不断建立l的下界（E步），然后优化下界（M步）。

**EM应用：**

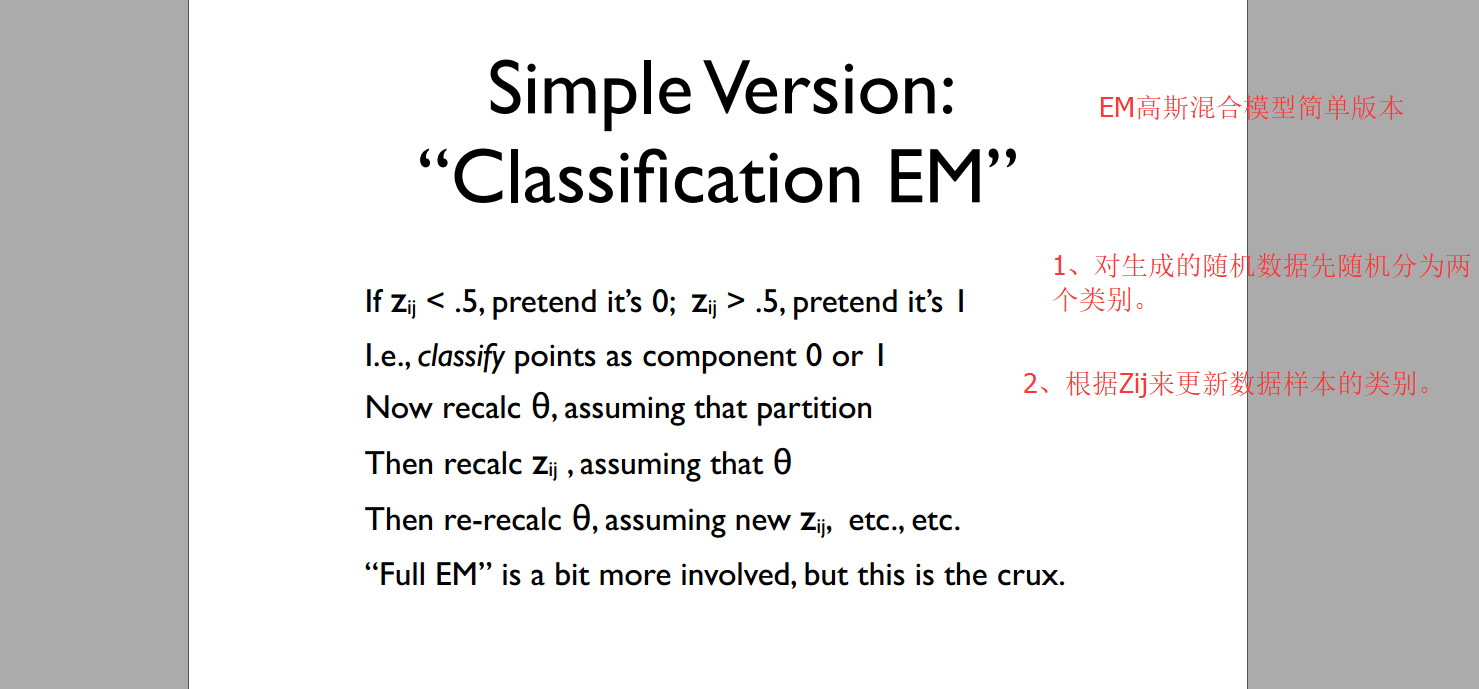
GMM混合高斯模型、聚类、HMM隐马尔科夫模型等。我对**高斯混合模型和聚类**做了实现。

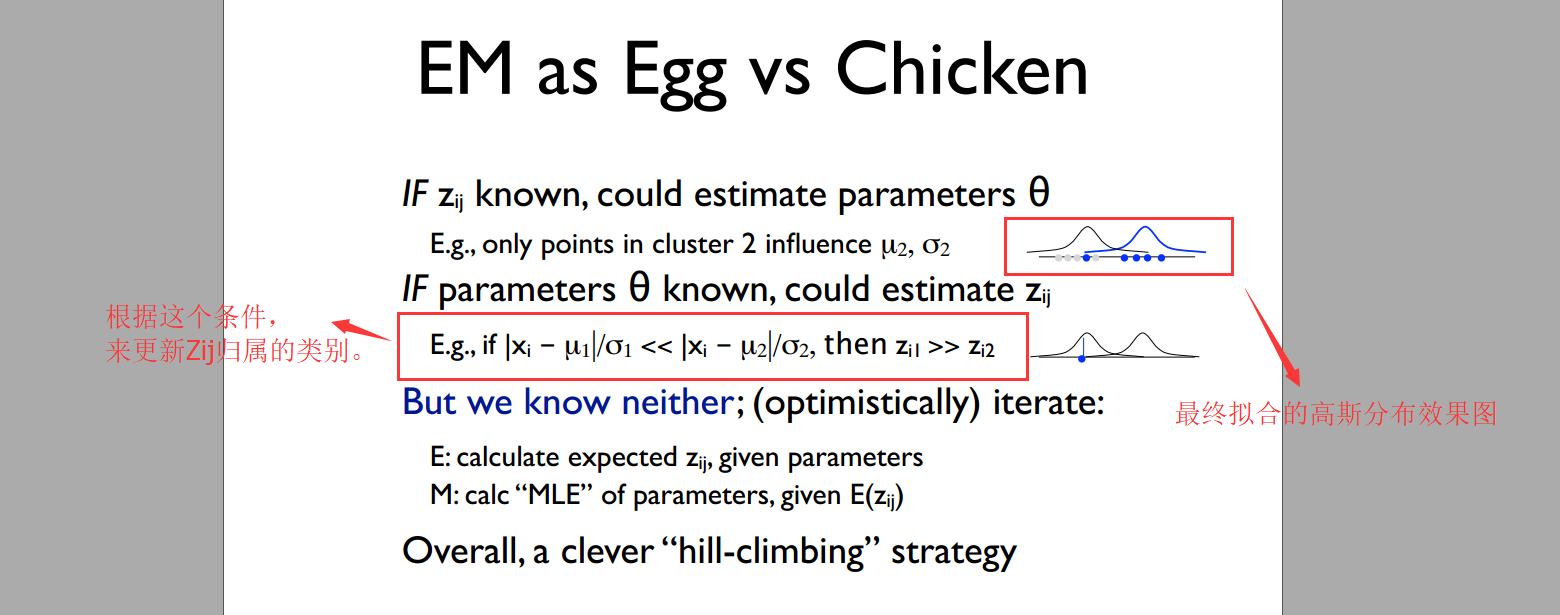
## EM高斯混合模型算法

### EM高斯混合模型算法介绍

1、简单概括来说，就是随机生成两个高斯分布的数据，合并成我的原始数据，然后通过EM模型，对我这堆数据进行归类，尽可能拟合成原始的高斯分布曲线。

2、由于EM数学理论太难，这里按老师要求，实现了EM高斯混合模型的简单版本：





3、EM高斯混合模型实现流程：

**E阶段：按 |x1-u1|/σ1 < |x2-u2|/σ2 不断更新Zij的类别。**

**M阶段：由归类结果计算u和σ新的值，并设定阈值终止迭代。**

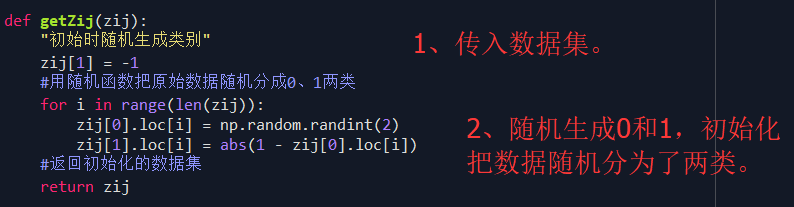
### EM高斯混合模型算法实现

1. 生成数据集：



根据官网生成正态分布的随机函数，传入我需要生成的两组的数据的均值和方差，生成数据，连接两组数据待使用。

2、初始化数据的类别：



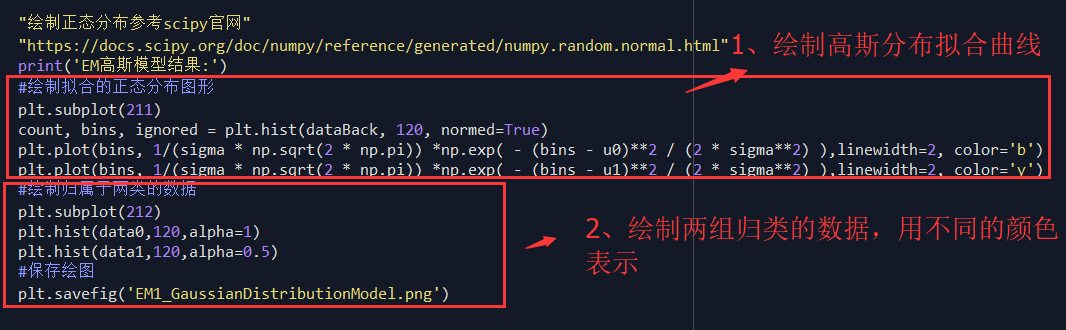
这里就是在构建Zij矩阵，第一列代表随机生成为0的类别，第二列则是为1的类别，相当于初始化数据集为两个类别，准备开始对两类数据进行EM算法的迭代。

3、EM函数：



传入初始化的随机归类数据，分别计算均值和方差，按归类条件不断迭代归类，直到达到指定阈值或者最大迭代次数时结束循环，此时的均值和归类结果就是EM得到结果，返回结果。

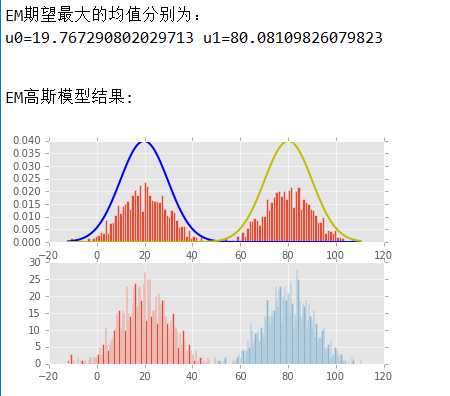
4、结果的可视化：



对于的返回的数据归类结果，分别绘制不同颜色的直方图进行表示，而返回的最大期望的方差用于拟合高斯分布曲线。

5、调用函数，生成结果：





如结果所示，返回最大期望的均值，对于图形可视化展示，上方是拟合的高斯分布曲线，下方是对于返回的两部分归类的数据按不同颜色的绘制结果。

6、缺点与不足，由于|x1-u1|/σ1 < |x2-u2|/σ2作为归类条件，模型不精确，会导致偶尔会出现在迭代过程中所有数据样本归于一类的情况。改进：使用EM完整版本。

## EM概率（隶属度）混合模型算法

### EM概率（隶属度）混合模型算法介绍

EM算法最突出的一点是利用了模糊数学的思想，计算刚开始，我们随机给出两类数据概率分布参数，任意一个数据x，假设在两个分布中的分布密度分别为 p(x) 和 q(x)，那么我们把这个数据对集合F和M的隶属度按照概率密度的比例分配为：

从模糊集合的观点看，K均值算法类集对应经典集合，EM算法对应模糊集合。实际上不难证明，K均值算法是EM算法退化到经典集合论的特例，EM算法是K均值算法向模糊集合的推广。

随机向两类数据概率分布参数赋以初值后，K均值算法可以得到一个近似的分类，同样，EM算法也可以得到两类数据初始的模糊集。虽然这两个集合距正确结果还有一定的差距，但是，和K均值聚类算法一样，随着迭代的逐步进行，两个集合对应的隶属函数，或者概率分布函数，距真正的模型越来越逼近。

理想情况下可以证明，EM算法最终能得到零偏差的逼近结果。

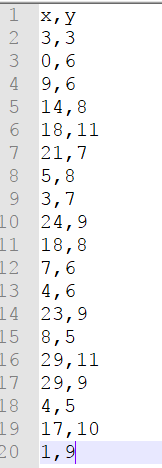
**EM算法的模糊聚类**：来源于数据挖掘概念与技术504页，每次迭代也是由两部分组成：

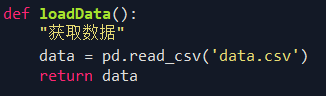
**E阶段**：根据当前的模糊聚类或概率簇的参数，把对象指定到簇中

**M阶段**：发现新的聚类或者参数，最大化模糊聚类的SSE或基于概率模型的聚类的期望似然。

### EM概率（隶属度）混合模型算法实现

1、传入数据集：





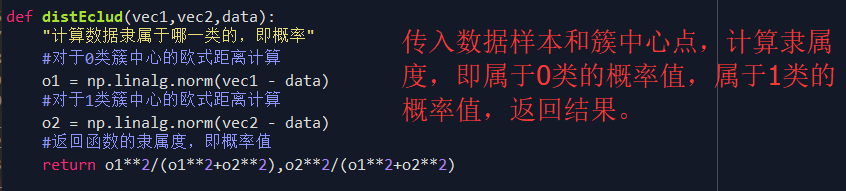
利用pandas传入一组数据点，准备进行EM聚类应用。

1. 初始选择聚类簇中心点：



这里生成初始化随机簇中心和KMeans算法的一致，先对数据加一列rand，对rand列随机生成0-length的随机数，随机数保证生成不同，然后按rand列升序排序，每次选择前K个作为初始化簇中心。

3、计算隶属度，即属于某个类别的概率值：



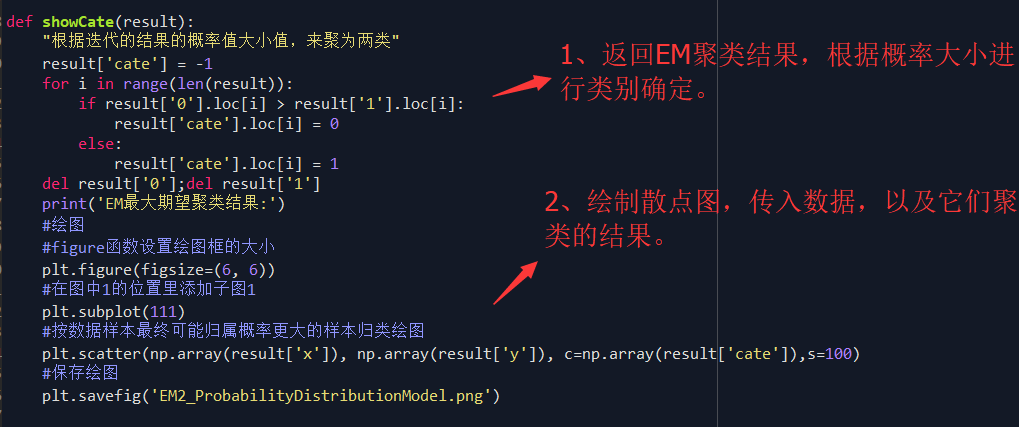
计算公式就是分别算这个样本与簇中心的欧式距离，隶属度就是其中一个欧式距离的平方除以两个欧式距离平方和，也就是概率值，二者加起来是为1的，也就是属于0的概率是0.3，那么属于1的概率就是0.7。返回计算的隶属度的值。

1. EM函数：

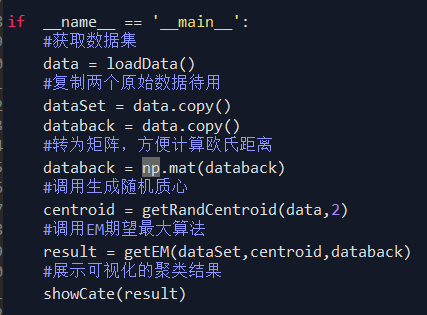


对于传入的数据样本，分别计算属于0类的隶属度，1类的隶属度，然后根据公式更新每个样本归属的簇，再更新簇中心，直到簇中心改变极小时，停止迭代。

1. 得到EM聚类结果，并进行可视化：

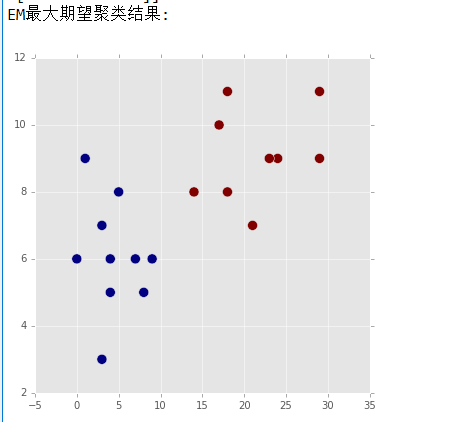


5、调用函数，得到结果：





以及可视化结果：



# 心得体会

本次学习了无监督学习，主要涉及和实现的算法是KMeans聚类和EM最大期望算法，两种算法都是不断迭代得到最终的结果。实现过程中也有类似的地方，并且EM也大量应用于聚类算法中，K均值算法类集对应经典集合，EM算法对应模糊集合。实际上不难证明，K均值算法是EM算法退化到经典集合论的特例，EM算法是K均值算法向模糊集合的推广。

中途出现了各种bug和小问题，解决得不算太好，KMeans的不足是时间效率太差，EM高斯混合模型的缺点是由于按照老师的简单EM版本实现的，偶尔会出现所有样本归于一类，导致无法计算均值方差，无法继续迭代的情况。