EM 算法在聚类中的研究与应用

Research and application of EM algorithm in clustering

学　　院：理学院

专　　业：信息与计算科学

班　　级：信息 171

学　　号：2017210218021

学生姓名：于晗丹

指导教师：李安水

二 0 二一　年　五　月

摘　要

在图像识别、模式匹配、用户画像等领域中，聚类是一种常用的方法，它能有效地提取和分析数据信息。从机器学习角度，聚类分析实质上是搜索簇的一种无监督学习过程，即对模型参数的估计，而聚类过程可以看作是一个参数估计迭代优化过程。最大期望算法——EM 算法，就是一种典型的基于极大似然估计的参数估计方法。但其本身存在一些缺点，如对参数初始值敏感，稳定性较差等。基于此，本文结合理论推导、模拟实验和实证来对 EM 算法做出改进。

首先，本文实现基于高斯混合模型的 EM 算法，并针对一维和二维不同分布数据集，将 EM 算法和 K-Means 算法进行比较。通过实验得到，对于一维数据，EM 算法无法体现其优势，两者都能对模型参数有很好的估计。而 EM 算法在二维数据上有优于 K-Means 算法的表现，但 EM 算法的时间复杂度大于 K-Means 算法。

接着，针对传统 EM 算法受初始值的影响导致其稳定性较差问题，本文提出了半监督学习方法，利用少量已标定样本数据集的信息作为先验信息指导聚类过程，进而提出了 SSL-EM 算法。本文通过实证和论证的方法，证明了该算法相比于 EM 算法能够在一定程度上减小聚类结果受初始值的影响程度。其中，SSL-EM 算法在高斯分布模拟数值上的聚类 FMI 高达 95.33%，且在 UCI 数据集 seeds、wine、customers 上的聚类 FMI 分别达到 92.8%、91.2%、82.6%，均优于 EM 算法。

其次，为了进一步研究 EM 算法的稳定性，本文引入最大熵原理，将其运用到 SSL-EM 算法中，有效解决了隐变量的后验概率估计严重依赖于已标定样本的先验信息的问题，于是提出 SSL-MEEM 算法。本文先对该算法中两个超参数取值进行研究，通过网格搜索方法得到最优超参数组合 (γ_2，α) 为 (0.1，2.5)，并将结果运用于后续实验。然后通过模拟存在已标记样本不能大致反映对应类别数据的真实分布情况的数据集进行实验，结合理论推导验证了 SSL-MEEM 算法能够较好的避免受异常初始值的影响而得到较优的聚类效果，从而证明了 SSL-MEEM 算法相较于 SSL-EM 算法具有更好的稳定性和鲁棒性。且在 UCI 数据集 seeds 上测试得到的聚类 FMI 达到 95.8%。

最后，本文将 K-Means、EM、SSL-EM、SSL-MEEM 算法分别运用于经过 autoEncoder 的 FashionMNIST 图像数据集进行聚类，通过聚类外部评价指标对各算法的聚类效果进行评判，得到本文提出的 EM 的改进算法 SSL-EM、SSL-MEEM 算法均能较准确的估计聚类模型参数，最终对图像的识别率均达到 70% 以上。

关键词：聚类；K-Means 算法；EM 算法；高斯混合模型；SSL-EM 算法；SSL-MEEM 算法

Abstract

Clustering is a common method to extract and analyze information effectively, which is widely applied in numerous fields, such as image-recognition, pattern-matching and user-portrait. The clustering process can be regarded as a parameter estimation iterative optimization process. Expectation-Maximization(EM) algorithm is a classical parameter estimation method based on maximum likelihood estimation. However, EM algorithm has some shortcomings. For example, it is sensitive to the initial value of parameters and has relatively poor stability. Therefore, in this thesis, the EM algorithm is improved from both theoretical and experimental studies.

Firstly, we implement the EM algorithm and compare it with the K-Means algorithm on 1- dimension and 2-dimension datasets with different distributions. The experimental results show that for 1-dimension datasets, EM algorithm cannot reflect its advantages and both of them can estimate model parameters well. While, for 2-dimension datasets, EM algorithm can get better results than K-Means algorithm, though its time complexity is greater than the K-Means algorithm.

Secondly, in order to resolve the poor stability of EM algorithm, we introduce the Semi-Supervised Learning EM(SSL-EM) algorithm, which utilizes a small number of sample data to guide the clustering process. Then through experiments and theoretical demonstration, we prove SSL-EM algorithm can solve the problem that the posteriori probability estimation is overly dependent on the estimated parameters in a certain degree. Where, the FM Index of clustering on Gaussian distribution datasets is up to 95.33%. Meanwhile, the FM Index of clustering on UCI datasets(seeds, wine, customers) achieve at 92.8%、91.2%、82.6% respectively, and the results are all better than EM algorithm.

Thirdly, we also introduce SSL-MEEM algorithm which is based on the principle of maximum entropy. At first, we study the optimal solution of hyper-parameters by using Grid Search and regard (0.1,2.5) as the optimal value. Then we test this algorithm on two simulation datasets including unusual sample data which cannot reflect the real distribution of the category. The results illustrate that compared with SSL-EM algorithm, SSL-MEEM algorithm has a more steady and better performance on unusual situation. We also test this algorithm on UCI datasets and the FM Index of clustering on dataset seeds is up to 95.8%.

Finally, in this thesis, K-Means、EM、SSL-EM、SSL-MEEM algorithms are applied on the autoEncoder processed FashionMNIST image dataset separately. The results show that SSL-EM and SSL-MEEM algorithm all perform very well, since the recognition rate of the images all achieve above 70%.

Keywords：Clustering; K-Means algorithm; EM algorithm; Gaussian mixture model; SSL-EM algorithm; SSL-MEEM algorithm

目　录

1 绪论

1.1 研究背景及意义

随着信息化时代的快速发展，海量数据围绕着人们的生活，如何有效利用这些丰富的数据，已经成为广大学者的重点关注的焦点之一。为有效解决数据丰富、知识贫乏这一问题，自二十世纪 80 年代开始，数据挖掘技术逐步发展起来。现在数据挖掘技术应用领域广泛，包括生物信息学、金融数据分析、电子商务等[1]。

聚类分析是数据挖掘的一项重要任务，也是人类成长过程中的一个重要行为。聚类分析是研究多个因素事物分类问题的定量方法，其基本原理是依据事物之间的相似度定量地确定事物之间的事物亲疏性，最终将相似度高的对象聚集在同一类别中，把相似度低的分开。从而达到了两个最大化目标即，类之间对象同质性最大化和类与类之间异质性最大化。在研究和处理事物的时候，经常会需要将事物进行归类，例如在商业市场中为了寻找潜在市场，需要对客户群体的行为特征进行分析归纳，也可根据不同客户群体的消费特点制定出不同的营销组合; 在金融领域，可通过对股票进行分类，来选择分类投资风险; 在地理学方面，很多时候由于观测数据量庞大，亦需要对它们进行分类归并，获得其典型特征再进行深入的分析。

从机器学习角度，聚类分析实质上是搜索簇的一种无监督学习过程，不依赖于预先定义拥有标签的的样本。同时作为数据挖掘的一种重要手段，聚类分析可独立作为一个工具来获取一批数据的分布特征，并对某一簇有分析价值的部分进行更深一步的分析研究。因此聚类分析还可以作为其他算法的预处理步骤。至今，人们根据不同数据类型以及聚类目的提出了多种聚类算法，包括 K-Means、DBSCAN、AGNES、STING 算法等。综上，可以得出，聚类分析在当下已经成为一个热点研究话题。

1.2 研究现状

在国外，对聚类的研究其实最早可以追溯到半个世纪以前，Lloyd 在 1957 年基于对 Voronoi 图的划分思想首次提出了 K-Means 聚类算法的雏形，之后在 1967 年 James MacQueen 又对 K-Means 聚类算法做了进一步的研究，并在他的论文《用于多变量观测分类和分析的一些方法》中首次提出了 “K-Means” 这一术语[2]。但是由于 K-Means 聚类算法具有一定的局限性，该算法通常只能做到局部收敛而无法达到全局收敛，且只适用于数值型数据的聚类，为改进该算法，研究进展在这一时期进展缓慢，直到 1990 年由 Kaufman L. 和 Rousseeuw 分别提出了 K 中心点算法 PAM 和 CLARA 以及分裂层次聚类算法 DIANA, 在一定程度上减少了噪声带来的影响[3]。在之后几年中人们的注意力集中到研究层次聚类，1996 年 Zhang T. 等人使用 CF 树进行层次聚类，提出了 BIRCH 聚类算法[4]，使在大型数据集中取得较高的速度和可伸缩性。而同年，Ester M. 等人另辟蹊径，基于密度的思想提出了新的聚类算法——DBSCAN（Density-Based Spatial Clustering of Applications with Noise)[5]，能够在具有噪声的数据中发现任意形状的簇。

随着信息量的不断扩大，产生了巨量的数据，而且数据的类型多种多样。有的数据样本中包含 “噪声” 数据或者不明数据，加之模型参数初始设定需具备大量先验知识，给聚类的实施带来了困难。而 EM 算法，即期望最大化算法，是一种当观测数据不完整时求解最大似然估计的迭代算法。因此近几年许多学者对 EM 算法的进行分析研究，并将其运用到聚类中。

通过查阅资料发现目前学者大多在对 EM 算法收敛性以及对初始值敏感性问题进行研究和改进。在参数初始化方面，Kwedlo, Wojciech（2015）提出了一种新的基于高斯混合模

型聚类的多重重启 EM 算法初始化方法。并将该方法与其他三种随机电磁初始化方法进行了比较。对合成数据的结果表明，对于分离良好的簇，其最大成对重叠不太高，所述方法产生的簇与原始数据分区相对应，如调整的 Rand 指数。在实际数据上的实验表明，对于两个较小的数据集，该方法的性能与其他三种方法相当，对于两个较大的数据集，该方法的性能明显优于其他三种方法[8]。岳佳（2007）在其论文中引入用于密度估计的 “binning” 法来初始化 EM，且实验结果表明，应用 binning 法来初始化 EM 的高斯混合模型聚类优于其它传统的初始化方法[9]。

在 EM 算法收敛性方面，其中夏筱筠、张笑东等在《一种半监督机器学习的 EM 算法改进算法》中提出了在最大似然函数中加入惩罚最小二乘因子，同时引入非负约束作为先验信息，有效解决 EM 算法容易局部最优化问题[10]。Estelle Kuhn 和 Catherine Matias(2020) 等人提出了一种基于小批量蒙特卡洛马尔可夫链随机逼近的 EM 算法，能够加快算法收敛速度[11]。

1.3 本文主要研究内容

总结目前聚类分析领域研究内容，发现在实际应用中，高斯混合模型聚类是一种较为合适的聚类方法，它的目的是试图找到多维高斯模型概率分布的混合表示，从而拟合出任意形状的数据分布，因此适用范围广，且性能较好。但该模型中存在隐变量（不可观测的随机变量）。对于处理隐变量参数估计问题，采用 EM 算法，即添加 “潜在变量”，有效地解决含有隐藏变量的问题。然而 EM 算法本身存在一定的缺陷，如：对参数初始值敏感，需要给予隐变量的所有可能的取值等。因此本文在提出半监督聚类学习的基础上引入基于最大熵的 EM 算法改进算法，来提升估计参数的性能。并通过随机模拟数值、人工数据集等实验数据对算法性能进行比较证明。

1.4 本文内容组织框架

本论文分为六章，具体内容组织框架如下：

第一章绪论，主要介绍了聚类以及 EM 算法的研究背景意义及现状，并给出了常见的 EM 算法改进方法。

第二章聚类的理论知识准备，先介绍了类的定义及相似度度量方法，并主要介绍了聚类算法的性能度量指标。

第三章主要介绍了传统 EM 算法并给出了详细的参数估计推导过程，同时提出了 EM 算法对基于无监督高斯混合模型进行参数估计求解的具体理论及步骤。

第四章 EM 算法的改进，先提出了半监督聚类学习，接着基于最大熵提出了一种 EM 改进算法——SSL-MEEM 算法。

第五章实验，先在服从不同分布的一维、二维模拟数据集上对 K-Means 以及 EM 算法进行对比实验，并证明了 K-Means 算法与 EM 算法在一维数据上的聚类效果相似且均有较好的聚类效果，而在二维数据上的应用，K-Means 算法的收敛速度快于 EM 算法，但其在收敛得到的最优解不如 EM 算法。接着在服从高斯混合分布的模拟数据集上运用 EM 算法及其改进算法进行聚类，得到基于半监督学习的 EM 算法能够在一定程度上减小初始值对聚类结果干扰程度，同时基于最大熵的 EM 算法在面对存在异常标定样本有较好的稳定性和鲁棒性。最后在 UCI 传统数据集 seeds，wine，customers，FashionMNIST 上进行应用测试，

第六章总结与展望，主要对本文对研究及成果进行回顾和总结，并对以后的研究工作进行了展望。

2 聚类的相关知识

这一部分将会对类的基本定义、相似度度量方法、聚类的评价准则以及聚类典型方法进行一般性的介绍，为下文中，对 EM 算法在聚类中的研究工作打下基础。

2.1 类的定义

由于在现实生活中客观事物存在形式千差万别，加之我们在选择特征变量的时候用的方法也不尽相同使得对于不同问题，关于类的定义也不相同。因此给类进行一个一定的定义是有必要的，本文提出以下几种不同的类的定义，来适应几种常见的不同应用场景。

设 C 表示一个拥有 n 个样本的类，C_i 表示其中的样本，$d(C_i, C_j)$ 为样本 C_i 与样本 C_j 之间的距离，T, V 为预设距离阈值，且满足 $0 \leq V \leq T$。

定义 2.1 如果对于 $\forall C_i, C_j \in C$，都有 $d(C_i, C_j) \leq T$，则称 C 为一个类。

定义 2.2 如果对于 $\forall C_i \in C$，都有 $\frac{1}{n-1}\sum_j d(C_i, C_j) \leq T$，则称 C 为一个类。

定义 2.3 如果对于 $\forall C_i, C_j \in C$，都有

$$\frac{1}{n \times (n-1)} \sum_i \sum_j d(C_i, C_j) \leq T \text{且} d(C_i, C_j) \leq V,$$

则称 C 为一个类。

定义 2.4 如果对于 $\forall C_i \in C$，都 $\exists C_j \in C$，满足 $d(C_i, C_j) \leq T$，则称 C 为一个类。

以上这四种定义的要求严格性依次降低。即当满足定义2.1要求的类也一定满足其他几种定义，当满足定义2.2的类必定满足定义2.3。

另外，我们常用以下几个值来描述类的特征：

(1) 样本均值，即样本中心

$$\mu = \frac{1}{n}\sum_i C_i.$$

(2) 样本协方差矩阵，表示样本点之间的离散程度

$$\Sigma = \frac{1}{n-1}\sum_i (C_i - \mu)(C_i - \mu)^T.$$

(3) 类的直径

$$R = \frac{1}{n-1}\sum_i (C_i - \mu)^T (C_i - \mu).$$

2.2 相似度度量

聚类分析的基本思想是根据对象间相似度将对象划分为不同的类，将相似度极高的对象分为一类，将相似度极低的对象分到不同的类。而在实际生活中数据对象的属性变量类型复杂，下图展示了对于不同类型变量一般采用的相似度计算方法。

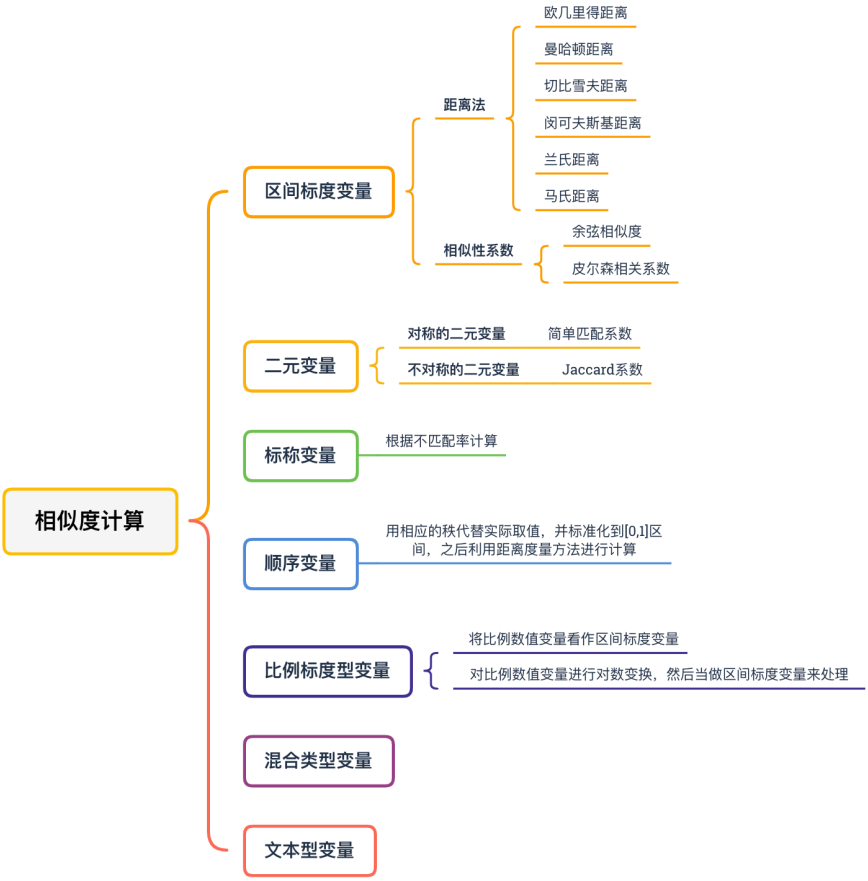

图 2.1 不同数据类型的相似度计算方法

在介绍聚类算法之前，在本节先对区间标度变量以及混合类型变量的相似度度量方法进行详细的介绍。

2.2.1 区间标度变量的相似度度量

基本呈线性的连续变量，典型的例子包括重量和高度、大气温度等。对于这类变量。通常度量标准有两种：距离和相似性系数。

(1) 距离法

基于距离的聚类算法是把距离较近的点可以归入同一类，距离远的点归入不同的类。由于类的分布情况是多种多样的，所以类与类之间的距离也有多种运算方法。

定义一个合适的距离函数 $d(x,y)$，通常需要满足以下四个性质：

- 非负性　$d(x,y) \geq 0$；
- 同一性　$d(x,x) = 0$；
- 对称性　$d(x,y) = d(y,x)$；
- 三角不等式性　$d(x,y) \leq d(x,z) + d(z,y)$。

假设 X 是一个 N 维欧氏空间中有 K 个数据对象集合，其中每个数据对象都有 $n(n \leq N)$ 个属性，取 $X_i = (X_{i1}, X_{i2}, \ldots, X_{in}), X_j = (X_{j1}, X_{j2}, \ldots, X_{jn}) \in X$，$d(X_i, X_j)$ 表示对象 X_i 与 X_j 之间的距离。

（i）曼哈顿距离（Manhattan Distance）

曼哈顿距离表示了多维空间内两个点之间的折线距离，又称为绝对距离。其计算公式如下

$$d(X_i, X_j) = \sum_{k=1}^{n} |X_{ik} - X_{jk}|.$$

（ii）欧几里得距离（Euclidean Distance）

欧几里得距离是计算距离最简单的方法，它定义了多维空间内点与点之间的直线距离。其计算公式如下

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^{n} (X_{ik} - X_{jk})^2}.$$

通常情况下，在计算欧式距离的时候，需要考虑各属性对对象具有不同的影响程度，也就是对象的各属性具有不同的权重，为了让各个属性得以更好表达对象的特征，引入加权欧氏距离，即

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^{n} \omega_k (X_{ik} - X_{jk})^2}, \quad 0 < \omega_k < 1, \quad \sum_{k=1}^{n} \omega_k = 1.$$

（iii）切比雪夫距离（Chebyshev Distance）

切比雪夫距离是向量空间中的一种度量，是两点坐标数值差绝对值的最大值。从数学角度来看，切比雪夫距离是由 1-范数所衍生的度量。其计算公式如下

$$d(X_i, X_j) = \max_{1 \le k \le n} |X_{ik} - X_{jk}|.$$

（iv）闵可夫斯基距离（Minkowski Distance）

闵可夫斯基距离的计算公式如下

$$d(X_i, X_j) = [\sum_{k=1}^{n} (X_{ik} - X_{jk})^q]^{1/q}, \quad q \ge 1.$$

可以明显发现当 $q = 1$ 时，Minkowski 距离就是曼哈顿距离；当 $q = 2$ 时，Minkowski 距离就是欧氏距离；当 $q \to \infty$ 时，Minkowski 距离就是切比雪夫距离

（v）兰氏距离（Lance Distance）

兰氏距离对数据的量纲不敏感，解决了闵可夫斯基距离与各属性的量纲有关的缺点。其计算公式如下

$$d(X_i, X_j) = \frac{1}{n} \sum_{k=1}^{n} \frac{|X_{ik} - X_{jk}|}{X_{ik} + X_{jk}}.$$

（vi）马氏距离（Mahalanobis Distance）

马氏距离是根据距离做判别，设 S 为样本协方差矩阵，其计算公式如下

$$d(X_i, X_j) = \sqrt{(X_i - X_j)^T S^{-1} (X_i - X_j)}.$$

(2) 相似度系数

（i）夹角余弦

$$cos\theta_{ij} = \frac{X_i \dot{X}_j}{|X_i||X_j|} = \frac{\sum_{k=1}^{n} X_{ik}X_{jk}}{\sqrt{\sum_{k=1}^{n} X_{ik}^2}\sqrt{\sum_{k=1}^{n} X_{jk}^2}}.$$

（ii）皮尔森相关系数

$$r_{ij} = \frac{\sum_{k=1}^{n}(X_{ik} - \bar{X}_i)(X_{jk} - \bar{X}_j)}{\sqrt{\sum_{k=1}^{n}(X_{ik} - \bar{X}_i)^2}\sqrt{\sum_{k=1}^{n}(X_{jk} - \bar{X}_j)^2}}.$$

其中 $\bar{X}_i = \frac{1}{n}\sum_{k=1}^{n} X_{ik}$。

2.2.2 混合类型变量的相似度度量

在实际数据集中，数据对象的属性往往是复合型的，因此研究混合类型变量的相似度度量也格外重要。下提出两种对混合类型变量进行相似度度量的方法。

- 方法一：将混合类型变量按照变量类型分类，然后单独对每个类型变量进行相似度度量。但此方法在真实践中并不实用
- 方法二：将不同类型的变量组合在单个相异度矩阵中，把所有有意义的变量进行归一化处理。

下对方法二进行详细解释：

设数据集 X 包含 p 种不同类型的变量，则定义混合类型变量的相异度 $d(i,j)$ 为

$$d(i,j) = \frac{\sum_{q=1}^{p} \delta_{ij}^{(q)} d_{ij}^{(q)}}{\sum_{q=1}^{p} \delta_{ij}^{(q)}}.$$

其中若数据对象 i 或 j 的变量 q 为缺失值（即无测量值），或数据对象 i 或 j 的变量 q 的值均为 0 且变量 q 为非对称二元变量，则记 $\delta_{ij}^{(q)} = 0$；否则 $\delta_{ij}^{(q)} = 1$。

$d_{ij}^{(q)}$ 表示变量 q 对对象 i 与 j 的直接相异度，它的计算方法与变量类型有关。

- 若变量 q 为二元变量，则

$$d_{ij}^{(q)} = \begin{cases} 0 & X_{iq} = X_{jq} \\ 1 & X_{iq} \neq X_{jq}. \end{cases}$$

- 若变量 q 为间隔数值变量，则 $d_{ij}^{(q)} = \frac{|X_{iq} - X_{jq}|}{\max_h X_{jq} - \min_h X_{hq}}$. 其中 h 为所有拥有变量 q 的对象序号。
- 若变量 q 为顺序变量，设变量 q 有 M_q 个有序状态，X_{iq} 可以用等级 $\{1, 2, \ldots, M_q\}$ 来代替表示，记替换后得到的值为 r_{iq}，又由于每个顺序变量所拥有的有序状态数量不等，因此将 r_{iq} 映射到 [0,1] 区间，得到 $z_{iq} = \frac{r_{iq}-1}{M_q-1}$，接着有 $d_{ij}^{(q)} = \frac{|z_{iq} - z_{jq}|}{\max_h z_{jq} - \min_h z_{hq}}$.

2.3 聚类算法的性能度量

常用聚类算法的评价指标来度量聚类算法的性能，也就是评估聚类算法聚类得到的结果的质量。一般来说，聚类性能度量分为两大类，一类是直接利用数据集的属性特征来评价

聚类算法的优劣，称为“内部质量评价指标”；另一类需要借助已知分类标签数据集进行评价，称为“外部评价指标”。

2.3.1 内部质量评价指标

对数据集 $X=\{X_1,X_2,\ldots,X_K\}$，设通过聚类得到的簇划分为 $C=\{C_1,C_2,\ldots,C_K\}$，定义

$$avg(C)=\frac{2}{|C|(|C|-1)}\sum\nolimits_{1\le i<j\le|C|}d(X_i,X_j), \tag{2.1}$$

$$diam(C)=\max\nolimits_{1\le i<j\le|C|}d(X_i,X_j), \tag{2.2}$$

$$d_{min}(C_i,C_j)=\min\nolimits_{X_i\in C_i,X_j\in C_j}d(X_i,X_j), \tag{2.3}$$

$$d_{cen}(C_i,C_j)=d(\mu_i,\mu_j). \tag{2.4}$$

其中,$d(X_i,X_j)$ 表示样本 X_i,X_j 之间的距离，μ 表示簇 C 的中心点 $\mu=\frac{1}{|C|}\sum_{1\le i\le|C|}X_i$，$avg(C)$ 表示簇 C 内样本间的平均距离，$diam(C)$ 代表簇 C 内样本间最远距离，$d_{min}(C_i,C_j)$ 代表簇 C_i,C_j 最近样本间的距离，$d_{cen}(C_i,C_j)$ 表示簇 C_i,C_j 中心点之间的距离。

基于式 (2.1)∼(2.4) 可推导出以下这些常用的聚类内部质量评价指标[15]

(1) DB 指数 (Davies-Bouldin Index，DBI)

$$DBI=\frac{1}{K}\sum_{i=1}^{K}\max_{j\ne i}\left(\frac{avg(C_i)+avg(C_j)}{d_{cen}(\mu_i,\mu_j)}\right). \tag{2.5}$$

(2) Dunn 指数 (Dunn Index，DI)

$$DI=\min_{1\le i\le K}\left\{\min_{j\ne i}\left(\frac{d_{min}(C_i,C_j)}{\max_{1\le l\le K}diam(C_l)}\right)\right\}. \tag{2.6}$$

(3) 轮廓系数 (Silhouette Coefficient，SC)

$$S(i)=\frac{b(i)-a(i)}{max\{a(i),b(i)\}}. \tag{2.7}$$

其中 $a(i)$ 表示簇内不相似度，$b(i)$ 为簇间不相似度。

通过公式可以看出，一个好的聚类算法需要使得 DBI 的值越小越好，DI 的值越大越好，SC 的值越大越好。其中 DBI，DI 的取值范围为 [0,1]，而 SC 的取值范围为 [-1,1]。

2.3.2 外部评价指标

已知分类标签数据的簇划分为 $C^*=\{C_1^*,C_2^*,\ldots,C_K^*\}$。另外，令 λ 与 λ^* 分别表示 C 和 C^* 对应的簇标记向量，于是有以下定义

$$a=|SS|,\quad SS=\{(X_i,X_j)|\lambda_i=\lambda_j,\lambda_i^*=\lambda_j^*,i<j)\}. \tag{2.8}$$

$$b=|SD|,\quad SD=\{(X_i,X_j)|\lambda_i=\lambda_j,\lambda_i^*\ne\lambda_j^*,i<j)\}. \tag{2.9}$$

$$c=|DS|,\quad DS=\{(X_i,X_j)|\lambda_i\ne\lambda_j,\lambda_i^*=\lambda_j^*,i<j)\}. \tag{2.10}$$

$$d=|DD|,\quad DD=\{(X_i,X_j)|\lambda_i\ne\lambda_j,\lambda_i^*\ne\lambda_j^*,i<j)\}. \tag{2.11}$$

其中集合 SS 表示包含 C 中属于相同簇企且在 C^* 中也属于相同簇的样本对，a 表示集

合 SS 中的样本对数量；集合 SD 表示包含 C 中属于相同簇企但在 C^* 中不属于相同簇的样本对，b 表示集合 SD 中的样本对数量，……

由于每个样本对 $(X_i, X_j)(i < j)$ 仅能出现在上面的其中一个集合中，因此有 $a+b+c+d = n(n-1)/2$。

基于式 (2.8)∼(2.11) 可推导出以下这些常用的聚类外部评价指标：

(1) Jaccard 系数 (Jaccard Coefficient)

$$JC = \frac{a}{a+b+c}. \tag{2.12}$$

(2) FM 指数 (Fowlkes and Mallows Index, FMI)

$$FMI = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}. \tag{2.13}$$

(3) Rand 指数 (Rand Index)

$$RI = \frac{2(a+d)}{n(n-1)}. \tag{2.14}$$

(4) 调整 Rand 指数 (Adjusted Rand index, ARI)

$$ARI = \frac{RI - E(RI)}{max(RI) - E(RI)}. \tag{2.15}$$

(5) 标准化互信息 (Normalized Mutual Information, NMI)

$$NMI(C^*, C) = \frac{MI(C^*, C)}{max\{H(C^*), H(C)\}}. \tag{2.16}$$

其中 $MI(C^*, C)$ 表示 C^* 和 C 之间的互信息，$H(.)$ 表示它们的熵。

显然，上述指标中 JC、FMI、RI、NMI 的结果值均在 [0,1] 区间内，ARI 的取值范围在 [-1,1] 之间，且值越大说明该聚类结果越好。

3 EM 算法及其在高斯混合模型中的应用

在本章节将对 EM 算法进行深入的理论研究推导。先概述其传统算法，接着介绍了高斯混合模型的原理，最后运用 EM 算法来求解高斯混合模型，并给出详细公式推导。

3.1 EM 算法

3.1.1 EM 算法的含义

EM(Expectation-Maximization algorithm) 算法最初是由 Ceppellini[16] 等人在上世纪五十年代生物学领域讨论基因频率估计等时候提出来的，之后由 Dempster，Rubin，Laind 于 1977 年正式在论文提出 EM 算法是求参数极大似然估计的一种方法。它可以对非完整数据集的模型参数进行最大似然估计[17][18]。所谓非完整数据集（不完全数据），一般指两种：一种是由于观察本身的失误导致观察数据成为缺失或错误的不完全数据；另一种是人为引入的参数（隐含或是丢失）用于优化参数的似然函数使其变与求解，从而使得原始数据变为“完全数据”，而原始数据就是一类“不完全数据”。

如今 EM 算法被广泛应用于求解混合聚类模型，隐式马尔科夫算法 (Hidden Markov Model，HMM) 以及 LDA(Latent Dirichlet Allocation) 主题模型的变分推断等。

3.1.2 EM 算法的原理

EM 算法的主要思想是预先设置一个隐变量，通过使用启发式的迭代法，不断求模型估计参数的极大对数似然，直至算法收敛，即模型参数基本没有变化，此时可以说得到的模型参数是最优的。

通过上述对 EM 算法的思想描述可以总结出，EM 算法主要由 E 和 M 步交替迭代进行。下对 EM 算法具体原理进行推导解析。

设 $x = \{x_1, x_2, \ldots, x_m\}$ 为拥有 m 个观察数据对象的样本。根据 EM 算法的基本算法思想——极大似然估计法，设观测数据 x 的总体概率分布函数为 P，则观测值的联合分布函数被定义为

$$L(x;\theta) = \prod_{i=1}^{m} P(x = x_i;\theta). \tag{3.1}$$

又将该函数视为参数 θ 的函数，称为样本的似然函数。为方便之后求解，对其求对数得

$$\log L(x;\theta) = \sum_{i=1}^{m} \log P(x = x_i;\theta). \tag{3.2}$$

假设这样本中含有未观察到的隐含参数 $z = \{z_1, z_2, \ldots, z_m\}$，则该样本的对数似然函数 $l(\theta)$ 定义为

$$l(\theta) = \sum_{i=1}^{m} \log \sum_{z_i} p(x_i, z_i;\theta). \tag{3.3}$$

最终目标是通过极大似然估计来求解最优参数 θ，即

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{m} \log \sum_{z_i} p(x_i, z_i; \theta). \tag{3.4}$$

然而直接求解是十分困难的，于是提出构造对数似然函数的下界函数，然后通过不断优化这个下界，迭代逼近最优参数。

为方便 EM 算法的公式推导，先提出以下几个函数定义：

(1) 随机变量的数学期望

$$E(X) = \sum_{x \in X} xP(X = x). \tag{3.5}$$

(2) 随机变量函数的数学期望

$$E(g(x)) = \sum_{x \in X} g(x)P(X = x). \tag{3.6}$$

(3) 相对墒（Kullback-Leibler divergence）

两个概率分布间差异的非对称性度量

$$KL(P||s) = \sum_{x \in X} P(X = x) \frac{P(X = x)}{Q(X = x)}. \tag{3.7}$$

下进行具体的公式推导：

首先引入隐变量 Z 的概率分布函数 $Q(Z)$，且 $Q(Z)$ 满足

$$\sum_{Z} Q(Z) = 1. \tag{3.8}$$

又有以下等式成立

$$P(X; \theta) = \frac{P(X, Z; \theta)}{P(Z; X, \theta)}. \tag{3.9}$$

则有式 (3.8) 和 (3.9) 可得等式 (4.7)

$$P(X; \theta) = \frac{P(X, Z; \theta)/Q(Z)}{P(Z; X, \theta)/Q(Z)}. \tag{3.10}$$

对等式 (4.7) 两边取对数得

$$\log P(X; \theta) = \log \frac{P(X, Z; \theta)/Q(Z)}{P(Z; X, \theta)/Q(Z)}. \tag{3.11}$$

再对等式 (3.11) 两边对 Z 求期望得

$$E_Z[\log P(X; \theta)] = E_Z[\log \frac{P(X, Z; \theta)/Q(Z)}{P(Z; X, \theta)/Q(Z)}]. \tag{3.12}$$

对等式3.12的左边进行化简计算，由于 $\log P(X;\theta)$ 与 Z 无关，因此

$$E_Z[\log P(X;\theta)] = \log P(X;\theta). \tag{3.13}$$

根据式 (3.6), 式 (3.7), 对等式3.12的右边进行化简计算

$$\begin{aligned} E_Z[\log \frac{P(X,Z;\theta)/Q(Z)}{P(Z;X,\theta)/Q(Z)}] &= \sum_Z \log \frac{P(X,Z;\theta)/Q(Z)}{P(Z;X,\theta)/Q(Z)} Q(Z) \\ &= \sum_Z Q(Z) \log \frac{P(X,Z;\theta)}{Q(Z)} + \sum_Z Q(Z) \log \frac{Q(Z)}{P(Z;X,\theta)} \\ &= \sum_Z Q(Z) \log \frac{P(X,Z;\theta)}{Q(Z)} + KL(Q(Z)||P(Z;X,\theta)) \\ &\geq \sum_Z Q(Z) \log \frac{P(X,Z;\theta)}{Q(Z)}. \end{aligned} \tag{3.14}$$

由此得到对数似然函数的下界，即

$$\log P(X,Z;\theta) \geq \sum_Z Q(Z) \log \frac{P(X,Z;\theta)}{Q(Z)}. \tag{3.15}$$

则最终可通过最大化这个下界得到我们的目标

$$\begin{aligned} \hat{\theta} &= \arg\max_\theta \sum_{i=1}^m \sum_{z_i} Q(Z=z_i) \log \frac{P(X=x_i, Z=z_i;\theta)}{Q(Z=z_i)} \\ &= \arg\max_\theta \sum_{i=1}^m \sum_{z_i} Q(Z=z_i) \log P(X=x_i, Z=z_i;\theta). \end{aligned} \tag{3.16}$$

其中求解 $\sum_{z_i} Q(Z=z_i) \log P(X=x_i, Z=z_i;\theta)$ 相当于对 $\log P(X=x_i, Z=z_i;\theta)$ 求基于条件概率分布 $Q(Z=z_i)$ 的期望，因此该步骤称为 E 步；接着极大化对数似然函数的下界的步骤称为 M 步。

以上就是 EM 算法原理及理论推导，下给出 EM 算法的一般流程。

表 3.1 EM 算法流程

输入： 观察数据 $x = \{x_1, x_2, \ldots, x_m\}$，联合分布 $p(x, z; \theta)$, 条件分布 $p(z|x; \theta)$, 最大迭代次数 J

过程：

step 1： 随机初始化模型参数 θ 并令其为 θ_0

step 2： **for** j in range(J) **do**

E-step： 计算联合分布的条件概率期望

$$Q(Z = z_i) = p(z_i|x_i, \theta_j)$$

$$L(\theta, \theta_j) = \sum_{i=1}^{m} Q(Z = z_i) \log p(x_i, z_i; \theta)$$

M-step： 极大化 $L(\theta, \theta_j)$，并求得 θ_{j+1}

$$\theta_{j+1} = \arg\max_{\theta} L(\theta, \theta_j)$$

step 3： **repeat** step 2; **until** θ_{j+1} 收敛

输出： 模型参数 θ

3.2 基于 EM 算法的高斯混合模型

3.2.1 高斯混合模型

大量独立同分布的随机变量的均值在经过适当标准化后会依分布收敛于高斯分布，因此高斯混合模型假设所有的数据均服从此混合高斯分布合理，并且每个高斯分布为一个簇。高斯混合模型实际上是一个软聚类过程，通过将样本数据按照概率大小分属于各个簇，而不是完全的属于某一个簇。通过将各个高斯分布进行加权求和，然后将样本数据分布投影到这些高斯分布上，得到这些样本点被划分在各个簇上的概率，最终选取概率最大的簇作为该数据点最终的划分结果。

现有一系列观测值服从某混合分布 P，且该分布由 K 个高斯分布组成，其中每个高斯分布分别代表一个不同的类别。假设观测样本为 $X = \{x_1, x_2, \ldots, x_n\}$，其中 $x_i \in R^{1\times p}$。则该混合分布 P 可由 K 个高斯密度函数的加权平均所表示的概率密度函数来描述，即

$$P(x|\theta) = \sum_{k=1}^{K} \pi_k f(x|\theta_k). \tag{3.17}$$

其中 π_k 表示第 k 个高斯分布在高斯混合模型中所占权重，也就是选中第 k 类的概率，这个参数一般是未知的，且满足

$$\sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0. \tag{3.18}$$

另外 $f(x|\theta_k)$ 表示第 k 个高斯分布的密度函数，同时 θ_k 表示第 k 个高斯模型参数，由于该分布服从高斯分布，因此可将该密度函数表示为

$$f(x|\theta_k) = \frac{exp\{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\}}{(2\pi)^{p/2}|\Sigma_k|^{1/2}}. \tag{3.19}$$

其中 μ_k 表示第 k 个高斯分布的均值，Σ_k 表示第 k 个高斯分布的协方差矩阵。

由以上分析可得，需要估计的高斯混合模型参数有三个，即混合系数 π_k，均值 μ_k，以及协方差矩阵 Σ_k。而估计模型参数最常用的方法就是极大化对数似然函数。其中，高斯混合模型的对数似然函数可以表示为

$$l(x|\theta,\pi)=\sum_{i=1}^{n}\ln[\sum_{k=1}^{K}\pi_k f(x_i|\theta_k)]. \tag{3.20}$$

3.2.2 EM 算法求解高斯混合模型参数

通常情况下，估计模型参数是通过概率模型中已知的参数来估计最优混合模型参数。然而由上一小节内容可知，样本数据集是一个完整数据，包含了观测得到的随机样本数据 X，以及未观测得到的隐含变量 $Z=\{z_1,z_2,\ldots,z_n\}, z_i\in R^{1\times K}$，且

$$z_{ij}=\begin{cases}1 & x_i\in k\\ 0 & x_i\notin k.\end{cases}$$

因此无法使用似然估计的方法进行求解。此时存在未知变量 z_{ik},π_k,θ_k，为估计这些参数，根据贝叶斯概率思想，通过初始化先验概率，求解后验概率，最后通过选择最大后验概率来完成参数估计。这过程类似 EM 算法流程。因此下对基于 EM 算法对高斯混合模型参数进行求解。

设隐变量 z_i 独立同分布，则首先对先验概率进行如下定义：

$$P(z)=\prod_{k=1}^{K}p(z_i)=\prod_{k=1}^{K}(\pi_k)^{z_i}. \tag{3.21}$$

则条件似然函数可定义为

$$P(x|z)=\prod_{k=1}^{K}N(x|\mu,\sigma^2)^z. \tag{3.22}$$

最后根据贝叶斯定理，可得后验概率为

$$\begin{aligned}P(z|x_i,\hat{\theta})&=P(z_{ik}=1|x_i,\hat{\theta})\\&=\frac{p(z_{ik}=1)P(x_i|z_{ik}=1)}{P(x=x_i)}\\&=\frac{\pi_k N(x_i|\mu_k,\Sigma_k)}{\sum_{k=1}^{K}\pi_k N(x_i|\mu_k,\Sigma_k)}.\end{aligned} \tag{3.23}$$

由于 z_{ik} 表示第 i 个样本数据属于 k 的概率，因此 $z_{ik}=P(z|x_i)=\frac{\pi_k N(x_i|\mu_k,\Sigma_k)}{\sum_{k=1}^{K}\pi_k N(x_i|\mu_k,\Sigma_k)}$

根据 EM 算法的基本原理思想[18]，根据式 (3.15) 得高斯混合模型条件概率期望为

$$L(\theta,\theta_k)=\sum_{i=1}^{n}\sum_{k=1}^{K}z_{ik}[\ln\pi_k f(x_i|\theta_k)]. \tag{3.24}$$

下对高斯混合模型各参数进行具体求解推导。

(i) $\mu_k^{(i+1)}$ 的迭代更新公式

将式 (3.24) 对 μ_k 求偏导，并令偏导为 0，得到

$$\frac{\partial L(\theta,\theta^{(i)})}{\partial \mu_k}=\sum_{j=1}^{n} z_{jk}^{(i)}\Sigma_k^{-1}(x_j-\mu_k)=0. \tag{3.25}$$

求解上式，可得到 $\mu_k^{(i+1)}$ 的迭代更新公式

$$\mu_k^{(i+1)}=\frac{\sum_{j=1}^{n} z_{jk}^{(i)}x_j}{\sum_{j=1}^{n} z_{jk}^{(i)}}. \tag{3.26}$$

(ii) $\Sigma_k^{(i+1)}$ 的迭代更新公式

将式 (3.24) 对 Σ_k 求偏导，并令偏导为 0，得到

$$\frac{\partial L(\theta,\theta^{(i)})}{\partial \Sigma_k}=\sum_{j=1}^{n} z_{jk}^{(i)}((x_j-\mu_k)(x_j-\mu_k)^T-\Sigma_k)=0. \tag{3.27}$$

求解上式，可得到 $\Sigma_k^{(i+1)}$ 的迭代更新公式

$$\Sigma_k^{(i+1)}=\frac{\sum_{j=1}^{n} z_{jk}^{(i)}(x_j-\mu_k)(x_j-\mu_k)^T}{\sum_{j=1}^{n} z_{jk}^{(i)}}. \tag{3.28}$$

(iii) $\pi_k^{(i+1)}$ 的迭代更新公式

由于 π_k 满足一个等式条件 $\sum_{k=1}^{K}\pi_k=1$，因此引入拉格朗日算子 λ，对对数似然函数 L 构造拉格朗日函数

$$L(\theta,\lambda)=L(\theta,\theta^{(i)})+\lambda(\sum_{k=1}^{K}\pi_k-1). \tag{3.29}$$

对式 (3.29) 关于 π_k 求偏导，并令偏导为 0，

$$\frac{\partial L(\theta,\lambda)}{\partial \pi_k}=\sum_{i=1}^{n}\frac{f(x_i|\theta_k)}{\sum_{k=1}^{K}\pi_k f(x_i|\theta_k)}+\lambda=0. \tag{3.30}$$

两边同乘以 π_k 得

$$\sum_{i=1}^{n}\frac{\pi_k f(x_i|\theta_k)}{\sum_{k=1}^{K}\pi_k f(x_i|\theta_k)}+\lambda\pi_k=0. \tag{3.31}$$

化简可得到

$$\pi_k=\frac{\sum_{i=1}^{n} z_{ik}}{\lambda}. \tag{3.32}$$

将上式带回 $\sum_{k=1}^{K}\pi_k=1$ 得到

$$\lambda=-\sum_{k=1}^{K}(\sum_{i=1}^{n}z_{ik})=-n. \tag{3.33}$$

因为 $\sum_{i=1}^{n}z_{ik}$ 表示样本中属于第 k 类的样本数量，则 $\sum_{k=1}^{K}(\sum_{i=1}^{n}z_{ik})$ 就表示总样本数 n。

将式 (3.33) 带入式 (3.32) 可得 $\pi_k^{(i+1)}$ 的迭代更新公式

$$\pi_k^{(i+1)}=\frac{\sum_{j=1}^{n}z_{jk}^{(i)}}{n}. \tag{3.34}$$

以上就是对高斯混合模型参数迭代公式的详细推导，下给出基于 EM 算法求解高斯混合模型的算法流程

表 3.2 基于 EM 算法求解高斯混合模型的算法流程

输入： 观察数据 $x=\{x_1,x_2,\ldots,x_n\}$，最大迭代次数 J

过程：

step 1： 随机初始化模型参数 $\theta^{(0)}=\{\mu^{(0)},\Sigma^{(0)},\pi^{(0)}\}$

step 2： **for** j in range(J) **do**

E-step： 计算模型混合系数 $z^{(j)}$ 以及对数似然函数 $L(\mu_j,\Sigma_j)$

$$z_{ik}^{(j)}=(\pi_k f(x_i|\mu_k,\Sigma_k))/(\sum_{k=1}^{K}\pi_k f(x_i|\mu_k,\Sigma_k))$$

$$L(\mu_j,\Sigma_j)=\sum_{i=1}^{n}\sum_{k=1}^{K}z_{ik}^{(j)}[\ln\pi_k\frac{exp\{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\}}{(2\pi)^{p/2}|\Sigma_k|^{1/2}}]$$

M-step： 最大化对数似然函数，并更新模型参数估计值

$$m_k=\sum_{i=1}^{n}z_{ik}^{(j)}$$

$$\pi_k=m_k/(\sum_{k=1}^{K}m_k)$$

$$\mu_k=(\sum_{i=1}^{n}z_{ik}^{(j)}x_i)/m_k$$

$$\Sigma_k=(\sum_{i=1}^{n}z_{ik}^{(j)}(x_i-\mu_k)(x_i-\mu_k)^T)/m_k$$

step 3： **repeat** step 2; **until** 对数似然函数 $L(\mu_j,\Sigma_j)$ 收敛

输出： 模型参数 θ

4 基于半监督机器学习的 EM 算法改进

通过资料查阅可以知道，对于 EM 算法，国内外学者已经提出了很多改进方法。如 PX-EM 算法[21] 是利用协方差对 M 步的计算进行修改，从而加快算法收敛速度；ECEM 算法[22] 是通过简化 E 步来加快收敛速度；另外还有最常见的 MCEM 算法[23]，它是用蒙特卡洛的方法对 EM 算法中期望显示表示进行了改善，从而也加速了收敛速度。但综上所述的这些改进方法在针对 EM 算法局部最优问题上还没有提出有效的方法。对此本章提出在半监督学习下的一种基于最大熵的 EM 算法改进方法。即首先在无监督学习中加入一部分已标记的样本，通过半监督机器学习方法确定 EM 算法初始值；然后通过最大熵原理来约束隐变量的后验概率。

4.1 半监督学习的聚类

半监督学习 (Semi-Supervised Learning, SSL) 由 Merz[24] 等首次提出，并且将半监督用于分类。接着由 Shahshahani 与 Landgrebe 于 1994 年证明了用无标签样本能减缓小样本下的"Hughes" 现象[25]，从此使得无标签样本和半监督学习得到大量学者关注与研究，半监督学习逐渐发展起来。传统的聚类是一种基于无监督学习的分析方法，属于对数据探索分析，在聚类过程中完全不依赖于任何实际背景，单纯根据数据分布情况按照相似性进行划分。但对于现实生活场景数据进行聚类，为了贴合实际情形，需要对部分数据进行一定的信息标注，并且在现实生活中的数据, 存在少量带有独立标签或成对约束的监督信息的数据样本, 只是这部分数据量不足以进行监督学习。学者们致力于将这些为数不多的监督信息运用于聚类, 以得到更优的聚类结果, 从而提出了半监督聚类[26]。即使对于那些完全不带标签的数据集，可以由特定方面专家人员对少量数据进行信息标注，这样可以使得聚类结果的性能更好，更贴合实际。因此半监督学习聚类结合少量已标记样本和大量无标记样本利用先验信息指导聚类过程既解决了监督学习耗时耗力的弊端，又有效提高了无监督学习聚类算法的泛化性能和准确性[27]。

4.2 基于半监督学习的 EM 算法 (SSL-EM)

在上一小节中介绍了半监督学习在聚类中应用的优势，接下来，在本小节中运用半监督学习方法初始化高斯混合模型参数，接着使用 EM 算法估计其参数。由于在3.2.2节中已详细介绍了 EM 算法求解高斯混合模型参数的推导过程，因此在本小节中不再重复推导公式。而是直接给出表达式以及算法步骤。

设有数据集 $D=\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$，包含标记样本数据集 $S=\{(x_1,y_1),(x_2,y_2),\ldots,(x_c,y_c)\}$，以及未标记样本数据集 $T=\{x_{c+1},x_{c+2},\ldots,x_n)\}$，$c<<n$，$y_i\in\{1,2,\ldots,K\}$。假设数据集 D 中所有样本独立同分布，且均服从一个高斯混合分布。则基于半监督学习的高斯混合模型的对数似然函数为

$$L(\theta,\theta_k)=\sum_{(x_i,y_i)\in S}\sum_{k=1}^{K}\ln(\pi_k f(x_i|\theta_k)z_{ik})+\sum_{x_i\in T}\sum_{k=1}^{K}\ln(\pi_k f(x_i|\theta_k)). \tag{4.1}$$

下给出算法具体步骤：

表 4.1 SSL-EM 算法流程

Step 1： 利用已标记样本计算高斯混合模型参数的初始值 $\theta^{(0)}=\{\mu^{(0)},\Sigma^{(0)}\}$

$$\mu_k^{(0)}=\frac{\sum_{(x_i,y_i)\in S_k}x_i}{\#S_k}.$$
$$\Sigma_k^{(0)}=\frac{\sum_{(x_i,y_i)\in S_k}(x_i-\mu_k^{(0)})(x_i-\mu_k^{(0)})^T}{\#S_k}.$$
$$\pi_k^{(0)}=\frac{\#S_k}{\sum_{k=1}^{K}\#S_k}.$$

其中 S_k 表示已标记样本中属于第 k 类的样本集合，$\#S_k$ 表示数据集 S_k 的样本数量。

Step 2： 根据第 i 轮得到的参数估计值 $\theta^{(i)}$，计算对数似然函数。

$$z_{jk}^{(i)}=\frac{\pi_k^{(i)}f(x_j|\theta_k^{(i)})}{\sum_{k=1}^{K}\pi_k^{(i)}f(x_j|\theta_k^{(i)})},$$
$$L(\theta,\theta^{(i)})=\sum_{k=1}^{K}\sum_{(x_j,y_j)\in S_k}\ln(\pi_k^{(i)}f(x_j|\theta_k^{(i)}))+\sum_{x_j\in T}\sum_{k=1}^{K}z_{jk}^{(i)}\ln(\pi_k^{(i)}f(x_j|\theta_k^{(i)})).$$

Step 3： 根最大化 L 得到第 $i+1$ 轮的参数估计值 $\theta^{(i+1)}$，并对各模型参数进行迭代更新。

$$\theta^{(i+1)}=\arg\max_{\theta}L(\theta,\theta^{(i)}).$$
$$\mu_k^{(i+1)}=\frac{\sum_{(x_j,y_j)\in S_k}x_j+\sum_{x_j\in T}z_{jk}^{(i)}x_j}{\#S_k+\sum_{x_j\in T}z_{jk}^{(i)}}.$$
$$\Sigma_k^{(i+1)}=\frac{\sum_{(x_j,y_j)\in S_k}(x_j-\mu_k^{(i+1)})(x_j-\mu_k^{(i+1)})^T+\sum_{x_j\in T}z_{jk}^{(i)}(x_j-\mu_k^{(i+1)})(x_j-\mu_k^{(i+1)})^T}{\#S_k+\sum_{x_j\in T}z_{jk}^{(i)}}.$$
$$\pi_k^{(i+1)}=\frac{\sum_{x_j\in T}z_{jk}^{(i)}+\#S_k}{n}.$$

Step4: 不断重复 Step2,3，直到对数似然函数 L 收敛。

4.3 基于半监督学习的最大熵 EM 算法 (SSL-MEEM)

我们知道在3.2.2节 EM 算法求解高斯混合模型参数中的式 (3.23)，即隐变量 Z 的后验概率，是通过贝叶斯公式推导得到的。可以看出受参数估计值 $\hat{\theta}$ 的影响很大，也正因为如此 EM 算法的结果在很大程度上受参数初始值的影响。特别是在第一次迭代过程中隐变量的后验概率严重依赖于初始值 θ_0，当采用半监督学习方式的时候则会使得隐变量的后验概率依赖于已有标签样本特征，若已有标签样本特征与整体差异较大或存在标签标错的样本，此时如果已有标签样本数不大的情况下，很容易造成最终结果偏离实际。通过上一章的实验过程也发现，对若初始值 θ_0 设置的与参数真实值 θ 相差很大的时候，由式 (3.23) 得到的隐变量后验概率的估计会非常不准确，从而影响 EM 算法的后续迭代过程，导致最终结果远偏离于真实。因此本节中提出采用最大熵的方法来约束每次迭代过程中隐变量的后验概率，可以避免 EM 算法在寻找最优解的过程中偏离方向收敛到局部最优，从而优化 EM 算法局部最优问题。

4.3.1 最大熵

二十世纪中叶，C.E.Shannon[28] 首次提出了信息熵这个概念，他在其论文中指出“信息是用来消除随机不确定性的东西”。信息熵是所有可能发生事件所带来的信息量的期望。并在其论文中给出了具体的计算公式。

定义 4.1 设 $X=\{x_i,x_2,\ldots,x_n\}$ 为离散型随机变量，则其熵的计算公式可定义为

$$Ent(X)=-\sum_{i=1}^{n}p_i\log_2 p_i. \tag{4.2}$$

其中，$p_i=P(X=x_i)$ 表示 X 取值为 x_i 的概率。

定义 4.2 设连续型随机变量 X 的概率密度函数为 $f(x)$，定义域为 $[a,b]$，则其熵的计算公式被定义为

$$Ent(X)=-\int_a^b f(x)\log_2 f(x)dx. \tag{4.3}$$

定义 4.3 设随机变量 $X=\{x_1,x_2,\ldots,x_n\}$ 服从 m 个已知约束条件 $E_i(g_i(X))=C_i,i=1,2,\ldots,m.$ 其中 $g_i(X)$ 为关于随机变量 X 的函数，$E_i(g_i(X))$ 一般表示 $g_i(X)$ 的期望，C_i 为常数。则基于这些约束条件的最大熵为

$$\max \quad Ent(X)=-\sum_{i=1}^{q}p(x_i)\log_2 p(x_i).$$

$$s.t\begin{cases}\displaystyle\sum_{i=1}^{q}p(x_i)=1,\\ E_i(g_i(X))=C_i,\quad i=1,2,\ldots,m.\end{cases} \tag{4.4}$$

信息熵作为一个随机事件“不确定性”的度量指标，当一随机事件越不确定，它的熵就越大。根据以上计算公式可看出，若该随机事件是个确定事件，则它的熵为 0；若该随机事件结果是均匀分布的，则它的熵为 $\log_2 n$。也就是信息熵的值域为 $[0,\log_2 n]$。

从上述公式定义可以看出，信息熵可由随机变量的概率分布或概率密度计算得到。1957 年 Jaynes[29] 提出了最大熵原理，证明了可以通过随机变量的熵值求其概率分布函数或概率密度函数。

最大熵原理的核心思想是：在预测随机变量的概率分布时，在充分依赖已知信息情况下，且不对任何未知信息进行假设，信息量的增加会引起信息熵的减少，此时会使得结果偏离实际。因此在已知信息的条件下，令熵值最大的概率分布的未知信息的假设最少，此时随机事件的概率分布是最均匀的、最符合实际的。

4.3.2 MEEM 算法

MEEM 算法是在传统 EM 算法基础上通过改变原来运用贝叶斯公式推导求解隐变量的后验概率的方法来改进 EM 算法的局部最优问题。

下对隐变量的后验概率进行详细推导。首先和传统 EM 算法一样，$\forall j=1,2,\ldots,n,$ 隐变量 z 的后验概率满足以下等式:

$$\sum_{k=1}^{K}P(z_j=k|x_j,\theta^{(0)})=1. \tag{4.5}$$

此时假设没有关于隐变量 z 的后验概率的先验知识，即不能使用贝叶斯公式进行求解，

而选择最大熵原理对其进行求解。也就是求在变量 X 和模型参数 $\theta^{(0)}$ 条件下 z 的条件熵。

$$\begin{aligned} Ent(z|X,\theta^{(0)}) &= -\sum_{j=1}^{n}\sum_{k=1}^{K} P(z_j=k|x_j,\theta^{(i)})\ln P(z_j=k|x_j,\theta^{(i)}) \\ &= -\sum_{(x_j,y_j)\in S}\sum_{k=1}^{K} P(z_j=k|x_j,\theta^{(i)})\ln P(z_j=k|x_j,\theta^{(i)}) \\ &\quad -\sum_{x_j\in T}\sum_{k=1}^{K} P(z_j=k|x_j,\theta^{(i)})\ln P(z_j=k|x_j,\theta^{(i)}). \end{aligned} \tag{4.6}$$

其中，当 $(x_j,y_j)\in S$ 时，可通过观测数据真实直接得到 $P(z_j=k|x_j,\theta^{(i)})$，而不需要通过估计得到。

$$P(z_j=k|x_j,\theta^{(i)}) = \begin{cases} 1, & k=y_j; \\ 0, & k\neq y_j. \end{cases} \tag{4.7}$$

于是根据信息熵的性质，由于 S 是个确定性事件，因此有以下恒等式

$$-\sum_{(x_j,y_j)\in S}\sum_{k=1}^{K} P(z_j=k|x_j,\theta^{(i)})\ln P(z_j=k|x_j,\theta^{(i)}) \equiv 0.$$

则 $Ent(z|X,\theta^{(0)})$ 可化简为

$$Ent(z|X,\theta^{(0)}) = -\sum_{x_j\in T}\sum_{k=1}^{K} P(z_j=k|x_j,\theta^{(i)})\ln P(z_j=k|x_j,\theta^{(i)}). \tag{4.8}$$

当 $x_j\in T$ 时，与3.2.2节中采用相同方法，引入拉格朗日算子。根据定义4.3可知在 $\theta^{(i)}$ 已知的情况下，样本的完全数据对数似然函数关于隐变量后验分布的期望是个定值，因此隐变量 z 的后验概率还满足以下条件：

$$\begin{cases} \displaystyle\sum_{(x_j,y_j)\in S}\sum_{k=1}^{K} \ln P(X=x_j,Y=y_j|\theta^{(i)})P(z_j=k|x_j,\theta^{(i)}) = C_1, \\ \displaystyle\sum_{x_j\in T}\sum_{k=1}^{K} \ln P(X=x_j,z_j=k|\theta^{(i)})P(z_j=k|x_j,\theta^{(i)}) = C_2. \end{cases} \tag{4.9}$$

根据约束条件 (4.5),(4.9) 引入拉格朗日算子 $\lambda_1,\lambda_2,\gamma_1,\gamma_2$，对条件熵 $Ent(z|X,\theta^{(i)})$ 构造拉格朗日函数。由于我们的目标是求最大熵，即 $\max Ent(z|X,\theta^{(i)})$，而拉格朗日乘子法适用于

求解最小化问题，因此构造得到的拉格朗日函数如下：

$$\begin{aligned} L(z,\lambda_1,\lambda_2,\gamma_1,\gamma_2) = &\sum_{x_j\in T}\sum_{k=1}^{K} P(z_j=k|x_j,\theta^{(i)})\ln P(z_j=k|x_j,\theta^{(i)}) \\ &+\sum_{(x_j,y_j)\in S}\lambda_{1j}(1-\sum_{i=1}^{K}P(z_j=k|x_j,\theta^{(i)})) \\ &+\sum_{x_j\in T}\lambda_{2j}(1-\sum_{k=1}^{K}P(z_j=k|x_j,\theta^{(i)})) \\ &+\gamma_1(C_1-\sum_{(x_j,y_j)\in S}\sum_{i=1}^{K}\ln P(X=x_j,Y=y_j|\theta^{(i)})P(z_j=k|x_j,\theta^{(i)})) \\ &+\gamma_2(C_2-\sum_{x_j\in T}\sum_{k=1}^{K}\ln P(X=x_j,z_j=k|\theta^{(i)})P(z_j=k|x_j,\theta^{(i)})). \end{aligned} \tag{4.10}$$

于是我们的目标转化为

$$\min_{P(z_j=k|x_j,\theta^{(i)})} L(z,\lambda_1,\lambda_2,\gamma_1,\gamma_2)$$

对拉格朗日函数 L 关于 $P(z_j=k|x_j,\theta^{(i)})$ 求偏导，并令其为 0。又由式 (4.7) 可得

$$\sum_{(x_j,y_j)\in S}\lambda_1(1-\sum_{i=1}^{K}P(z_j=k|x_j,\theta^{(i)}))=b_1.$$

$$\sum_{(x_j,y_j)\in S}\sum_{i=1}^{K}\ln P(X=x_j,Y=y_j|\theta^{(i)})P(z_j=k|x_j,\theta^{(i)})=b_2.$$

其中 b_1,b_2 是常数。因此得到

$$\frac{\partial L}{\partial P(z_j=k|x_j,\theta^{(i)})}=\ln P(z_j=k|x_j,\theta^{(i)})+1-\lambda_{1j}-\gamma_2\ln P(X=x_j,z_j=k|\theta^{(i)})=0.$$

$$P(z_j=k|x_j,\theta^{(i)})=exp(\gamma_2\ln P(X=x_j,z_j=k|\theta^{(i)})+\lambda_{1j}-1). \tag{4.11}$$

带入式 (4.5) 可得

$$exp(\lambda_{1j}-1)=\frac{1}{\sum_{k=1}^{K}exp(\gamma_2\ln P(X=x_j,z_j=k|\theta^{(i)}))}. \tag{4.12}$$

可以观察得到，当 $\gamma_2=0$ 的时候，后验概率退化为均匀分布概率；当 $\gamma_2=1$ 的时候，即为贝叶斯公式求解得到的后验概率。因此 γ_2 的取值范围为 $(0,1)$ 将式 (4.12) 代入式 (4.11)

得到

$$z_{ik}^{(i)} = P(z_j = k|x_j, \theta^{(i)}) = \frac{P(X = x_j, z_j = k|\theta^{(i)})^{\gamma_2}}{\sum_{k=1}^{K} P(X = x_j, z_j = k|\theta^{(i)})^{\gamma_2}}. \\ = \frac{f(x_j|\theta_k^{(i)})^{\gamma_2}}{\sum_{k=1}^{K} f(x_j|\theta_k^{(i)})^{\gamma_2}}. \tag{4.13}$$

至此，已分析解释了运用最大熵的方法求解隐变量后验概率，也通过公式证明了该方法求得的隐变量后验概率更具有鲁棒性。下通过对4.2节中基于半监督学习的 EM 算法的算法流程进行改进得到以下算法流程。（具体公式不再赘述）

表 4.2 SSL-MEEM 算法流程

输入：设定参数 γ_2 为一个较小的正初始值，且满足 $0 < \gamma_2 < 1$

过程：

step 1：利用已标定样本计算高斯混合模型参数的初始值 $\theta^{(0)} = \{\mu^{(0)}, \Sigma^{(0)}\}$

step 2 （E-step）：根据参数估计值 $\theta^{(i)}$ 以及更新后的 γ_2，计算隐变量后验概率 $z_{ik}^{(i)}$ 并计算完全数据对数似然函数 L，将其中式 (4.1) 替换为 (4.13)

step 3（M-step）：最大化 L 得到参数估计值 $\theta^{(i+1)}$，并对各模型参数进行迭代更新；同时对参数 γ_2 进行更新 $\gamma_2 = \min(\alpha\gamma_2, 1)$，其中 $\alpha > 1$

step 3：repeat step 2，3; **until** 对数似然函数 L 收敛，$\gamma_2 = 1$

输出：模型参数 θ

5 实验

在本章中首先通过随机产生服从不同分布类型的一维数据集以及二维数据集分别利用 EM 算法与 K-Means 算法对数据集进行聚类，进而根据聚类内外部指标对聚类效果进行评判，同时研究两算法的时间复杂度和收敛速度。接着对基于半监督学习的 EM 算法 (SSL-EM) 以及基于半监督学习的 MEEM 算法 (SSL-MEEM) 进行了算法实现，并通过在模拟数据集，人工数据集以及真实数据集上分别利用 EM 算法及其改进算法进行聚类，验证了基于半监督学习的聚类算法具有更好的稳定性，且 MEEM 算法受标记样本的随机性影响较小，能够在一定程度上避免算法收敛到局部最优从而使得模型获得更好的聚类效果。本章实验环境为 Python3.7。

5.1 EM 算法与 K-Means 算法对比实验

本小节为研究 EM 算法与 K-Means 算法在聚类中的应用情况及其性能，通过在一维数据集以及二维数据集进行测试，并模拟产生不同分布的数据集，通过聚类算法内外部评价指标以及算法执行时间来对比两算法的稳定性、鲁棒性以及时间复杂度。

5.1.1 一维数据集模拟实验

通过三组随机产生的不同分布的模拟数据集对基于 EM 算法的高斯混合模型聚类与 K-Means 聚类进行对比。每组数据集包括 6000 个随机数，从而可以更好的分析 EM 算法的鲁棒性。通过研究在均匀分布数据集中的应用，可以观察算法在临界情况下的性能；同时高斯分布虽是比较理想的数据分布情况，但在现实生活中大多数数据集的分布可以近似看为高斯分布，因此研究算法在高斯分布下的应用，可以得出算法的实用性；而通过模拟算法在布朗运动中的应用，可以观察算法在复杂情形下的鲁棒性。

a. 均匀分布一维随机数

首先通过 random 包中的 uniform 函数随机产生（100，200）区间内的服从均匀分布的 6000 个随机数。该数据集分布情况如图5.1所示。

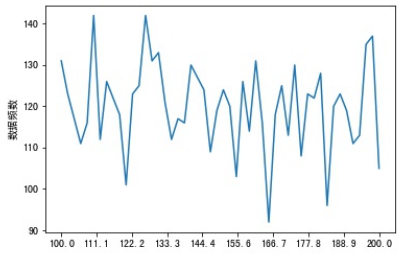

图 5.1 均匀分布数据集

b. 正态分布一维随机数

接着通过 random 包中的 normal 函数随机从 $N(100,5), N(150,20)$ 的两个高斯分布中各产生 3000 个数，得到一个混合高斯分布数据集。该数据集分布情况如图5.2所示。

c. 几何布朗运动随机数

布朗运动是连续时间情况下最基础的随机过程之一，几何布朗运动在金融数学中有较多应用[19]。下对几何布朗运动进行简单解释。

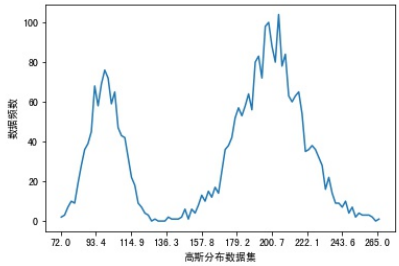

图 5.2 高斯分布数据集

定义 5.1 当随机过程 $B(t)$ 的每个时间段 Δt 的增量都相互独立，且服从正态分布 $N(0, \Delta t)$，则将这个随机过程称为布朗运动。

定义 5.2 若随机过程 $W(t)$ 满足以下随机微分方程，则称该随机过程为几何布朗运动。

$$dW(t) = \mu W(t)dt + \sigma dB(t). \tag{5.1}$$

其中 $B(t)$ 为布朗运动，μ, σ 为常量。

定义 5.3 根据伊藤公式[20]，可得对式 (5.1) 进行求解得到几何布朗运动的通式 (5.2)

$$W(t) = W(0)e^{(\mu - \frac{\sigma^2}{2})t + \sigma B(t)}. \tag{5.2}$$

本实验中分别选取初始值 10，20，各进行 3000 次连续均匀采样，得到两个几何布朗运动轨迹，得到数据集分布如图5.3所示

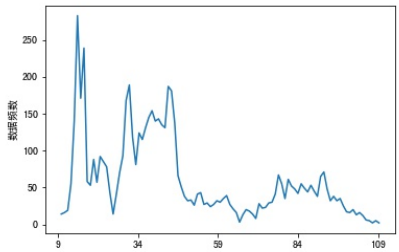

图 5.3 几何布朗运动

对上述三种不同的分布数据集分别聚类成 2 类和 3 类，并将属于同一类别的数在原始数据分布图中用同一水平线 (如图5.4中绿色水平线) 进行标注。下分别将均匀分布一维数据集、高斯分布一维数据集以及一维布朗运动数据集称为数据集 a，数据集 b 和数据集 c。由于此实验中的模拟数据集不存在原始标签，因此对聚类效果的评价指标选择为内部指标 DBI(2.5) 以及 SC(2.7)。具体结果如下：

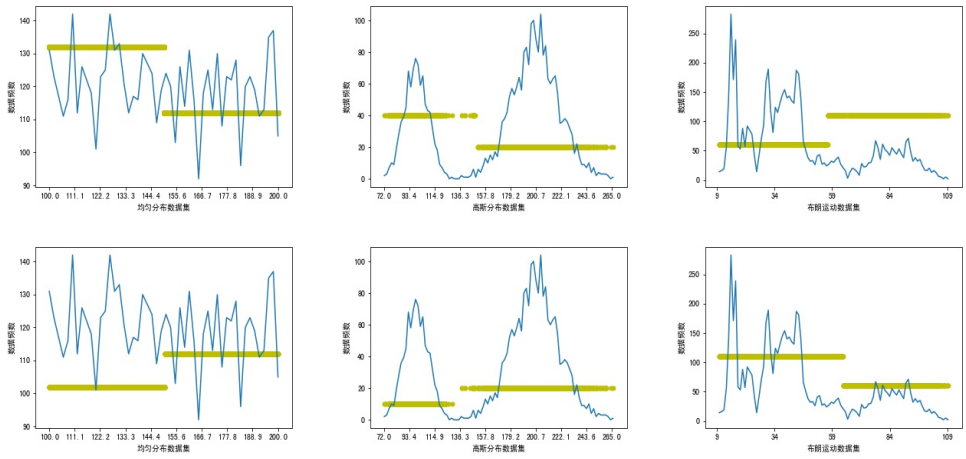

图 5.4 K=2 的聚类效果.

图5.4是对三个数据集分别进行 2 聚类后的效果图，第一行是由 K-Means 算法聚类得到的效果图，第二行则是由 EM 算法聚类得到的，从左到右的数据集依次服从均匀分布、高斯分布以及布朗运动（下同）。从该图可以明显看出，对于 K=2，两个算法的聚类效果基本一致。

表 5.1 K-Means 算法和 EM 算法对一维数据集的聚类效果.

算法	K-Means		EM	
指标	DBI	SC	DBI	SC
数据集 a	0.497	0.628	0.497	0.628
数据集 b	0.241	0.799	0.236	0.799
数据集 c	0.399	0.706	0.397	0.708

通过表5.1能够更清晰的看出，两个算法在每个一维数据集上的聚类效果相当，且均表现良好。虽然其中对数据集 b 的聚类效果最佳。

5.1.2 二维数据集模拟实验

常见的二维数据分布有均匀分布、正态分布。本小节的实验，通过随机产生三组分别服从两种不同分布的二维随机数据集，并对不同数据集分别运用 K-Means 算法和 EM 算法进行聚类，通过比较聚类内外部评价指标以及执行时间对聚类效果进行评判分析。本次实验设置最大迭代次数为 15 次，同时 K-Means 算法与 EM 算法给予相同的初始值。

a. 均匀分布二维随机数

通过 random 包中的 uniform 函数随机产生服从 U(1.5,2.5)，U(5.5,7.5)，U(2.0,6.0) 的三个均匀分布，且每个分布中各产生 300 组二维随机数。该数据集分布情况如图5.5所示。

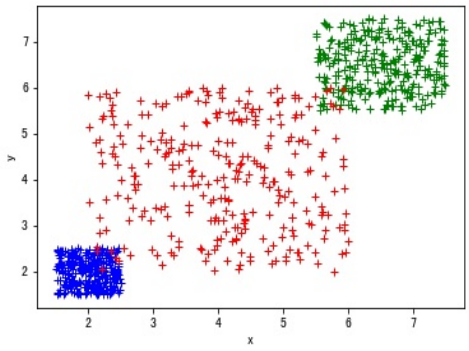

图 5.5 二维均匀分布数据集

记录 K-Means 算法以及 EM 算法收敛后分别得到的聚类结果的指标，包括 SC、DBI、ARI、FMI，以及算法执行时间。得到如表5.2所示。

表 5.2 K-Means 算法和 EM 算法对二维均匀分布数据集的聚类效果.

评价指标 / 聚类算法	SC	DBI	ARI	FMI	执行时间 (s)
K-Means	0.6208	0.6324	0.8441	0.8963	0.7314
EM	0.6118	0.6231	0.9139	0.9426	7.0609

从表5.2所示，K-Means 算法的执行时间大大短于 EM 算法，根据内部指标（SC 以及 DBI）无法明显比较两算法对该数据集的聚类效果，但根据外部指标（ARI 以及 FMI）可以看出 EM 算法对该数据集的聚类效果较优于 K-Means。

b. 二维正态分布随机数

与产生一维正态分布随机数产生方法类似，利用 random 包中的 normal 函数随机从 N(1,0.3)，N(3,0.7)，N(-1,5.1) 三个正态分布中产生 300 组二维随机数。该数据集分布情况如图5.6所示。

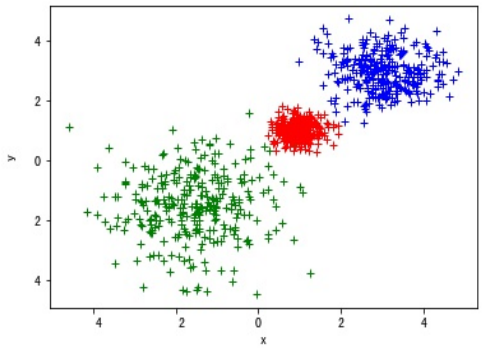

图 5.6 二维均匀分布数据集

同样对上述二维正态分布数据集运用 K-Means 算法以及 EM 算法进行聚类，得到如表5.3所示结果。

表 5.3 K-Means 算法和 EM 算法对二维正态分布数据集的聚类效果.

聚类算法 \ 评价指标	SC	DBI	ARI	FMI	执行时间 (s)
K-Means	0.6331	0.4499	0.9267	0.9511	0.7667
EM	0.6206	0.4496	0.9889	0.9926	7.4075

通过表5.3可以看出同样 K-Means 算法的执行时间远短于 EM 算法，但最终聚类效果不如 EM 算法。横向比较，可以看出对于 K-Means 算法来说，相较于均匀分布，更适用于正态分布。而对于 EM 算法则对两种分布都有较好的适应性。

为研究 K-Means 算法与 EM 算法的收敛性，同样将最大迭代次数设置为 15，算法收敛条件统一设置为 $|L(\theta^{(i+1)},\theta^{(i)})-L(\theta^{(i)},\theta^{(i-1)})|<1$，绘制两个算法分别在两种分布数据集上聚类结果的 FMI 随迭代次数的变化曲线图，如图5.7所示。

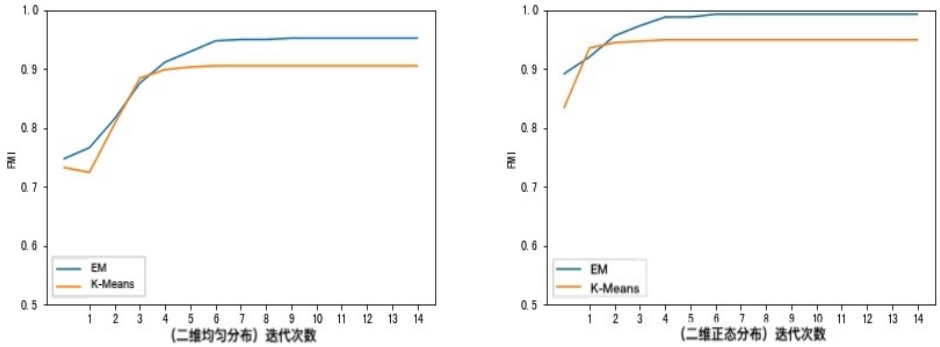

图 5.7 二维实验数据 FMI 随迭代次数变化图.

从图5.7可以看出，虽然 K-Means 算法的收敛速度明显快于 EM 算法，但 EM 算法最终收敛得到的结果优于 K-Means 算法。

5.2 EM 算法及其改进算法的数值模拟对比实验

5.2.1 数据集定义

本小节中通过随机数值模拟的方式产生满足如下混合高斯模型的二维数据，

$$f(x|\theta)=\sum_{k=1}^{K}\pi_k\frac{1}{\sqrt{2\pi}^2|\Sigma_k|^{1/2}}exp(\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)), \tag{5.3}$$

其中

$$K=3$$

$$\pi_1=0.2,\quad \pi_2=0.4,\quad \pi_3=0.4$$

$$\mu_1=(-1,2)^T,\quad \mu_2=(3,1)^T,\quad \mu_3=(2.5)^T$$

$$\Sigma_1=\begin{pmatrix}0.774 & -0.268\\ -0.268 & 0.158\end{pmatrix},\Sigma_2=\begin{pmatrix}0.265 & 0.194\\ 0.194 & 0.197\end{pmatrix},\Sigma_2=\begin{pmatrix}0.912 & 0.023\\ 0.023 & 0.065\end{pmatrix}$$

本小节将随机产生满足该模型的 40 组数据作为样本数据，并保留它们真实标记以便模型评价。其中每组数据包含 1000 个样本点，并按分层抽样方法选取已标注样本集（已标记样本与总样本个数的比值设为 10%）。通过传统 EM 算法，SSL-EM 算法以及 SSL-MEEM 算法利用未标记样本数据对模型参数进行估计，并对这些样本进行标记预测。

5.2.2 SSL-MEEM 算法中超参数 γ_2、α 的调优

首先选取第 5，20，30 组模拟实验数据作为该小节实验的实验数据，采用网格搜索（grid_search）的方法搜索 SSL-MEEM 算法中 γ_2、α 两参数的最优解组合。γ_2 的取值范围为 [0.1，0.3]，间隔为 0.05；α 的取值范围为 [1，3]，间隔为 0.5。通过比较在每组 (γ_2，α) 下算法收敛后得到的聚类模型的 FMI，搜索出超参数 (γ_2，α) 的最优解。下表展示了通过网格搜索得到各两参数组合下聚类模型的 FMI 值。

表 5.4 网格搜索超参数最优解结果表.

γ_2 \ α	1.0	1.5	2	2.5	3 .0
0.10	0.7670	0.8989	0.9631	**0.9796**	0.9735
0.15	0.8411	0.9719	0.9716	0.9428	0.9111
0.20	0.9230	0.8083	0.9110	0.8483	0.8169
0.25	0.9428	0.9423	0.8177	0.7203	0.6888
0.30	0.9720	0.7823	0.4609	0.6244	0.5928

表 5.4中加粗数值是表中 FMI 最大值，因此可以看出通过本文搜索方法得到的最优超参数组合 (γ_2，α) 为 (0.1，2.5)。并将该结果用于本文后续实验。

5.2.3 一般性已标定样本情况

首先，通过对 40 组模拟数据分别运用三种算法进行聚类，同时计算聚类算法的内部评价指标 DBI(2.5) 和 SC(2.7)、外部评价指标 FMI(2.13) 和 ARI(2.15)，以此来评价各算法模型的聚类效果。得到各算法在该混合高斯模型下的 40 组样本数据的平均聚类效果如表5.5所示。

表 5.5 三种算法聚类效果指标评价表.

聚类算法 \ 指标	SC	DBI	ARI	FMI
传统 EM	0.6690	0.4304	0.8994	0.9355
SSL-EM	0.6769	0.4164	0.9829	0.9533
SSL-MEEM	0.6786	0.4053	0.9830	0.9827

根据2.3节中介绍的聚类内外部评价指标标准，可从表5.5看出，四种算法对该模型进行的聚类效果从好到差依次为 SSL-MEEM、SSL-EM、传统 EM，符合本文的理论推导。但同时也能看出基于半监督学习的聚类算法总体聚类效果优于无监督学习的聚类算法，这也体现了半监督学习的必要性。

接着，为了更明显的比较传统 EM 算法与 SSL-EM 算法，选择第 10，15 组实验数据分别对两组数据进行 40 次已标注样本集随机选取（已标注样本集为每类别的 10%），将已标

注样本作为 SSL-EM 算法的初始值，同时进行 40 次随机从两组数据中每类各选取一个点作为 EM 算法聚类初始中心点。然后分别使用传统 EM 算法与 SSL-EM 算法，计算得到参数估计值得到的模型 FMI。结果如图5.8所示。

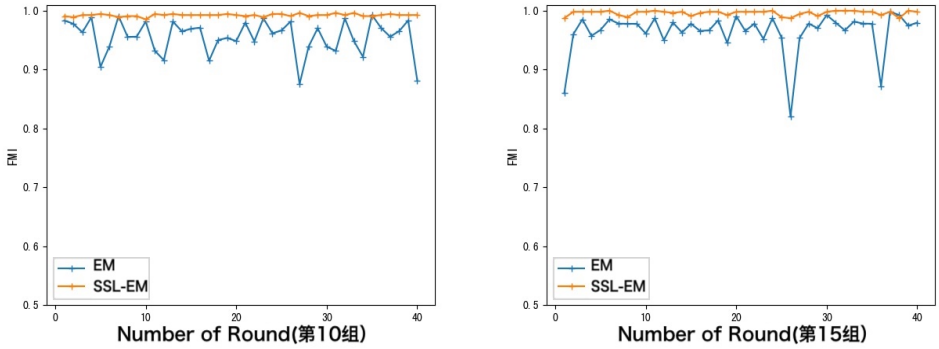

图 5.8 两组实验数据 FMI

从上折线图可以清晰看出，对于两组实验数据，分别进行的这 40 次模拟实验中，无论怎样随机选取已标记样本，利用 SSL-EM 算法得到的参数估计值对应的聚类模型有着很高的 FMI，均在 98% 以上，且不随已标记样本的随机选取而有较大波动；然而，通过随机选取聚类初始中心点的 EM 算法计算得到的参数估计值对应的模型中，随着初始样本点的随机选取不同，FMI 呈现较大波动，特别从第 15 组实验数据上可以看出第 26 次抽取的样本与真实数据情况相差甚远，从而导致通过传统 EM 算法聚类得到的模型 FMI 只有 82% 左右。因此从这整组实验中可以看出，初始值的选取对 SSL-EM 算法的干扰程度小于传统 EM 算法。

5.2.4 特殊已标定样本情况

又为验证 SSL-MEEM 算法相比 SSL-EM 算法是否更具有稳定性，选取第 20 组实验数据，模拟了以下在已标记样本为两种特殊情况下两算法的聚类效果。

- **已标定样本为异常点**

通过异常点选取方法选取第 20 组实验数据中每个类别中边缘特殊点作为已标定样本，来对比 SSL-EM 算法与 SSL-MEEM 算法。由于在实验中发现当 SSL-EM 算法收敛后的结果与 SSL-MEEM 算法几乎一致且聚类效果可观，因此综合考虑运行效率，设置迭代次数为 5 次，通过两种算法对第 20 组实验数据聚类结果如图5.9所示。

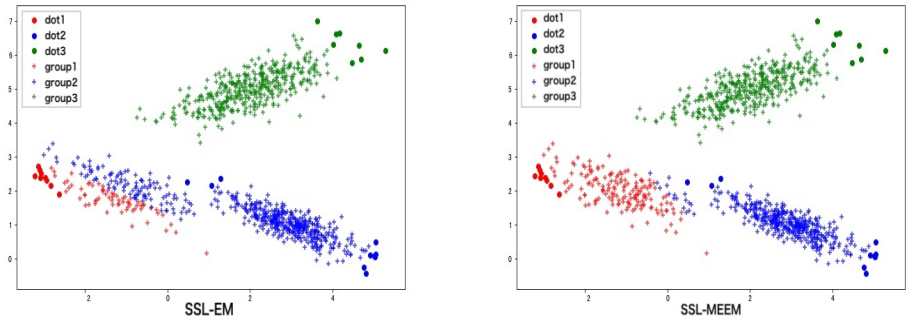

图 5.9 取异常点下第 20 组实验数据聚类效果图

从图5.9能看出 SSL-EM 算法聚类效果更容易受异常点的影响而使得结果有所偏离实际，而 SSL-MEEM 算法相较而言聚类效果更好一些。

• 已标定样本为不准确标记样本

通过手动模拟错误标记样本，来模拟当标记样本不准确的情况。处理后的第 20 组实验数据原始分类图如图5.10所示，其中每组中深色标记的为其对应浅色标记类别中已标记样本。标记样本点 1，2 是由原来的类别 1 手动改为类别 2，标记样本点 3，4，5 是由原来的类别 2 手动改为类别 3。

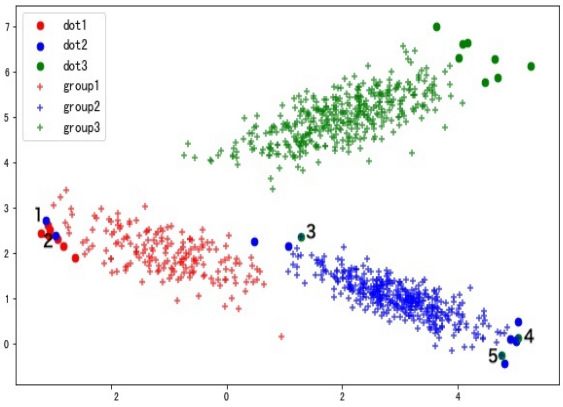

图 5.10 第 20 组实验数据原始分类图

接着分别利用 SSL-EM 算法与 SSL-MEEM 算法对该处理后的第 20 组实验数据进行聚类，同样都进行 5 次迭代后得到聚类结果，如图5.11所示。

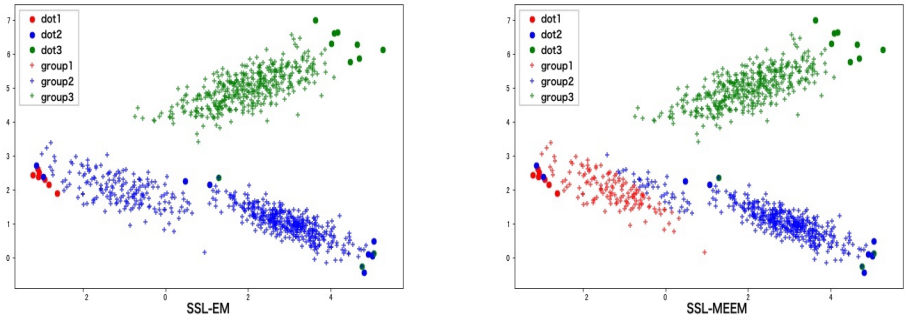

图 5.11 标记样本不准确时第 20 组实验数据聚类效果图

可从图5.11中对比发现，SSL-EM 算法由于标注样本的不准确而将第 20 组实验数据聚成了两类。而对于 SSL-MEEM 算法很明显不大受标注样本的准确性的影响，仍能将第 20 组实验数据聚成 3 类。因此通过该实验可以证明当已标定样本点无法较好反应其所代表类别整体情况时，SSL-MEEM 算法可以不依赖于已标注样本而在一定程度上避免收敛到局部最优。

通过上述数值模拟实验验证了 EM 算法及其改进算法在二维数据上的聚类效果，并着重对比了 EM 算法与 SSL-EM 算法对初始值的敏感性以及 SSL-EM 算法与 SSL-MEEM 算法在特殊标记样本下的稳定性。可看出，基于半监督学习的聚类算法聚类效果总体优于传统聚类算法；虽然 SSL-MEEM 算法并不能完全避免收敛到局部最优的情况，相较于其他算法模型 FMI 平均提高了 8%。且当已标记样本偏离真实数据分布情况时，SSL-MEEM 算法表现可观。说明 SSL-MEEM 算法具有较强的稳定性和鲁棒性。

5.3 EM 算法及其改进算法基于 UCI 传统真实数据集的对比实验

5.3.1 实验数据集

本小节选择了三个维度不同样本数量不等的传统 UCI 真实数据集 seeds,wine,customers。其中 seeds 与 customers 两个数据集类别分布均匀，而 wine 数据集类别分布不均匀；同时 seed，customers 数据集属性数量较少，而 wine 数据集的属性数量较多；另外 customers 数据量较大些。因此选取这三组数据集进行实验，具有一定研究意义。图5.6是 UCI 三组真实数据集信息表。

表 5.6 UCI 三组真实数据集信息表

数据集名称	样本数量	样本属性数量	样本类别
seeds	210	7	3
wine	178	13	3
customers	440	6	2

5.3.2 实验评价指标

本小节选择 FMI 指标 (2.13) 作为衡量聚类结果的指标，分别对传统 EM 算法、基于半监督学习的 EM 算法以及基于半监督学习的 MEEM 算法四种算法在不同数据集上进行实验对比分析。以通过对比三个算法在这三个数据集上的聚类效果，看出各算法随维度的升高的稳定性及其鲁棒性。

5.3.3 实验数据预处理

由于聚类算法是基于距离度量的，因此受属性量纲的影响，于是首先对每个数据集按属性进行归一化处理，此次实验中均选择 MinMaxScaler 进行归一化处理。通过数据探索发现：对于 wine 数据集，由于该数据集拥有 13 个特征属性，若保留所有属性，容易造成过拟合从而使得结果偏离实际。因此在归一化数据集的基础上对其进行 PCA 降维，将 13 个属性压缩到 5 个属性。而对于 customers 数据集，其特征数据不服从正态分布且存在异常值，因此首先对该数据集应用非线性缩放方法 Box-Cox[30] 使数据接近正态分布。至此完成实验数据的预处理。

5.3.4 实验参数设定

- **初始值设定**

所有实验中均采用 Python3.7 中 numpy 包下 random.uniform 方法在均匀分布中随机选取样本索引来初始化 K-Means 以及传统 EM 算法的聚类中心点。并采用 sklearn 下 Stratified-ShuffleSplit 方法通过分层抽样方法进行已标定样本和未标定样本的划分，从而确保 SSL-EM 算法以及 SSL-MEEM 算法中已标定样本的每个标签对应的样本的比例一致，同时将已标记样本与总样本个数的比值设为 10%。

- **算法收敛条件设定**

在每次实验中控制算法收敛的条件均为 $|L(\theta^{(i+1)},\theta^{(i)}) - L(\theta^{(i)},\theta^{(i-1)})| < 1$，且对于 SSL-MEEM 算法的另一个收敛条件为 $1-\gamma_2 <$ 1e-4。

5.3.5 实验结果及分析

在表5.9对比了在三个不同数据集上利用传统 EM 算法、SSL-EM 算法以及 SSL-MEEM 算法进行聚类算法收敛后的 FMI。

表 5.7 三种算法在不同数据集下聚类 FMI.

聚类算法 \ 数据集	seeds	wine	customers
传统 EM	0.919	0.712	0.638
SSL-EM	0.928	0.912	0.826
SSL-MEEM	0.958	0.908	0.818

从上表中可以看出，相较于通过随机初始化的传统 EM 算法，利用本文提出的 SSL-EM 算法以及对其改进后的 SSL-MEEM 算法得到的聚类模型的 FMI 值也大幅度提升，表明估计得到的模型参数更精确，在各个数据集上均有较好的聚类效果。横向对比，发现 SSL-MEEM 算法对 customers 数据集的适用性较差，说明 SSL-MEEM 算法也并不是一个对任何数据集都能精确估计得到模型的参数，也存在一定误差。这需要之后在大量不同数据集中测试，调整 SSL-MEEM 算法中 γ_2 以及 α 的值。另外，各算法在 wine，customers 上的聚类效果均欠佳，分析其原因和原始数据集本身有关。这也体现了针对不同数据集选择合适的聚类算法的重要性。

为研究各算法随迭代次数的变化对模型参数估计准确性的影响程度，对三个数据集分别进行 10 次随机初始值选取，即采用分层抽样方法对 seed 数据集进行 10 次已标定样本选取作为 SSL-EM、SSL-MEEM 算法的训练集，同时随机选取中心点作为 EM 算法的初始值。计算 10 次实验后各算法每次迭代后估计得到的模型 FMI 平均值，四个算法估计得到的模型 FMI 随迭代次数的变化情况如下图所示

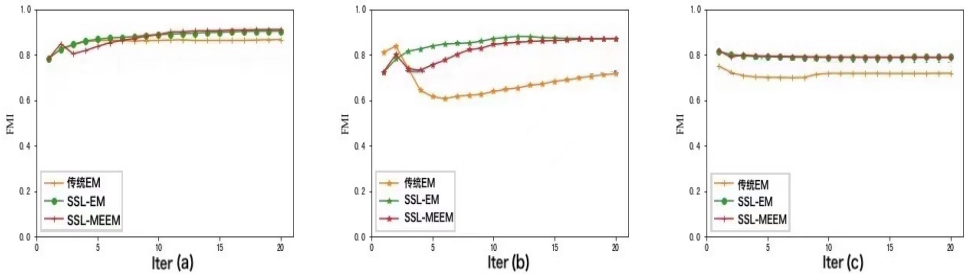

图 5.12 在 seed、wine、customer 上的 FMI 变化趋势

由图5.12(a) 所示，在 seed 数据集上 SSL-MEEM 算法在收敛后的 FMI 稍高于其余三种算法。传统 EM 算法收敛速度快，但搜索到的是局部最优解，而未能找到全局最优。从图中可以看出 SSL-MEEM 算法在迭代到第 3 次的时候 FMI 剧烈下降，分析训练集可分析得到此时由于各个类别的训练集样本比例不均衡，导致某个类别测试样本被聚类到其他一个类中。而通过不断迭代，可再次努力找回全局最优。

对于 wine 数据集，从图5.12(b) 中可看出 SSL-EM 对其聚类效果最佳。另外 SSL-MEEM 算法虽然对于 wine 数据集聚类效果不是最佳的，但其聚类准确率仍比较高，且在找到其中一个局部最优解之后没有立刻收敛，而是继续寻找最优解，使最终收敛后的 FMI 基本达到 90%。

从图5.12(c) 中可以看出，SSL-EM 与 SSL-MEEM 算法对 customers 数据集进行聚类的效果随迭代次数的变化趋势基本一致，且同样也均优于其余两个算法，

为更好的比较 SSL-EM 算法与 SSL-MEEM 算法，记录两算法在各数据集 10 组不同初始值下聚类收敛后的 FMI 最小值、最大值以及平均值。具体数值如表5.8所示。

表 5.8 SSL-EM 算法与 SSL-MEEM 算法聚类效果比较表

算法	SSL-EM			SSL-MEEM		
	最小值	最大值	平均值	最小值	最大值	平均值
seeds	0.805	0.931	0.898	0.856	0.938	0.903
wine	0.597	0.951	0.884	0.656	0.937	0.869
customers	0.767	0.809	0.790	0.789	0.830	0.811

从表5.8中可以看出 SSL-MEEM 算法相较于 SSL-EM 算法更具有稳定性和鲁棒性，受已标定样本的影响较小。对于 wine 数据集，虽然 SSL-MEEM 算法总体聚类效果不敌 SSL-EM 算法，但随已标定样本的选择结果波动性小。

5.4 EM 算法及其改进算法在图像识别上的应用

本小节中选用 FashionMNIST 数据集作为实验数据，该数据集包含 70000 张灰度图像（其中 60000 张为训练样本，10000 张为测试样本），共 10 个类别。每张图像的大小均为 28×28，像素值范围是 0 到 255。图5.13是其中 4 个类别中一张原始图像。

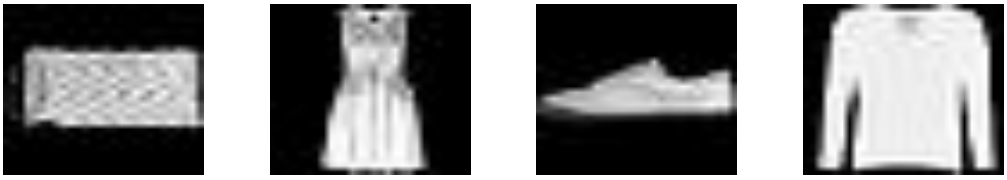

图 5.13 FashionMNIST 图像（部分）

首先对该图像数据进行预处理：先将各像素值缩放到 0 和 1 之间，也就是进行归一化处理；接着将每个二维图像数组 28×28 展开为一维数组（$28 \times 28 = 784$）；最后通过查阅资料选择使用 autoEncoder 自动编号[31] 方法提取 784 个像素点中 128 个重要信息点，保存为大小为 128 的一维数组。图5.14是图中 4 张图像经过 autoEncoder 重构后的效果图。

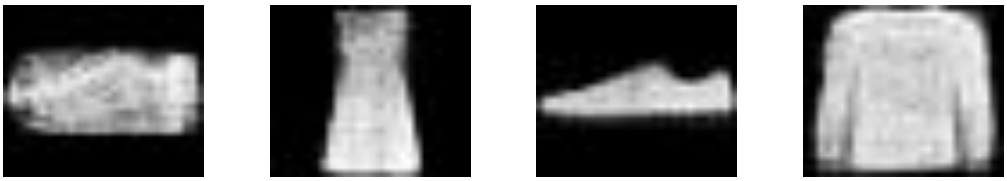

图 5.14 FashionMNIST 处理后图像（部分）

通过对比观察可发现，重构图像基本能够复原原始图像，辨识度较高，同时证明了自编码能为聚类提供有效的特征向量。下为了将聚类效果可视化，实验采用 t-SNE[32] 方法在两种不同二维特征空间中对 autoEncoder 的输出结果进行可视化。

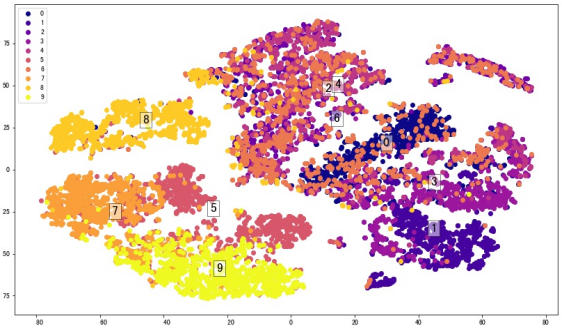

图 5.15 t-SNE 降维可视化效果图

接下来使用经过 autoEncoder 编码后的图像数据进行聚类，由于各属性之间相关性较强，从而导致在求解高斯混合密度函数的时候协方差矩阵会出现奇异的现象，因此对原始算法中计算密度函数 $f(x|\theta_k)$3.19进行微调，加入 L2 惩罚项 λ 后得：

$$f(x|\theta_k) = \frac{exp\{-\frac{1}{2}(x-\mu_k)^T(\Sigma_k+\lambda I)^{-1}(x-\mu_k)\}}{(2\pi)^{p/2}|\Sigma_k+\lambda I|^{1/2}} \tag{5.4}$$

其中 λ 取 1e-3。

由于该数据集已知标签，因此本次实验选择外部指标 ARI(2.15)、FMI(2.13) 以及 NMI(2.16) 作为模型评判指标，表5.4展示了 FashionMNIST 分别通过四种算法进行参数估计得到的模型聚类效果。

表 5.9 四种算法对 FashionMNIST 的聚类效果.

评价指标 / 聚类算法	ARI	FMI	NMI
传统 EM	0.3889	0.4634	0.5706
SSL-EM	0.6857	0.7178	0.7395
SSL-MEEM	0.6876	0.7196	0.7403

由表5.9可以看出通过传统 EM 算法聚类得到的效果较差，而利用给基于半监督学习的 EM 算法和 MEEM 算法得到的聚类效果相近，且聚类效果客观。

6 总结与展望

总结目前聚类分析领域研究内容，发现在实际应用中，高斯混合模型聚类是一种较为合适的聚类方法，它的目的是试图找到多维高斯模型概率分布的混合表示，从而拟合出任意形状的数据分布，因此适用范围广，且性能较好。但该模型中存在隐变量（不可观测的随机变量）。对于处理隐变量参数估计问题，采用 EM 算法，即添加“潜在变量”，有效地解决含有隐藏变量的问题。然而 EM 算法本身存在一定的缺陷，如：对参数初始值敏感，需要给予隐变量的所有可能的取值等。如今，对 EM 算法性能等改进主要有两方面：选择更合适的参数初始值设定方法，加速 EM 算法收敛。

因此本文在半监督学习的理论指导下，首先提出了基于半监督学习的 EM 算法，并通过理论推导以及在随机产生的服从高斯分布的二维数值、UCI 三组人工数据集和 FashionMNIST 数据集上进行实验，验证了相较于传统的 EM 算法，该算法能极大程度上减小 EM 算法迭代过程受初始值的影响程度，从而降低了收敛到局部最优的风险，提高了对高斯混合模型参数估计的性能。其次，当采用贝叶斯方法对隐变量后验概率进行计算时同样很大程度上受初始值（已标定样本）的不同选择而得到不同效果，因此本文提出了基于最大熵的 EM 算法（MEEM），通过约束每次迭代过程中计算的隐变量的后验概率，来减少模型受初始值的影响程度，同样通过模拟实验和真实数据集的实验，验证了该算法在对于存在异常初始值的情况下仍能较好地估计出模型参数，从而提升聚类效果。另外，本文对 MEEM 算法中超参数 γ_2、α 的取值进行研究，并通过添加 L2 惩罚项的方法解决高斯混合模型的协方差矩阵不可逆问题。

然而，本文对 EM 算法的改进方面主要是针对其初始值敏感性的问题，而从实验结果可以看出其执行速度和收敛速度并没有优于 K-Means 这样的传统聚类算法，因此将来可以对本文提出的 MEEM 算法继续做收敛速度的优化，如通过结合 EM 算法加速算法。另外，本文选择的数据集均是数值型，缺少非数值型属性，在后续研究中可以针对存在非数值型属性数据集的聚类进行研究，从而更全面地验证 EM 的改进算法的优缺点，扩大其适用范围。

参考文献

[1]（美）陈封能. 数据挖掘导论 (完整版)［M］. 北京：人民邮电出版社，2011.

[2] 王千，王成，冯振元等.K-Means 聚类算法研究综述 [J]. 电子设计工程,2012(07).

[3] 周恩波, 毛善君, 李梅等.GPU 加速的改进 PAM 聚类算法研究与应用 [J]. 地球信息科学学报，2017,19(6):782-791.

[4] Zhang T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases[J]. Acm Sigmod Record, 25(2), 103-114.

[5] Ester M, Kriegel H P, Sander, et al. A density based algorithm for discovering clusters in large spatial databases with noise[C]//In:Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining(KDD),1996,96:226-231.

[6] Andreea B. Dragut. Stock Data Clustering and Multiscale Trend Detection. Methodogy and Computing in Applied Probability[J], Vol. 14.I.1, March 2012, PP:87-105.

[7] Chen, Wei-chen,Ostrouchov, George. Michael. A Parallel EM Algorithm for Model-Based Clustering Applied to the Exploration of Large Spatio-Temporal Data[J]. Technometrics,2013,55(4).

[8] Kwedlo, Wojciech. A new random approach for initialization of the multiple restart EM algorithm for Gaussian model-based clustering[J]. Pattern Analysis and Applications , 2015 , 18(4):757-770.

[9] 岳佳. 基于 EM 算法的模型聚类的研究与应用［D］. 江苏无锡市：江南大学，2007.

[10] 夏筱筠, 张笑东, 王帅, 罗金鸣, 崔露露, 赵智阳. 一种半监督机器学习的 EM 算法改进方法 [J]. 小型微型计算机系统, 2020, 41(02): 230-235.

[11] Estelle Kuhn, Catherine Matias, Tabea Rebafka. Properties of the stochastic approximation EM algorithm with mini-batch sampling[J]. Statistics and Computing,2020,30(6).

[12] 徐义峰，陈春明，徐云青. 一种改进的 k-均值聚类算法 [J]. 计算机应用与软件，第 25 卷第 3 期，2008 年 03 月.

[13] 刘靖明, 韩丽川, 侯立文; 基于粒子群的 K 均值聚类算法 [J]; 系统工程理论与实践. 第 6 期，2005 年 06 月.

[14] 邬启为. 基于向量空间的文本聚类方法与实现 [D]. 北京: 北京交通大学, 2014.

[15] 周志华. 机器学习［M］. 北京: 清华大学出版社，2016.

[16] Ceppellini, r. , Siniscalco, M. & Smith, C. A. Ann. Hum. Genet[J], 1955, 20:97-115.

[17] Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the EM algorithm[J]. Journal of the Royal Statistical Society, 1977, Series B 39(1):1-38.

[18] Fraley, C. and A. E. Raftery. Model-based clustering, discriminant analysis and density estimation[J]. Journal of the American Statistical Association 97, (2002), 611-631.

[19] 邢雪丹. 伊藤过程理论及其在金融中的应用 [D] . 山东: 山东大学, 2014.

[20] 余胜威.MATLAB 优化算法案例分析与应用 [M]. 北京: 清华大学出版社, 2015.

[21] David K W Ho, Bhaskar D Rao, Antithetic dithered 1-bit massive MIMO architecture: efficient channel estimationvia parameter ex-pansion and PML [J]. IEEE Transactions on Signal Processing, 2019, 67(9): 2291-2303.

[22] 余振华.EM 算法及其加速 [D] . 江西南昌：江西师范大学，2004.

[23] 罗季.Monte Carlo EM 加速算法 [J] . 应用概率统计，2008，24(3): 312-318.

[24] Merz C J , Clair D S , Bond W E . SeMi-supervised adaptive resonance theory (SMART2)[J]. IEEEIJCNN International Joint Conference on Neural Networks - Baltimore, MD, USA, 1992, 3: 851-856.

[25] B.M. Shahshahani and D.A. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon [J]. IEEE Transactions on Geoscience and Remote Sensing, 1994, 32(5): 1087-1095.

[26] 秦悦, 丁世飞. 半监督聚类综述 [J]. 计算机科学,2019,46(09):15-21.

[27] 杜阳, 姜震, 冯路捷. 结合支持向量机与半监督 K-means 的新型学习算法 [J]. 计算机应用,2019,39(12):3462-3466.

[28] Shannon C E . A mathematical theory of communication [J]. The Bell System Technical Journal, 1948, 27(4):379-423.

[29] Edwin T. Jaynes . Information theory and statistical physics [J]. Physics Review, 1957, 106(4): 620-630.

[30] Box, G. E. P. and Cox, D. R. An analysis of transformations [J]. Journal of the Royal Statistical Society, 1964, B(26): 211-252.

[31] Rumelhart DE, Hinton GE, Williams R J. Learning representations by back-propagating errors [J]. Nature, 1986，323: 533-536.

[32] Laurens van der Maaten and Geoffrey Hintton. Visualizing data using t-SNE[J]. Journel of machine learning research, 2008, 9(86): 2579-2605

致谢

时光飞逝，我的大学生活即将进入尾声。回顾过去三年，虽然我的大学生活没有想象中的那么多姿多彩，但我始终在追求梦想的轨道上平稳前行。我始终相信兴趣是最好的老师，我热爱我这个专业，因此我会情不自禁地去钻研它，在这过程中无论过程是多么艰辛复杂，过程中收获到的足以让我喜悦。

在这儿，我十分感谢我的同学们，在专业课学习和学科竞赛项目中互相讨论，使得学习过程变得欢乐美好；感谢老师们和学长学姐们，当我遇到抉择的时候，他们总会提供我真实可靠的专业建议，让我有了更加明确的规划。同时也感谢学校学院提供给我们的学习环境和学习机会，让我在大学期间有机会去外交流学习，有机会参加各类学科竞赛。

最后更要感谢我的父母和长辈对我的无条件的付出和支持，他们总是听取我的意见，遵循我的想法，并不时提醒我指引我。