软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(9):2664−2680 [doi: 10.13328/j.cnki.jos.005265]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

单趟贝叶斯模糊聚类算法*

刘解放[1,2], 蒋亦樟[1], 王 骏[1], 邓赵红[1], 王士同[1]

[1](江南大学 数字媒体学院,江苏 无锡 214122)

[2](湖北交通职业技术学院 交通信息学院,湖北 武汉 430079)

通讯作者: 刘解放, E-mail: l.jiefang@gmail.com

摘 要: 对于概率模糊聚类,贝叶斯模糊聚类方法表现出良好的聚类性能,它从先验知识和贝叶斯理论的角度出发,采用最大后验概率理论处理模糊划分,进而获取最终的聚类结果.该方法有效地结合了概率论和模糊论两者的优点,较之传统的模糊聚类算法(如 FCM 算法),该方法能够获取全局最优解并估计聚类个数.但在大数据时代,该方法较高的时间复杂度限制了它的实用性.针对此问题,首先在贝叶斯模糊聚类中引入加权机制,提出了加权贝叶斯模糊聚类算法;然后将其与单趟聚类框架相结合,提出了面向大规模数据的快速单趟贝叶斯模糊聚类算法,并从理论上对相关性质进行了较为深入的分析.所提出的单趟贝叶斯模糊聚类新算法较之贝叶斯模糊聚类算法在时间复杂度和收敛性上均有着不同程度的性能提升,同时继承了贝叶斯模糊聚类的良好的聚类性能.最后,相关实验结果亦验证了所提方法的有效性.

关键词: 概率模糊聚类;单趟聚类;大规模数据;贝叶斯推理;时间复杂度

中图法分类号: TP181

中文引用格式: 刘解放,蒋亦樟,王骏,邓赵红,王士同.单趟贝叶斯模糊聚类算法.软件学报,2018,29(9):2664−2680. http://www.jos.org.cn/1000-9825/5265.htm

英文引用格式: Liu JF, Jiang YZ, Wang J, Deng ZH, Wang ST. Single pass Bayesian fuzzy clustering. Ruan Jian Xue Bao/ Journal of Software, 2018,29(9):2664−2680 (in Chinese). http://www.jos.org.cn/1000-9825/5265.htm

Single Pass Bayesian Fuzzy Clustering

LIU Jie-Fang[1,2], JIANG Yi-Zhang[1], WANG Jun[1], DENG Zhao-Hong[1], WANG Shi-Tong[1]

[1](School of Digital Media, Jiangnan University, Wuxi 214122, China)

[2](School of Traffic Information, Hubei Communications Technical College, Wuhan 430079, China)

Abstract: Based on the maximum a posteriori (MAP) principle and Bayesian framework, the Bayesian fuzzy clustering (BFC) method recently proposed exhibits promising characteristics in estimating the number of clusters and finding the globally optimal clustering solution, for the method effectively combines the advantages of both probability theory and fuzzy theory. However, since it suffers from its high computational burden, BFC becomes impractical for large-scale datasets. In this paper, in order to circumvent this drawback of BFC, a weighted Bayesian fuzzy clustering (WBFC) algorithm is first proposed by introducing weighting mechanism in BFC. Then, a fast single pass Bayesian fuzzy clustering (SPBFC) algorithm is developed by combining WBFC with a single pass clustering framework. Theoretical analysis on convergence and time complexity is also discussed. The experimental results show that SPBFC not only inherits the promising characteristics, but also has a fast convergence speed for large-scale datasets.

Key words: probabilistic fuzzy clustering; single pass clustering; large-scale data; Bayesian inference; time complexity

* 基金项目: 国家自然科学基金(61300151, 61572236); 江苏省自然科学基金(BK20130155, BK20160187); 江苏省杰出青年基金(BK20140001)

Foundation item: National Natural Science Foundation of China (61300151, 61572236); Natural Science Foundation of Jiangsu Province (BK20130155, BK20160187); Jiangsu Province Outstanding Youth Fund (BK20140001)

收稿时间: 2015-12-18; 修改时间: 2016-08-09, 2016-12-12; 采用时间: 2017-01-11; jos 在线出版时间: 2017-03-31

CNKI 网络优先出版: 2017-03-31 21:54:34, http://kns.cnki.net/kcms/detail/11.2560.TP.20170331.2154.003.html

在众多的数据中,挖掘有价值的信息或热点话题对当今社会发展具有非常重要的推动作用.聚类是一种典型的无监督数据分析手段,对于发现无标记数据中内在模式和重要信息尤为适用,并已被广泛应用于众多领域中,如数据挖掘[1]、计算机视觉[2]、网络安全[3]、生物信息学[4]等诸多领域[5−7].近年来,新理论和新思想不断涌现,日益丰富着该项的研究内容[8,9].

在该项研究中,研究人员针对不同的应用场景提出了各种各样的聚类方法.文献[10]对当前的聚类算法进行了总结,其认为目前聚类方法大致可分为两类:模糊聚类和概率聚类,并指出硬聚类只是模糊聚类或概率聚类的一个特例.一方面,模糊聚类方法使用模糊集理论来描述数据对象对各类簇的隶属度关系,具有代表性的工作有 FCM(fuzzy c-means)算法[11]、SPFCM(single-pass FCM)[12]和 OFCM(online FCM)[13];另一方面,概率聚类将隶属度、聚类中心和聚类数目等变量视作遵循一定分布规律的随机值,根据问题的不同特点,使用合适概率模型来估计这些变量的观测值,代表性的工作包括 Gauss mixture 模型聚类算法[14]、Bayesian 模型聚类[15]、Markov 模型聚类[16]、Dirichlet 过程混合模型聚类[17]等.尽管上述这些方法均基于不同的聚类理论,但其目的都是为了发现数据的“自然分组”[8].实验表明:在特定的场景下,这些方法都可以有效地挖掘数据的内在模式和信息.

虽然上述这些基于模糊或概率论的聚类方法取得了一定的成果,但其均只采用了一种理论(模糊或概率)而未将两大理论进行有效地的融合,达到取长补短的效果.近年来,这方面的研究有了突破性的进展,文献[18−21]分别介绍了几种将模糊聚类理论及概率聚类理论相融合所获得的性能更佳的聚类模型,在这些模型中,尤以 Glenn 等人[22]提出的贝叶斯模糊聚类(Bayesian fuzzy clustering,简称 BFC)方法最为典型和有效.该方法的核心思想是:从先验知识和贝叶斯理论的角度出发,采用最大后验概率(maximum a posteriori,简称 MAP)理论处理模糊聚类,并进一步获得聚类个数自适应学习的能力;另外,该方法还从全局最优的角度求解相关参数以获得最优值.因此,该算法在多个方面均优于以往的模糊或概率聚类算法.然而,BFC 具有较高的时间复杂度,这一缺陷使得该方法并不适用于大规模数据,其应用范围受到了极大的限制,不符合当前的实际应用需求.

受上述思想的启发,如何找出一种新的方法,既能保持 BFC 的良好聚类性能,同时又适用于大规模数据场景,正是本文的出发点.本文首先在 BFC 中引入加权机制,提出了一种加权贝叶斯模糊聚类(weighted Bayesian fuzzy clustering,简称 WBFC)方法用于判定各数据对象对聚类结果的贡献程度,从而挑选出更具代表性的聚类对象以达到浓缩样本的目的.该方法不但保留了 BFC 算法的优点,还能够自适应地学习得到各样本的重要性程度,为后续进一步提出针对大规模数据的概率模糊聚类算法奠定了基础.然后,在 WBFC 算法的基础上,本文进一步地结合单趟聚类框架提出了针对大规模数据的快速单趟贝叶斯模糊聚类算法(single pass Bayesian fuzzy clustering,简称 SPBFC).与原始的 BFC 算法相比,最终获得的 SPBFC 具有以下优点.

1) 通过概率方法来实现模糊聚类,从而实现了概率和模糊论两者的相互融合及概率聚类和模糊聚类两者等价关系的建立,所提方法拥有了两者的优点,这在以前的聚类研究中较为少见;
2) 由于所提方法的参数求解过程使用了马尔科夫链蒙特卡洛(Markov chain Monte Carlo,简称 MCMC)采样方法,而没有采用封闭式解(closed form solution),从理论上讲,该方法可以估计全局最优解;另外,该方法进一步突破经典模糊聚类模糊指数 m 必须大于 1 的约束,使其不但可以小于等于 1,甚至可为负值,这是以前模糊聚类(如 FCM)所不具有的;
3) 由于采用了单趟聚类框架,SPBFC 算法可以处理无法导入内存的大规模数据集聚类问题,并通过对象加权机制浓缩代表点和初始化加速技术,提高了聚类效率.实验结果表明:所提方法扩展了传统模糊聚类性能,并改进了聚类结果.

1 BFC 原理与方法

给定数据集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}\in R^{N\times D}$,其中,$N$ 为总的数据对象个数,D 为数据的维数;$\boldsymbol{Y}=\{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_C\}\in R^{C\times D}$ 为数据聚类中心,其中,C 为总的聚类个数,m 是模糊指数,$\boldsymbol{I}$ 是 D 维单位矩阵,u_{nc} 是第 n 个数据对象 $\boldsymbol{x}$ 在第 c 个聚类中的隶属度,其中,$\sum_{c=1}^{C}u_{nc}=1,u_{nc}>0,n=1,...,N,c=1,...,C.$

BFC 的核心思想是从概率的角度解决模糊聚类,首先定义了 BFC 概率模型,它由 3 部分组成:模糊数据似

然、模糊隶属度先验和聚类中心先验,分别定义如下.

定义 1. 模糊数据似然(FDL):

$$p(\boldsymbol{X}\mid\boldsymbol{U},\boldsymbol{Y})=\prod_{n=1}^{N}FDL(\boldsymbol{x}_n\mid\boldsymbol{u}_n,\boldsymbol{Y})=\prod_{n=1}^{N}\frac{1}{Z(\boldsymbol{u}_n,m,\boldsymbol{Y})}\prod_{c=1}^{C}\mathcal{N}(\boldsymbol{x}_n\mid\boldsymbol{\mu}=\boldsymbol{y}_c,\boldsymbol{\Lambda}=u_{nc}^{m}\boldsymbol{I}) \tag{1}$$

定义 2. 模糊隶属度先验(FCP):

$$\tilde{p}(\boldsymbol{U}\mid\boldsymbol{Y})=\prod_{n=1}^{N}FCP(\boldsymbol{u}_n\mid\boldsymbol{Y})=\prod_{n=1}^{N}Z(\boldsymbol{u}_n,m,\boldsymbol{Y})\left(\prod_{c=1}^{C}u_{nc}^{-mD/2}\right)Dirichlet(\boldsymbol{u}_n\mid\boldsymbol{\alpha}) \tag{2}$$

定义 3. 聚类中心先验:

$$p(\boldsymbol{Y})=\prod_{c=1}^{C}\mathcal{N}(\boldsymbol{y}_c\mid\boldsymbol{\mu}_y,\Sigma_y) \tag{3}$$

其中,$Z(\boldsymbol{u}_n,m,\boldsymbol{Y})$是归一化常量.然而,该归一化常量被模糊隶属度先验消去,因此它不必计算.超参$\boldsymbol{\mu}_y$ 和Σ_y 可以通过经验贝叶斯方法计算得到,如 $\boldsymbol{\mu}_y=1/N\sum_{n=1}^{N}\boldsymbol{x}_n,\Sigma_y=\gamma/N\sum_{n=1}^{N}(\boldsymbol{x}_n-\boldsymbol{\mu}_y)(\boldsymbol{x}_n-\boldsymbol{\mu}_y)^T$,其中,γ为用户预设定参数,它影响高斯分布的密度.大量实验表明,γ=3 较为合适.$Dirichlet(\boldsymbol{u}_n,\boldsymbol{\alpha})$是一个狄里克雷似然,其定义如下:

$$Dirichlet(\boldsymbol{u}_n\mid\boldsymbol{\alpha})=\frac{\Gamma(\sum_{c=1}^{C}\alpha_c)}{\prod_{c=1}^{C}\Gamma(\alpha_c)}\prod_{c=1}^{C}u_{nc}^{\alpha_c-1} \tag{4}$$

其中,$u_{nc}\geqslant 0,\sum_{c=1}^{C}u_{nc}=1,\alpha>0$. 当把它看做是一个对称狄利克雷分布时,其参数$\boldsymbol{\alpha}$=$\mathbf{1}_C$,也即是$\boldsymbol{\alpha}$为 C×1 的列向量,且每个元素都为 1.

通过把公式(1)、公式(2)和公式(3)相乘,可得数据 $\boldsymbol{X}$ 和参数 $\boldsymbol{U},\boldsymbol{Y}$ 的联合似然,如公式(5):

$$\begin{aligned}p(\boldsymbol{X},\boldsymbol{U},\boldsymbol{Y})&=p(\boldsymbol{X}\mid\boldsymbol{U},\boldsymbol{Y})\tilde{p}(\boldsymbol{U}\mid\boldsymbol{Y})p(\boldsymbol{Y})\\&\propto\exp\left\{-\frac{1}{2}\sum_{n=1}^{N}\sum_{c=1}^{C}u_{nc}^{m}\parallel\boldsymbol{x}_n-\boldsymbol{y}_c\parallel^2\right\}\times\left(\prod_{n=1}^{N}\prod_{c=1}^{C}u_{nc}^{\alpha_c-1}\right)\times\exp\left\{-\frac{1}{2}\sum_{c=1}^{C}(\boldsymbol{y}_c-\boldsymbol{\mu}_y)^T\sum_{y}^{-1}(\boldsymbol{y}_c-\boldsymbol{\mu}_y)\right\}\end{aligned} \tag{5}$$

它正比于参数的先验分布 $p(\boldsymbol{U},\boldsymbol{Y}\mid\boldsymbol{X})\propto p(\boldsymbol{X},\boldsymbol{U},\boldsymbol{Y})$.根据 MAP 理论,公式(5)联合似然的目标函数形式是它自身的负对数,为了简化,乘上因子 2,得目标函数如下:

$$J(\boldsymbol{X},\boldsymbol{U},\boldsymbol{Y})=\sum_{n=1}^{N}\sum_{c=1}^{C}u_{nc}^{m}\parallel\boldsymbol{x}_n-\boldsymbol{y}_c\parallel^2-2\sum_{n=1}^{N}\sum_{c=1}^{C}(\alpha_c-1)\log u_{nc}+\sum_{c=1}^{C}(\boldsymbol{y}_c-\boldsymbol{\mu}_y)^T\sum_{y}^{-1}(\boldsymbol{y}_c-\boldsymbol{\mu}_y) \tag{6}$$

最后,BFC 采用 MAP 推理理论,通过 Metropolis-Hastings[23]算法对隶属度和聚类中心参数进行采样迭代,求解它们的最优值.图 1 展示了 BFC 的模型及求解过程.左边两个模块是从概率角度建立 BFC 模型,虚线中的 3 个模块用于求解 BFC 模型中最优参数.

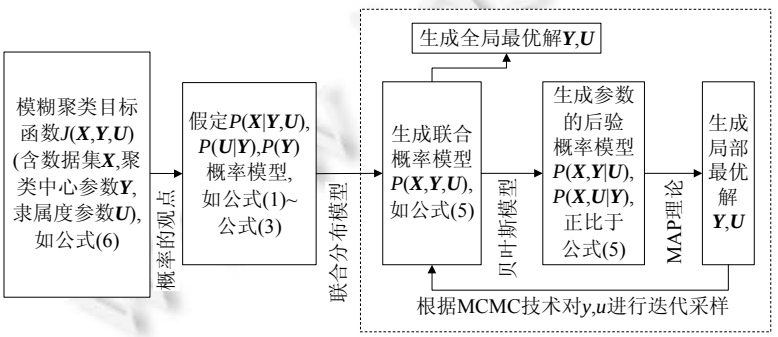


Fig.1 BFC model and procedure of estimating parameters

图 1 BFC 模型及求解过程

我们注意到:BFC 实现了概率与模糊两种方法的融合,建立了概率聚类和模糊聚类两者的等价关系,突破了传统模糊聚类模糊指数的约束,并且理论上可以得到全局最优解;然而,它的时间复杂度过高,因此无法处理大

规模数据聚类问题.

2 SPBFC 方法

对于大数据聚类,目前面临的难题是数据集太大而无法导入内存[24–26].本文采用了单趟聚类(single pass)框架[12]处理这一难题,该框架把整个大的数据集分成多个易管理的小块(chunk)进行处理,块的大小由用户决定.在这个框架中,每块中各聚类中心被标识为代表点.然后,将当前块的代表点进行加权合并到下一块中,循环直到最后一块,生成整个数据的聚类中心.为加速聚类,我们采用当前块的聚类中心作为输入参数初始化下一块聚类中心.这种知识传播加速了收敛并提高数据聚类性能.第 4.2 节及图 2 展示了 SPBFC 算法的原理及求解过程.

由于 SPBFC 方法采用了分块、加权机制,把大的数据进行分块,并为每一个数据对象引入了权重,因此,我们首先要考虑的是对拥有不同权重数据对象组成的各数据块进行聚类,因此,针对每块数据提出了加权模糊聚类(weighted Bayesian fuzzy clustering,简称 WBFC)算法.第 2.1 节给出了它的详细介绍.

2.1 WBFC算法

为了在聚类的过程中进一步判定各数据对象的聚类贡献程度,从而挑选出具备代表性的数据对象,本文在 BFC 方法中引入了对象加权机制,从而针对数据块提出了一种具备对象自适应加权的贝叶斯模糊聚类算法 WBFC.其具体目标函数定义如下:

$$p(\boldsymbol{X},\boldsymbol{U},\boldsymbol{Y})=p(\boldsymbol{X}\mid\boldsymbol{U},\boldsymbol{Y})p(\boldsymbol{U}\mid\boldsymbol{Y})p(\boldsymbol{Y})$$
$$\propto\exp\left(-\frac{1}{2}\sum_{n=1}^{N}\sum_{c=1}^{C}w_n u_{nc}^{m}\parallel\boldsymbol{x}_n-\boldsymbol{y}_c\parallel^2\right)\times\left(\prod_{n=1}^{N}\prod_{c=1}^{C}u_{nc}^{\alpha-1}\right)\times\exp\left(-\frac{1}{2}\sum_{c=1}^{C}(\boldsymbol{y}_c-\boldsymbol{\mu}_y)^T\sum\nolimits_y^{-1}(\boldsymbol{y}_c-\boldsymbol{\mu}_y)\right)\tag{7}$$

其中,$w_n>0$ 表示第 n 个对象对最终聚类划分的贡献程度值,关于它的计算方法详见第 2.2 节.

本文中,我们同样采用类似BFC算法的参数优化策略来求解WBFC算法中各模型参数.首先,根据新的目标公式(7)可得到相应的后验概率模型和接受概率模型如下:

$$p(\boldsymbol{x}_n,\boldsymbol{u}_n\mid\boldsymbol{Y})=p(\boldsymbol{x}_n\mid\boldsymbol{u}_n,\boldsymbol{Y})p(\boldsymbol{u}_n\mid\boldsymbol{Y})\propto\prod_{c=1}^{C}\exp\left\{-\frac{1}{2}w_n u_{nc}^{m}\parallel\boldsymbol{x}_n-\boldsymbol{y}_c\parallel^2\right\}u_{nc}^{\alpha_c-1}\tag{8}$$

$$R_u=\min\left\{1,\frac{p(\boldsymbol{x}_n,\boldsymbol{u}_n^{+}\mid\boldsymbol{Y})}{p(\boldsymbol{x}_n,\boldsymbol{u}_n\mid\boldsymbol{Y})}\right\}\tag{9}$$

$$p(\boldsymbol{X},\boldsymbol{y}_c\mid\boldsymbol{U})=p(\boldsymbol{X}\mid\boldsymbol{U},\boldsymbol{y}_c)p(\boldsymbol{y}_c)\propto\exp\left\{-\frac{1}{2}\sum_{n=1}^{N}w_n u_{nc}^{m}\parallel\boldsymbol{x}_n-\boldsymbol{y}_c\parallel^2\right\}\times\exp\left\{-\frac{1}{2}(\boldsymbol{y}_c-\boldsymbol{\mu}_y)^T\sum\nolimits_y^{-1}(\boldsymbol{y}_c-\boldsymbol{\mu}_y)\right\}\tag{10}$$

$$R_y=\min\left\{1,\frac{p(\boldsymbol{X},\boldsymbol{y}_c^{+}\mid\boldsymbol{U})}{p(\boldsymbol{X},\boldsymbol{y}_c\mid\boldsymbol{U})}\right\}\tag{11}$$

通过上述模型,此处进一步给出了 WBFC 算法的具体实现过程,见算法 1.

算法 1. WBFC.

输入:数据矩阵 $\boldsymbol{X}$,模糊指数 m,聚类个数 C,迭代次数 N_{iter},对象权重 w;

输出:隶属度矩阵 $\boldsymbol{U}^*$和聚类中心矩阵 $\boldsymbol{Y}^*$.

1 初始化参数$\boldsymbol{\mu}_y,\Sigma_y$

2 初始化 $\boldsymbol{u}_n$~*Dirichlet*($\boldsymbol{\alpha}=\mathbf{1}_C$),$n$=1,…,$N$

3 初始化 $\boldsymbol{y}_c$~$\mathcal{N}(\boldsymbol{\mu}_y,\Sigma_y)$,$c$=1,…,$C$

4 $\boldsymbol{u}_n^*=\boldsymbol{u}_n,\boldsymbol{y}_c^*=\boldsymbol{y}_c$, //把 MAP 样本赋给当前样本

5 for *iter*=1,…,N_{iter}

//采样 $\boldsymbol{U}$~$p(\boldsymbol{U}|\boldsymbol{X},\boldsymbol{Y})\propto p(\boldsymbol{X},\boldsymbol{U},\boldsymbol{Y})$

6 for n=1,…,N

7 采用公式(4)生成新的建议隶属度样本 $\boldsymbol{u}_n^{+}$

```
8       采用公式(9),以概率 R_u 接受 u_n = u_n^+
9       if  p(x_n, u_n^+ | Y^*) > p(x_n, u_n^* | Y^*)   //采用公式(8)
10          u_n^* = u_n^+
11      end if
12    end for
      //采样 Y~p(Y|X,U)∝p(X,U,Y)
13    for c=1,…,C
14      采用N(y_c, Σ_y/δ)生成新的建议聚类中心样本 y_c^+
15      采用式(11),以概率 R_y 接受 y_c = y_c^+
16      if  p(X, y_c^+ | U^*) > p(X, y_c^* | U^*)   //采用式(10)
17          y_c^* = y_c^+
18      end if
19    end for
      //检查整个样本的最大似然
20    if p(X,U,Y)>p(X,U^*,Y^*)   //采用公式(7)
21      U^*=U, Y^*=Y
22    end if
23 end if
```

- 第 1 步~第 4 步根据经验贝叶斯方法和参数建议分布初始化 MAP 样本$\{\boldsymbol{U}^*,\boldsymbol{Y}^*\}$;
- 第 6 步~第 12 步求解隶属度参数,其中:第 7 步根据公式(4)狄利克雷建议分布生成新建议样本$\boldsymbol{u}_n^+$;第 8 步根据公式(9)的概率接受$\boldsymbol{u}_n=\boldsymbol{u}_n^+$;第 9 步~第 11 步根据公式(8)比较每一个建议隶属度向量$\boldsymbol{u}_n^+$,如果$\boldsymbol{u}_n^+$增大了当前 MAP 样本的似然,它将成为新的$\boldsymbol{u}_n^*$;
- 第 13 步~第 19 步求解聚类中心参数,其中:第 14 步根据高斯建议分布$\mathcal{N}(\boldsymbol{y}_c,\Sigma_y/\delta)$生成新建议样本$\boldsymbol{y}_c^+$,其中,$\delta$是用户预设定参数,用来控制建议聚类中心的紧度,它以当前马尔科夫链状态为中心,它的大小关系到样本的接受率,在应用中,我们设定δ=10;第 15 步根据公式(11)的概率接受$\boldsymbol{y}_c=\boldsymbol{y}_c^+$;第 16 步~第 18 步根据公式(10)比较每一个建议聚类中心向量$\boldsymbol{y}_c^+$,如果$\boldsymbol{y}_c^+$增大了当前 MAP 样本的似然,它将成为新的$\boldsymbol{y}_c^*$;
- 第 20 步~第 22 步根据公式(7)对比当前概率样本$\{\boldsymbol{U},\boldsymbol{Y}\}$和当前 MAP 样本$\{\boldsymbol{U}^*,\boldsymbol{Y}^*\}$的似然,如果概率样本具有更高的似然,保留作为新的 MAP 样本.我们把以概率接受的样本与 MAP 样本的比较看作是“加速搜索”,它为样本的改进提供了更多可能性.

2.2 SPBFC算法

加权贝叶斯模糊聚 WBFC 算法的核心思想是:在 BFC 方法的基础上引入了对象权重,进而在聚类的同时可进一步选出更具聚类特性的数据对象即代表点.基于 WBFC 算法的聚类特性,本节将进一步在该算法的基础上提出一种具备大规模数据聚类分析能力的 SPBFC 算法.该算法的核心思想是:把大规模数据分块并对各块所包含的聚类对象(包括代表点)进行加权计算,其中,权重分为两种情况:一种是原数据块的数据对象权重,我们设置为 1;另一种是代表点(即来自上一块的聚类中心)的权重,可通过下列公式计算:

$$w_c'=\sum_{n=1}^{N_l+K} w_n u_{nc}, c=1,...,C \tag{12}$$

其中,w_c'是第 c 个代表点(即类中心)的权重,N_l是第 l 块的对象个数,C 是数据的聚类个数,u_{nc}是第 n 对象属于第 c 个聚类的隶属,w_n是第 n 个对象的权重.对于第 1 块数据(l=1),每个对象的权重 w_n=1,K=0.对于第 2 块数据(l≠1),

$K=C$,C 个加权中心结合第 l 块的数据,共 N_l+C 个数据对象通过 WBFC 进行聚类,其中,N_l 个对象的权重为 1,C 个聚类中心的权重基于上一块数据计算得到.继续重复上述步骤,直到最后一块.算法 2 给出了 SPBFC 算法的具体实现过程.

算法 2. SPBFC 算法.

输入:数据矩阵 $\boldsymbol{X}$,模糊指数 m,聚类个数 C,迭代次数 N_{iter},对象权重 $\boldsymbol{w}$,数据块个数 d;

输出:隶属度矩阵 $\boldsymbol{U}^*$和聚类中心矩阵 $\boldsymbol{Y}^*$.

1 随机划分 $\boldsymbol{X}$ 为 d 块 $\boldsymbol{X}=\{\boldsymbol{X}_1,\ldots,\boldsymbol{X}_d\}$ //每块有 N_l 对象,$1\leqslant l\leqslant d$

2 $\boldsymbol{w}=\boldsymbol{1}_{N_l}$

3 $\boldsymbol{U}^*,\boldsymbol{Y}^*=WBFC(\boldsymbol{X}_1,C,m,N_{iter},\boldsymbol{w})$ //采用算法 1

for l=2 to d

4 采用公式(12)计算当前代表点的权重 w'_c,c=1,…,C

5 $\boldsymbol{w}=\{\boldsymbol{1}_{N_l}\cup\boldsymbol{w}'\}$

6 $\boldsymbol{U}^*,\boldsymbol{Y}^*=WBFC(\{\boldsymbol{X}_l\cup\boldsymbol{Y}^*\},C,m,N_{iter},\boldsymbol{w},\boldsymbol{Y}^*)$ //采用算法 1

end if

- 第 1 步把 $\boldsymbol{X}$ 随机分成 d 块,并导入内存;
- 第 2 步为第 1 块的每个数据对象权重赋值为 1,每块的数据对象个数为 N_l,因此整块对象权重向量为:
$$\boldsymbol{w}=\boldsymbol{1}_{N_l};$$
- 第 3 步采用 WBFC 算法对第 1 块数据进行聚类,返回 C 个聚类中心点 $\boldsymbol{Y}^*$;然后,SPBFC 在余下块上进行迭代,每次迭代,WBFC 将处理不同的数据块,该数据块由上次的聚类中心 $\boldsymbol{Y}^*$和当前数据块 $\boldsymbol{X}_l$ 组成 $\{\boldsymbol{X}_l\cup\boldsymbol{Y}^*\}$,因此,每次迭代要聚类$(N_l+C)$个数据对象;
- 第 4 步和第 5 步计算$\{\boldsymbol{X}_l\cup\boldsymbol{Y}^*\}$中各对象权重:第 4 步用公式(12)计算当前各类的加权隶属度和,可得 C 个聚类中心 $\boldsymbol{Y}^*$的权重;第 5 步为下次聚类创建(N_l+C)维权重向量,该向量由 N_l 个对象的权重向量 $\boldsymbol{1}_{N_l}$ 和 C 个聚类中心 $\boldsymbol{Y}^*$的权重向量组成;
- 第 6 步表明:WBFC 的输入参数包括数据对象$\{\boldsymbol{X}_l\cup\boldsymbol{Y}^*\}$、聚类个数 C、模糊指数 m、迭代次数 N_{iter}、权重 $\boldsymbol{w}$ 和初始化聚类中心 $\boldsymbol{Y}^*$(用于加速收敛).

3 SPBFC 相关讨论

3.1 收敛性分析

文献[27−29]证明了:随着迭代次数的增加,MCMC 方法能保证收敛到全局最优解.并且,文献[22]同样表明,BFC 采用 MCMC 方法后可以保证收敛到全局最优值.尽管 SPBFC 算法也采用了 MCMC 方法,但是我们引入了数据对象权重,所以必须证明该算法在加权对象上收敛.显然,权重可能是整数,也可能是分数或无理数,因为它是由计算机产生,所以它存在有限精度.下面我们对相应情况分别给出定理和证明.

定理 1. 权重为整数时,SPBFC 算法收敛.

证明:因为 SPBFC 算法的执行过程是由 WBFC 算法在随机划分的各块上依次循环执行所构成,且 SPBFC 算法最终聚类结果也是由 WBFC 算法(在最后一块联合上一块代表点聚类)求解得到,所以要证明 SPBFC 收敛,只需证明 WBFC 算法在数据块上收敛.因为 WBFC 算法是在 BFC 算法中引入了加权机制,所以只需证明该算法在加权对象上收敛.当对象权重是整数时,这等同于该对象有多个拷贝.文献[29]给出的最初收敛性证明仅要求对象存在有限个拷贝,因此原始的相关证明依然有效.也即权重是整数时,WBFC 算法收敛,因此,SPBFC 算法收敛. □

定理 2. 权重为分数时,SPBFC 算法收敛.

证明:当权重是分数时,通过乘上所有权重的最小公分母,它可转化为整数,因此由定理 1,定理 2 成立. □

引理 1[30]. 在两个不同的无理数 a,b 之间,至少存在一个有理数.

定理 3. 权重为无理数时,SPBFC 算法收敛.

证明:首先定义一个无理数作为有理数收敛序列的极限[30].从文献[30,31]得出:对于任何无理数,必定存在一个无限接近于它的有理数.根据引理 1,假设 R 是一个有理数且位于两个无理数 IR 和 $IR+\varepsilon$之间,ε为无穷小量,因此我们有:对于$\forall IR,\forall \varepsilon>0,\exists R: IR<R<IR+\varepsilon$.因此有:对于$\forall IR,\forall \varepsilon>0,\exists R: (R-IR)<\varepsilon$.因为当所有的权重都是有理数,我们能找到一个最小公分母,然后乘上它,可转换为整数权重.所以对于权重是无理数的情况,我们选择一个无限接近的有理数替代它.此时,算法的解会无限接近于真实值,因为这里仅通过无穷小量修改了有限个权重,这对最终结果的影响可忽略不计.根据定理 1 和定理 2,权重为有理数时,SPBFC 收敛,因此定理 3 成立. □

根据定理 1~定理 3,可知所提算法是收敛的.第 4 节的实验也验证了它收敛性.但是,收敛速度及结果是不可预测的,它依赖于数据的结构和所寻参数个数及模型和建议分布中参数设置.关于更多 MCMC 方法收敛速度的相关讨论,还可见文献[32].

3.2 时间复杂度分析

WBFC 算法的时间复杂度包括 2 部分,分别为搜索 MAP 样本 $\boldsymbol{U}^*$和 $\boldsymbol{Y}^*$所耗时间.因此,单次迭代时间复杂度为 $O(NCD+CD^2)$,其中,D^2 由公式(3)中的协方差矩阵产生.实际应用中,一般为对角协方差矩阵,因此,时间复杂度降为 $O(NCD)$.若迭代 N_{iter} 次,总时间复杂度为 $O(NCDN_{iter})$.然而,SPBFC 算法把数据 $\boldsymbol{X}$ 分成 d 块 $\boldsymbol{X}=\{\boldsymbol{X}_1,\ldots,\boldsymbol{X}_d\}$,并采用加速的方法,极大降低了迭代次数,此时,时间复杂度为 $O((NCDN_{iter}/d)+(NCDN'_{iter}/d)(d-1))$.

由于 $N_{iter} >> N'_{iter}$,因此,SPBFC 的时间复杂度降为 $O(NCDN_{iter}/d)$.

3.3 参数设置

SPBFC 算法有下列参数必须在运行前设置.

模糊指数 m:参数 m 等同于经典 FCM 算法中模糊指数的作用,因此 m 接近于 1,隶属度变得更硬.然而,因为所提算法没有使用封闭解形式,所以可自由设置模糊指数 m 为 1,获得硬聚类;甚至可以小于 1 或负值.在大多数实验中,我们设置 m 为 1.2(为获得相对硬的结果)或者为 2(为获得相对模糊的结果).尽管模糊指数的设置依赖于特定数据集及应用,然而我们大量的实验表明:对于所提聚类方法,m=1.2 是较合适的选择;

聚类个数 C:聚类个数是 SPBFC 算法中关键参数,正如其他大部分聚类算法一样,用户必须预先设定.该参数也依赖于特定数据集及应用;

Dirichlet 先验参数$\boldsymbol{\alpha}$:参数$\boldsymbol{\alpha}$控制 Dirichlet 分布形状,应该大于 0.当$\boldsymbol{\alpha}$小于 1 时,隶属度值趋向于二值(0 或 1);当$\boldsymbol{\alpha}$大于 1 时,隶属度更模糊.随着$\boldsymbol{\alpha}$的增加,Dirichlet 分布的方差以隶属度平均值为中心递减.实验中,隶属度分布是未知的,因此$\boldsymbol{\alpha}$应为 1,生成均匀的隶属度分布.

4 实验与分析

为验证所提算法的有效性,我们采用了合成数据集、UCI 和图像数据集,进行了 5 种类型的实验.

1) 比较 SPBFC 算法与标准 FCM 算法,观察 SPBFC 是否扩展了 FCM[11]性能及改进了结果;
2) 比较 SPBFC 算法与两种经典单趟模糊聚类算法 SPFCM[12]和 OFCM[13],观察本文算法否提高了聚类性能;
3) 应用 SPBFC 算法到大型图像分割,观察它的真实应用性能;
4) 数据块的不同比例划分实验,观察块的划分比例对 SPBFC 运行时间的影响;
5) 赋予参数 m,$\boldsymbol{\alpha}$不同的数值,观察 SPBFC 算法对参数的敏感性.

实验平台:Intel i7-4770,四核 CPU,8GB 内存,Windows 7 操作系统,算法采用 Matlab2010a 编写.

4.1 评价标准

聚类结果采用 4 种常用评价标准:准确度(accuracy)[26]、归一化互信息(normalized mutual information,简称 NMI)[33]、芮氏指数(rand index,简称 RI)[34]和加速因子(speedup).所有结果都由算法随机初始化并独立运行 10

次求平均值得到.

1) 准确度

$$Accuracy=\sum_{c=1}^{C}\frac{N_c^p}{N} \tag{13}$$

2) 归一化互信息

$$NMI=\frac{\sum_{c=1}^{C}\sum_{p=1}^{C}N_c^p\log\left(\frac{N\cdot N_c^p}{N_c\cdot N_p}\right)}{\sqrt{\left(\sum_{c=1}^{C}N_c\log\left(\frac{N_c}{N}\right)\right)\left(\sum_{p=1}^{C}N_p\log\left(\frac{N_p}{N}\right)\right)}} \tag{14}$$

公式(13)、公式(14)中,N 表示数据集中数据对象个数,N_c 是在第 c 个聚类的对象个数,N_p 是第 p 类真实标签的对象个数, N_c^p 是第 c 个聚类的对象和第 p 类真实标签的对象匹配个数.

3) 芮氏指数

$$RI=\frac{N_{00}+N_{11}}{N(N-1)/2} \tag{15}$$

其中,N_{00} 表示具有不同类标签且属于不同聚类的对象配对个数,N_{11} 表示具有相同类标签且属于相同聚类的对象配对个数,N 表示整个数据集对象个数.

以上 3 种评价标准取值范围为[0,1],结果越大越好,当接近 1 时,说明聚类结果与真实标签完全一致.

4) 加速因子

该标准代表聚类算法的实际运行时间比较.加速因子被定义为 t_{full}/t_{chunk},其中,t_{full} 为整个数据集上的运行时间,t_{chunk} 为各块上的运行时间总和.尽管我们的实验是一次性加载全部数据,但我们认为,加速因子对于真实不能加载的数据是相似的,因为相同数据量被读入内存,不论它是逐块处理或一次性处理.

4.2 数据集

实验采用了 2 个合成数据集、4 个 UCI 数据集和 4 幅图像数据集.

- 合成数据集

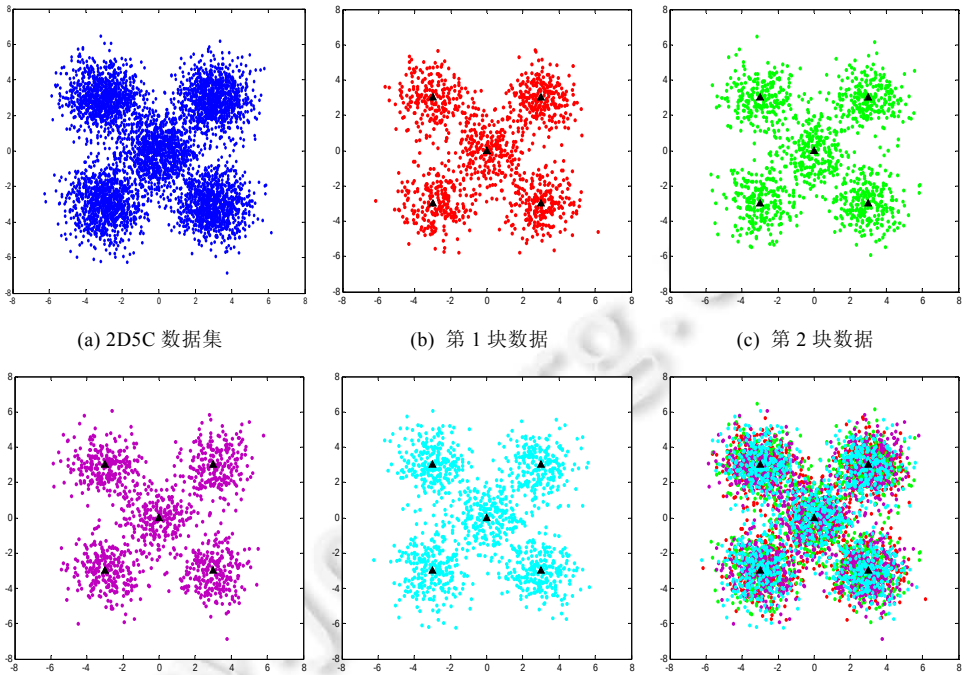
第 1 个大规模合成数据由 6 400 个 2 维数据点分成 5 类组成.它用来测试 SPBFC 算法及展示它的执行过程.它的分布图如图 2(a)所示.数据被随机分为 4 块,每块各类选择一个聚类中心为代表点.算法的过程如图 2(b)~图 2(f)所示.第 1 块的数据分布如图 2(b)所示,黑色三角表示每个聚类中心.可以发现,所选择的中心恰好在每个聚类的中心区域.然后,同样的过程分别对其他 3 块进行聚类,如图 2(c)~图 2(e)所示.后一块和上一块的聚类中心进行合并聚类,依次循环,最终得到整个数据的聚类中心,如图 2(f)所示.可以发现,最终的聚类中心位于数据集的理想位置.第 4.4 节给出了聚类结果.

第 2 个合成数据集由两个高斯生成器 $\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{I})$和 $\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{I})$产生,其中,$\boldsymbol{\mu}_1=(1,1)^T$,$\boldsymbol{\mu}_2=(4,4)^T$,分为 2 类,每类包含 250 数据点,如图 3 所示,它用来验证 SPBFC 算法提供了扩展性能.为便于叙述,我们命名第 1 个和第 2 个合成数据集分别为 2D5C 和 2D2C.

- UCI 数据集

Skin 由 24 5057 个样本组成,分为 2 类,每个样本是一个 3 维特征向量[8].Brainweb 由 100 000 样本组成,分为 2 类,每个样本是一个 3 维特征向量[8].ISOLET6 是 ISOLET 的一个子集,包含 6 类 1 440 个样本,其中 7 个属性被随机选取[12].Pen Digits 包含 10 类 3 498 样本,其中 6 个属性被随机选取[12].

- 4 幅图像:Plane,Tank,Clock and Hawk,如图 5 所示.



(a) 2D5C 数据集 (b) 第 1 块数据 (c) 第 2 块数据

(d) 第 3 块数据 (e) 第 4 块数据 (f) 最终聚类结果

Fig.2 2D5C dataset and processing procedures of SPBFC

图 2 2D5C 数据集及 SPBFC 的处理过程

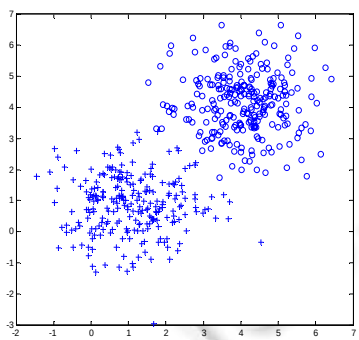


Fig.3 2D2C dataset ('+' generated by $\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{I})$, '∘' generated by $\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{I})$)

图 3 2D2C 数据集('+'由$\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{I})$生成,'∘'由$\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{I})$生成)

4.3 SPBFC与FCM的对比

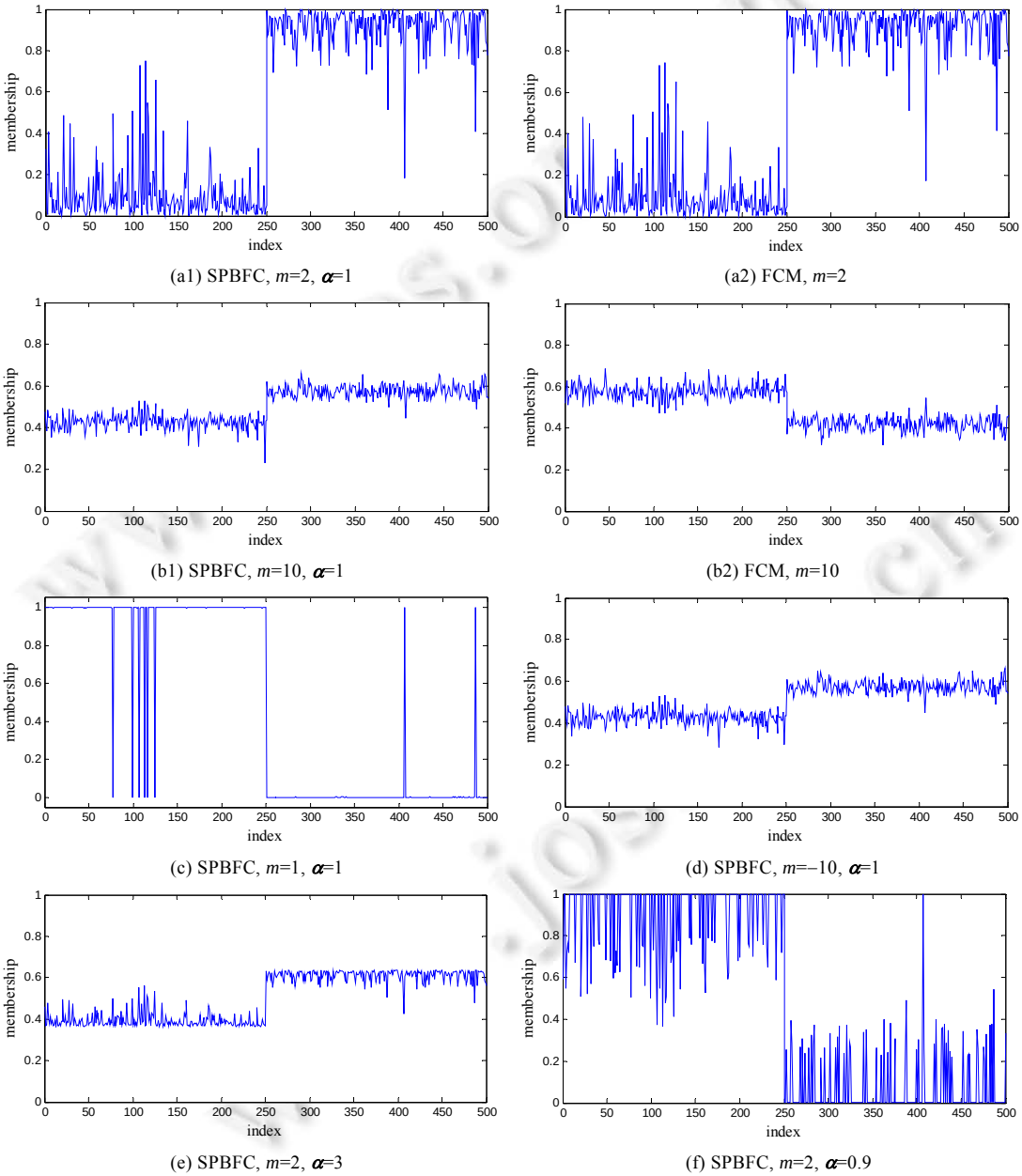
在这个实验中,FCM 运行直到收敛,而 SPBFC 运行 100 次的采样迭代.我们展示了两个算法在整个 2D2C 数据集上的运行结果,该结果给出了 SPBFC 和 FCM 在不同参数数值下的直观行为.我们也证明了:由于 SPBFC 没有使用封闭解形式进行迭代,因此它可以运行在一些 FCM 无法执行的 m 参数之上.

图 4(a)、图 4(b)分别展示了 SPBFC 和 FCM 模糊指数 m 分别为 2 和 10 的结果.结果表明:SPBFC 在相对较少的采样迭代下,可以找到与 FCM 相似的结果.

图 4(c)、图 4(d)展示了$\boldsymbol{\alpha}$=1 时,SPBFC 分别置模糊指数 m 为 1 和10 的输出结果.但是对于这些模糊指数值,FCM 是无效的,所以无法展示其结果.对于 m=1 时,SPBFC 产生一个类似 k-means 的硬聚类划分;当 m=–10 时,

隶属较模糊.

图 4(e)、图 4(f)分别展示了 SPBFC 通过改变参数α可以实现一些 FCM 无法完成的行为.图 4(e)展示了 m=2 和α=3 时 SPBFC 的聚类结果,结果显示,隶属度的值总是围绕 0.5 在[0.4,0.6]范围内.然而,当 m=2,α=0.9 时,图 4(f)显示:距中心点近的隶属度较硬,而远离中心点的隶属度较模糊.这些实验结果也符合文献[22]的运行结果.需要说明的是:当 SPBFC 在整个数据集上聚类时,所有的对象权重都为 1,此时 SPBC 等价于 WBFC,并且退化为 BFC.



(a1) SPBFC, m=2, α=1 (a2) FCM, m=2

(b1) SPBFC, m=10, α=1 (b2) FCM, m=10

(c) SPBFC, m=1, α=1 (d) SPBFC, m=−10, α=1

(e) SPBFC, m=2, α=3 (f) SPBFC, m=2, α=0.9

Fig.4 Comparition of SPBFC and FCM outputs for different m, α on 2D2C dataset
(indices 1~250 generated by $\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{I})$, indices 251~500 generated by $\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{I})$)

图 4 SPBFC 和 FCM 的结果对比(1~250 点由$\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{I})$生成,251~500 点由$\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{I})$生成)

表 1 列出了 SPBFC 和 FCM 相应参数的聚类结果.

Table 1 Results of SPBFC and FCM with different m, α on 2D2C dataset

表 1 SPBFC 和 FCM 对于不同 m,α参数的结果

Algorithms	SPBFC	FCM	SPBFC	FCM
Parameters	m=2,α=1	m=2	m=10,α=1	m=10
Accuracy	0.986 0	0.986 0	0.988 0	0.986 0
NMI	0.895 6	0.895 6	0.907 2	0.893 9
RI	0.972 3	0.972 3	0.976 2	0.972 3

从图 4 和表 1 可以发现:SPBFC 可以获得与 FCM 相当的结果,甚至有些结果要好于 FCM.

4.4 2D5C,Brainweb和Skin数据集上的实验结果

为了使 SPFCM[12]和 OFCM[13]算法较好的运行,根据文献[24]的建议,我们设定模糊指数 m=1.7.对于 2D5C, Brainweb 和 Skin 数据集,聚类个数 C 分别设置为 5,2 和 2.整个数据集被随机划分成大小相等的块,每个块的大小由用户决定.通常情况下,它指整个数据集大小的一定比例.最后一块的大小可能比其他要小,如果整个数据集不能被块的个数整除.对于 2D5C 数据集,我们选择了整个数据集的 1%,2.5%,5%,10%,25%和 50%分别进行了实验.对于 Brainweb 和 Skin 数据集,由于内存的限制,我们选择了更小的百分比,分别为整个数据集的 0.1%, 0.25%,0.5%,1%,2.5%和 5%.为了观察每种方法的稳健性,我们把每种方法在随机初始化的基础上独立运行 10 次.然后,计算出 Accuracy,NMI 和 RI 的平均值、标准方差和最小及最大值.平均值反映了聚类算法的平均性能,标准方差、最小值和最大值反映聚类算法的稳健性.3 个数据集的结果见表 2~表 4.结果显示:对于数据不同比例的分块,SPBFC 每次总是产生最好的划分,并且 SPBFC 总是具有最低的标准方差.这表明 SPBFC 更稳健.

Table 2 Accuracy of OFCM, SPFCM and SPBFC

表 2 OFCM,SPFCM 和 SPBFC 的准确度

(a) 2D5C			
	Algorithm		
	OFCM	SPFCM	SPBFC
Chunk size (%)	avg./std./min./max.	avg./std./min./max.	avg./std./min./max.
1	0.9684/0.0031/0.9641/0.9725	0.9707/0.0028/0.9669/0.9745	**0.9870**/0.0027/0.9810/0.9909
2.5	0.9697/0.0028/0.9656/0.9739	0.9697/0.00200.9675/0.9728	**0.9884**/0.0019/0.9848/0.9906
5	0.9695/0.0044/0.9628/0.9784	0.9701/0.0026/0.9661/0.9739	**0.9887**/0.0022/0.9825/0.9950
10	0.9681/0.0040/0.9616/0.9742	0.9693/0.0031/0.9648/0.9750	**0.9895**/0.0030/0.9846/0.9949
25	0.9701/0.0033/0.9658/0.9756	0.9698/0.0036/0.9616/0.9744	**0.9897**/0.0032/0.9821/0.9947
50	0.9719/0.0033/0.9675/0.9769	0.9697/0.0032/0.9623/0.9730	**0.9901**/0.0031/0.9846/0.9967
(b) Brainweb			
	Algorithm		
	OFCM	SPFCM	SPBFC
Chunk size (%)	avg./std./min./max.	avg./std./min./max.	avg./std./min./max.
0.1	0.9302/0.0015/0.9279/0.9320	0.9303/0.0017/0.9279/0.9336	**0.9504**/0.0003/0.9499/0.9511
0.25	0.9304/0.0011/0.9283/0.9321	0.9305/0.0018/0.9279/0.9332	**0.9508**/0.0004/0.9498/0.9513
0.5	0.9307/0.0016/0.9274/0.9328	0.9308/0.0014/0.9292/0.9332	**0.9509**/0.0007/0.9497/0.9520
1	0.9304/0.0014/0.9285/0.9332	0.9302/0.0015/0.9270/0.9322	**0.9510**/0.0009/0.9490/0.9521
2.5	0.9310/0.0015/0.9289/0.9331	0.9302/0.0017/0.9273/0.9338	**0.9511**/0.0009/0.9491/0.9524
5	0.9312/0.0010/0.9297/0.9331	0.9311/0.0016/0.9280/0.9331	**0.9514**/0.0009/0.9500/0.9529
(c) Skin			
	Algorithm		
	OFCM	SPFCM	SPBFC
Chunk size (%)	avg./std./min./max.	avg./std./min./max.	avg./std./min./max.
0.1	0.4458/0.0041/0.4385/0.4494	0.4523/0.0043/0.4439/0.4573	**0.5562**/0.0037/0.5511/0.5701
0.25	0.4519/0.0040/0.4448/0.4565/	0.4521/0.0043/0.4444/0.4580	**0.5570**/0.0040/0.5495/0.5679
0.5	0.4523/0.0048/0.4430/0.4580	0.4516/0.0030/0.4469/0.4567	**0.5570**/0.0033/0.5473/0.5641
1	0.4518/0.00370.4442/0.4573	0.4491/0.0050/0.4400/0.4593	**0.5572**/0.0037/0.5491/0.5621
2.5	0.4500/0.0052/0.4427/0.4590	0.4490/0.0046/0.4419/0.4560	**0.5573**/0.0046/0.5497/0.5635
5	0.4500/0.0053/0.4408/0.4595	0.4499/0.0049/0.4419/0.4563	**0.5577**/0.0043/0.5490/0.5668

Table 3 NMI of OFCM, SPFCM and SPBFC

表 3 OFCM,SPFCM 和 SPBFC 的 NMI

(a) 2D5C			
	Algorithm		
	OFCM	SPFCM	SPBFC
Chunk size (%)	avg./std./min./max.	avg./std./min./max.	avg./std./min./max.
1	0.9374/0.0066/0.9270/0.9570	0.9409/0.0055/0.9326/0.9488	**0.9549**/0.0049/0.9407/0.9631
2.5	0.9396/0.0044/0.9328/0.9569	0.9398/0.0060/0.9317/0.9480	**0.9575**/0.0035/0.9515/0.9622
5	0.9390/0.0093/0.9247/0.9679	0.9408/0.0046/0.9327/0.9473	**0.9579**/0.0039/0.9428/0.9687
10	0.9360/0.0081/0.9228/0.9587	0.9385/0.0056/0.9317/0.9494	**0.9590**/0.0055/0.9487/0.9715
25	0.9397/0.0073/0.9228/0.9587	0.9395/0.0071/0.9293/0.9512	**0.9598**/0.0050/0.9503/0.9698
50	0.9438/0.0067/0.9349/0.9644	0.9402/0.0062/0.9239/0.9478	**0.9601**/0.0045/0.9491/0.9740
(b) Brainweb			
	Algorithm		
	OFCM	SPFCM	SPBFC
Chunk size (%)	avg./std./min./max.	avg./std./min./max.	avg./std./min./max.
0.1	0.6858/0.0063/0.6768/0.6935	0.6869/0.0074/0.6760/0.7008	**0.7174**/0.0029/0.7110/0.7211
0.25	0.6870/0.0045/0.6789/0.6941	0.6877/0.0076/0.6754/0.6989	**0.7182**/0.0030/0.7119/0.7221
0.5	0.6861/0.0064/0.6725/0.6961	0.6890/0.0069/0.6765/0.6991	**0.7191**/0.0030/0.7142/0.7238
1	0.6871/0.0065/0.6786/0.6995	0.6884/0.0061/0.6725/0.6987	**0.7187**/0.0021/0.7155/0.7212
2.5	0.6894/0.0062/0.6711/0.6989	0.6862/0.0074/0.6736/0.7024	**0.7196**/0.0012/0.7170/0.7210
5	0.6883/0.0068/0.6746/0.6979	0.6904/0.0044/0.6843/0.6984	**0.7223**/0.0012/0.7210/0.7247
(c) Skin			
	Algorithm		
	OFCM	SPFCM	SPBFC
Chunk size (%)	avg./std./min./max.	avg./std./min./max.	avg./std./min./max.
0.1	0.1244/0.0026/0.0893/0.2117	0.1260/0.0029/0.0627/0.2011	**0.2298**/0.0025/0.1965/0.2699
0.25	0.1256/0.0030/0.0818/0.1702	0.1296/0.0033/0.0874/0.1732	**0.2366**/0.0030/0.1860/0.2894
0.5	0.1434/0.0038/0.0842/0.2181	0.1385/0.0031/0.1037/0.1673	**0.2437**/0.0030/0.1935/0.2916
1	0.1270/0.0034/0.0761/0.1784	0.1231/0.0029/0.0715/0.1617	**0.2396**/0.0036/0.1661/0.2842
2.5	0.1459/0.0029/0.1014/0.1955	0.1168/0.0026/0.0660/0.1410	**0.2528**/0.0024/0.1832/0.3136
5	0.1343/0.0026/0.0988/0.1751	0.1226/0.0025/0.0859/0.1540	**0.2556**/0.0023/0.2212/0.2984

Table 4 RI of OFCM, SPFCM, SPBFC

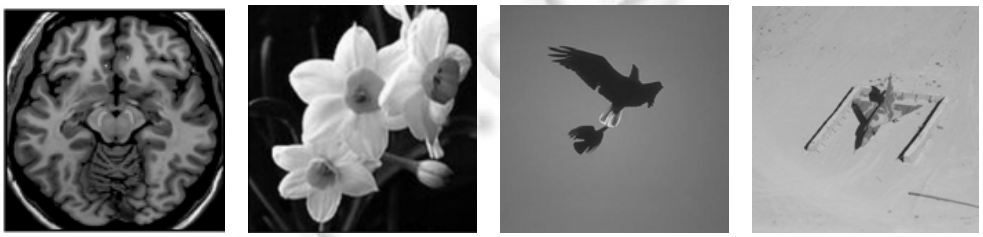
表 4 OFCM,SPFCM,SPBFC 的 RI

(a) 2D5C			
	Algorithm		
	OFCM	SPFCM	SPBFC
Chunk size (%)	avg./std./min./max.	avg./std./min./max.	avg./std./min./max.
1	0.9693/0.0029/0.9653/0.9731	0.9715/0.0026/0.9678/0.9750	**0.9880**/0.0024/0.9824/0.9916
2.5	0.9705/0.0026/0.9667/0.9744	0.9705/0.0019/0.9685/0.9734	**0.9893**/0.0017/0.9860/0.9914
5	0.9703/0.0042/0.9641/0.9787	0.9709/0.0024/0.9672 /0.9743	**0.9896**/0.0020/0.9838/0.9954
10	0.9690/0.0037/0.9629/0.9747	0.9701/0.0028/0.9660/0.9755	**0.9903**/0.0024/0.9858/0.9953
25	0.9708/0.0031/0.9670/0.9760	0.9706/0.0034/0.9630/0.9748	**0.9905**/0.0030/0.9836/0.9952
50	0.9725/0.0031/0.9685/0.9772	0.9705/0.0030/0.9637/0.9735	**0.9909**/0.0030/0.9858/0.9971
(b) Brainweb			
	Algorithm		
	OFCM	SPFCM	SPBFC
Chunk size (%)	avg./std./min./max.	avg./std./min./max.	avg./std./min./max.
0.1	0.8760/0.0027/0.8719/0.8793	0.8764/0.0031/0.8719/0.9022	**0.9066**/0.0006/0.9048/0.9071
0.25	0.8764/0.0020/0.8726/0.8795	0.8766/0.0032/0.8719/0.9015	**0.9067**/0.0013/0.9045/0.9085
0.5	0.8764/0.0025/0.8730/0.8815	0.8768/0.0026/0.8733/0.8988	**0.9071**/0.0017/0.9043/0.9115
1	0.8765/0.0008/0.8748/0.8774	0.8759/0.0031/0.8709/0.9026	**0.9071**/0.0016/0.9034/0.9093
2.5	0.8769/0.0029/0.8710/0.8808	0.8759/0.0026/0.8703/0.8997	**0.9075**/0.0017/0.9051/0.9101
5	0.8776/0.0028/0.8721/0.8813	0.8775/0.0027/0.8737/0.9013	**0.9078**/0.0018/0.9052/0.9113
(c) Skin			
	Algorithm		
	OFCM	SPFCM	SPBFC
Chunk size (%)	avg./std./min./max.	avg./std./min./max.	avg./std./min./max.
0.1	0.4042/0.0008/0.4029/0.4048	0.4055/0.0009/0.4038/0.4065	**0.5102**/0.0007/0.5092/0.5134
0.25	0.4054/0.0008/0.4040/0.4063	0.4054/0.0009/0.4039/0.4067	**0.5104**/0.0007/0.5089/0.5128
0.5	0.4048/0.0010/0.4032/0.4070	0.4048/0.0009/0.4035/0.4062	**0.5104**/0.0008/0.5085/0.5119
1	0.4055/0.0010/0.4036/0.4067	0.4053/0.0010/0.4039/0.4065	**0.5104**/0.0009/0.5089/0.5117
2.5	0.4053/0.0009/0.4043/0.4064	0.4050/0.0011/0.4036/0.4069	**0.5104**/0.0008/0.5088/0.5114
5	0.4050/0.0010/0.4035/0.4063	0.4050/0.0011/0.4033/0.4070	**0.5105**/0.0009/0.5088/0.5125

表 2(a)~表 2(c)分别显示了 2D5C,Brainweb 和 Skin 数据集的准确度.结果表明:当块较小时,SPFCM 的性能好于 OFCM;随着块大小的增大,当达到一定比例时,OFCM 的性能好于 SPFCM.SPBFC 表现的得更好,无论准确度或稳健性.对于 3 个数据集各自所有划分比例的平均准确度,与 OFCM 相比,SPBFC 分别提高了 2.1%,4.6%和 23.7%.与 SPFCM 相比,SPBFC 分别提高了 1.9%,4.5%,及 23.6%.如表 3 和表 4 所示,NMI 和 RI 的结果显示出类似特征.

4.5 图像分割

4 幅图像用于验证 SPBFC 算法对于大型图像分割的有效性,它们分别是 MRI(128×128 像素),Flower(501×501 像素),Hawk(256×256 像素)和 Airplane(521×521 像素).限于版面,所有的图像都缩小为 128×128 像素,如图 5 所示.

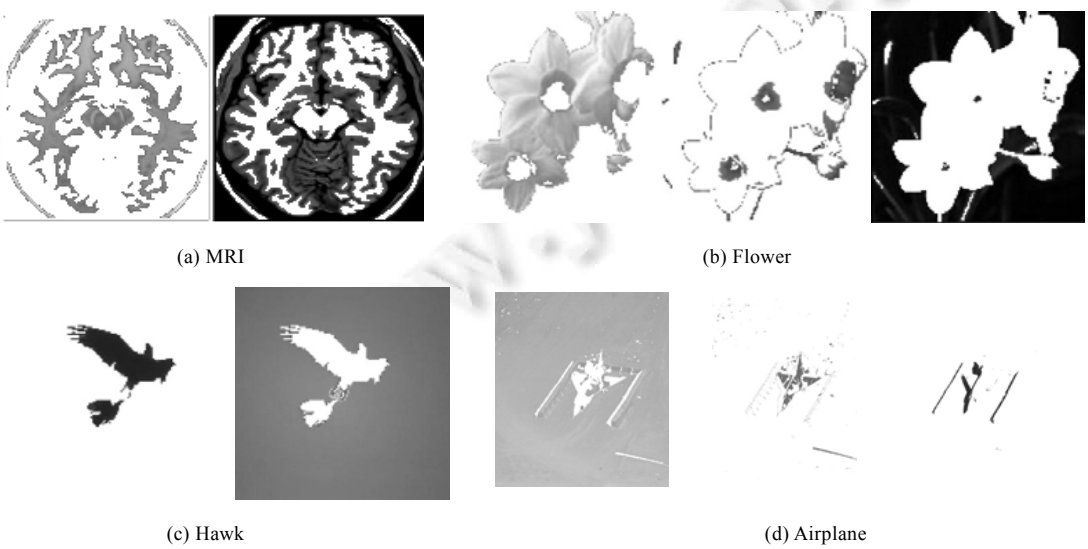


(a) MRI (b) Flower (c) Hawk (d) Airplane

Fig.5 Images used in segmentation experiments

图 5 用于图像分割的四幅图像

为了观察 SPBFC 分割图像的性能,实验中,我们把 SPBFC 和最新贝叶斯模糊集聚类 BFC 算法[11]进行了比较研究.根据文献[22],我们把两种算法的 m 和 α 参数都设置为 1.实验中,我们以 5%的比例分块各个图像,并设置 4 幅图像的聚类个数分别为 2,3,2 和 3.图 6 和图 7 分别给出了 SPBFC 和 BFC 对 4 幅图像分割结果,表 5 给出 2 个算法的运行时间.实验结果表明:BFC 可以获得一个好的划分效果,但是运行时间过长,无法应用于大规模数据集;而 SPBFC 算法能够在相对较短的时间内得到更好的分割结果.

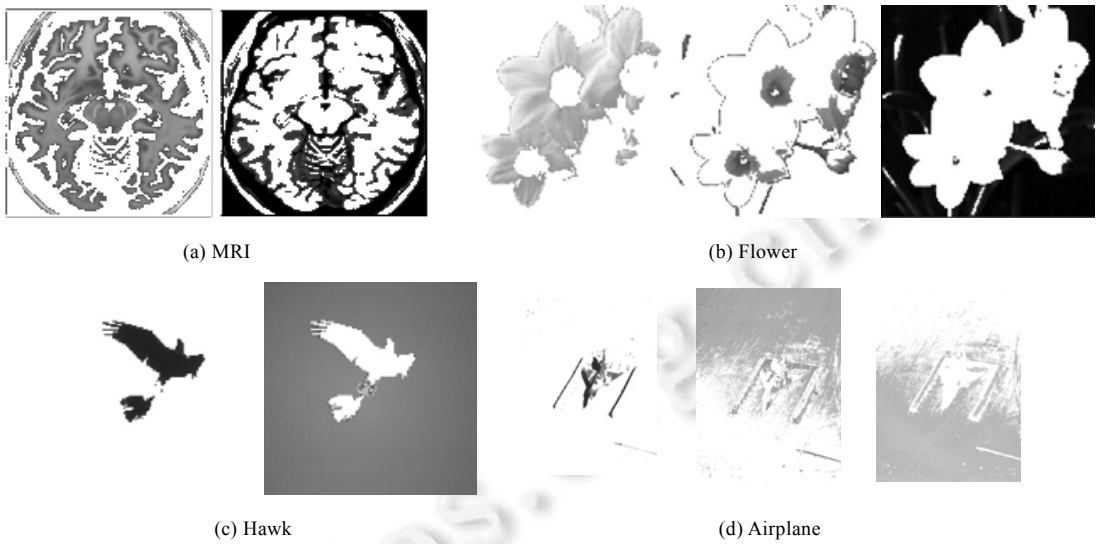


(a) MRI (b) Flower

(c) Hawk (d) Airplane

Fig.6 Segmentation results of SPBFC on four images

图 6 SPBFC 在 4 幅图上的分割结果



(a) MRI (b) Flower

(c) Hawk (d) Airplane

Fig.7 Segmentation results of BFC on four images

图 7 BFC 在 4 幅图上的分割结果

Table 5 Comparison about running time on 4 images (s)

表 5 4 幅图像的运时间对比 (秒)

Image	SPBFC	BFC
MRI	27.83	657.26
Flower	288.97	12 495.82
Hawk	139.48	3 292.87
Airplane	303.42	13 120.61

4.6 分块比例对SPBFC的影响

该实验中,我们展示了块的划分比例对 SPBFC 加速的影响.实验表明:即使我们有足够的内存加载数据,SPBFC 处理小块数据比整个数据集会有显著的加速,且输出几乎同等质量的聚类效果.

表 6 展示了 SPBFC 在数据集上不同划分比例的运行时间及相比整个数据集的平均加速.尽管 SPBFC 主要面向大规模数据,我们也展示了其在 Isolet6 和 Pen Digits 小规模数据集上的加速.表 6 表明:聚类以 1%划分的数据集要快于以 10%划分的数据集;大规模数据集 2D5C,Brainweb 和 Skin 数据集实现了良好的加速.在小规模数据集,也取得了显著的加速.结果表明:SPBFC 可用于加快聚类,即使数据可以完全加载到内存.

Table 6 Speed-up of SPBFC compared with itself on the entire datasets

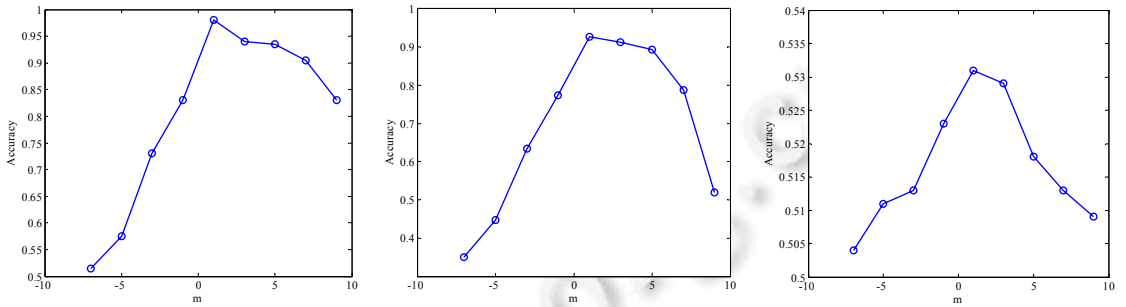
表 6 SPBFC 在不同比例分块上的运行时间及加速

Datasets	Chunk sizes				
	100% (s)	10% (s)	1% (s)	speed-up (10%)	speed-up (1%)
Isolet6	75.79	23.61	17.50	3.21	4.33
Pen Digits	188.12	37.25	20.74	5.05	9.07
2D5C	308.64	35.68	8.01	8.65	38.55
Brainweb	4388.81	549.49	115.74	8.09	37.92
Skin	12454.78	1329.22	283.64	9.37	43.91

4.7 参数敏感性分析

为了验证模糊指数 m 和 Dirichlet 先验参数α对 SPBFC 性能的影响,我们设置了α=1 时,m 为分别为{−7,−5,−3,−1,1,3,5,7,9}和 m=1.2 时,α分别为{0.1,0.3,0.5,0.7,0.9,1,3,5,7,9}的运行参数.图 8 和图 9 显示,SPBFC 在 2D5C,Brainweb 和 Skin 数据集上以 5%比例分块的聚类准确度.在这里,我们只展示准确度的结果,*NMI* 和 *RI* 具有相似

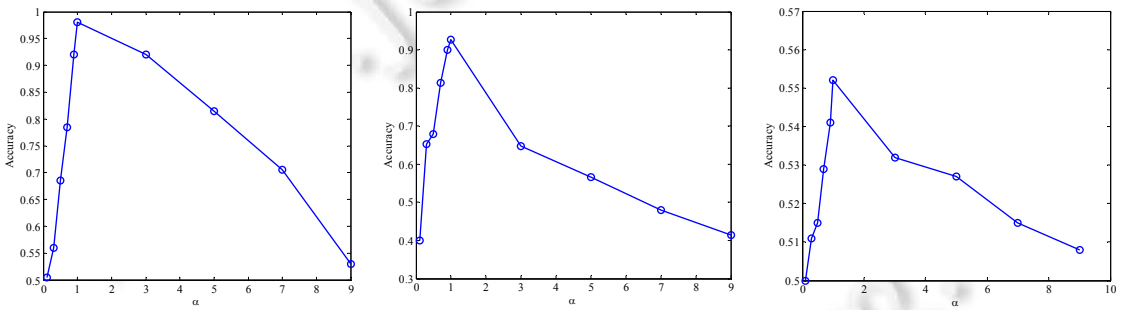
的趋势特征.如图 8 和图 9 所示,两个参数对 SPBFC 性能的影响较大.当 m 和α接近 1 时,SPBFC 达到最佳性能.一般来说,非常小或大的值,将导致 SPBFC 性能退化.因此在实践中,我们建议两个参数在[0.7,1.5]之间微调,或者两者都直接赋值为 1.



2D5C Brainweb Skin

Fig.8 Accuracy of SPBFC with various value of parameters m

图 8 SPBFC 对不同参数 m 的准确度



2D5C Brainweb Skin

Fig.9 Accuracy of SPBFC with various value of parameters α

图 9 SPBFC 对于不同参数α的准确度

5 结 论

由于模糊技术和概率技术的盛行,模糊聚类和概率聚类已被广泛研究和应用,许多算法和模型已被提出.一些学者也建立了它们之间的等价关系,甚至把它们融合于一个系统之中.这些融合后的概率模糊聚类表现出突出的优越性能,但是这些方法存在一个共同的缺点,就是无法处理大规模数据聚类问题.本文从概率的观点研究并实现模糊聚类,继而弥补了这一缺点.本文首先通过引入数据对象和各参数的先验分布及加权机制,提出了一个加权概率模糊聚类计算模型 WBFC 及其贝叶斯推理算法,它主要针对每块聚类中心代表点进行加权聚类;其次,在此基础上,我们通过引入单趟聚类框架,提出一种面向大规模数据的快速单趟概率模糊聚类 SPBFC 算法,它主要通过分块和初始化机制加速了概率模糊聚类的收敛速度.实验结果表明,该算法具有以下优点.

1) 融合了概率与模糊方法到模糊聚类中,并建立概率聚类与模糊聚类的等价关系;
2) 突破了传统模糊聚类 m 必须大于 1 的限制,并且理论上可以保证收敛到全局最优解;
3) 可以更有效地聚类大规模数据集.

References:

[1] Jiang YZ, Chung FL, Wang ST. Enhanced fuzzy partitions vs data randomness in FCM. Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, 2014,27(4):1639–1648. [doi: 10.3233/IFS-141130]

[2] Wu JZ, Chen FL, Hu DW. A detection-based person tracking algorithm. Journal of National University of Defense Technology, 2014,36(2):113−117 (in Chinese with English abstract). [doi: 10.11887/j.cn.201402019]

[3] Leung K, Leckie C. Unsupervised anomaly detection in network intrusion detection using clusters. In: Proc. of the 28th Australasian Conf. on Computer Science, Vol.38. Australian Computer Society, Inc., 2005. 333−342.

[4] Menon N, Ramakrishnan R. Brain tumor segmentation in MRI images using unsupervised artificial bee colony algorithm and FCM clustering. In: Proc. of the 2015 Int'l Conf. on Communications and Signal Processing. IEEE, 2015. 6−9. [doi: 10.1109/iccsp.2015.7322635]

[5] Pei X, Lyu Z, Chen C. Manifold adaptive label propagation for face clustering. IEEE Trans. on Cybernetics, 2015,45(8):1681−1691. [doi: 10.1109/tcyb.2014.2358592]

[6] Zhang C, Jin CQ, Zhou AY. Clustering algorithm over uncertain data streams. Ruan Jian Xue Bao/Journal of Software, 2010,21(9): 2173−2182 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3654.htm [doi: 10.3724/SP.J.1001.2010.03654]

[7] Lin KP, Pai PF, Lu YM, Chang PT. Revenue forecasting using a least-squares support vector regression model in a fuzzy environment. Information Sciences, 2013,220:196−209. [doi: 10.1016/j.ins.2011.09.003]

[8] Ying WH, Chung FL, Wang ST. Scaling up synchronization-inspired partitioning clustering. IEEE Trans. on Knowledge and Data Engineering, 2014,26(8):2045−2057. [doi: 10.1109/tkde.2013.178]

[9] Jiang YZ, Chung FL, Wang ST, Deng ZH. Collaborative fuzzy clustering from multiple weighted views. IEEE Trans. on Cybernetics, 2015,45(4):688−701.

[10] Wang ST, Chung FL. Note on the relationship between probabilistic and fuzzy clustering. Soft Computing, 2004,8(7):523−526. [doi: 10.1007/s00500-003-0302-2]

[11] Bezdek JC, Ehrlich R, Full W. FCM: The fuzzy *c*-means clustering algorithm. Computers & Geosciences, 1984,10(2):191−203. [doi: 10.1016/0098-3004(84)90020-7]

[12] Hore P, Hall L, Goldgof D. Single pass fuzzy c means. In: Proc. of the IEEE Int'l Fuzzy System Conf. 2007. 1−7. [doi: 10.1109/fuzzy.2007.4295372]

[13] Hore P, Hall L, Goldgof D, Cheng W. Online fuzzy c means. In: Proc. of the Annual Meeting of the North American on Fuzzy Information Processing Society. 2008. 1−5. [doi: 10.1109/nafips.2008.4531233]

[14] Ozonat KM. Gauss mixture model clustering for noisy images under rate constraints. In: Proc. of the 39th Asilomar Conf. on Signals, Systems and Computers. IEEE, 2005. 1226−1229. [doi: 10.1109/acssc.2005.1599957]

[15] Ramoni M, Sebastiani P, Cohen P. Bayesian clustering by dynamics. Machine Learning, 2002,47(1):91−121.

[16] Szilágyi L, Szilágyi SM. Fast implementations of Markov clustering for protein sequence grouping. In: Proc. of the Modeling Decisions for Artificial Intelligence. Berlin, Heidelberg: Springer-verlag, 2013. 214−225. [doi: 10.1007/978-3-642-41550-0_19]

[17] Campbell T, Liu M, Kulis B, How JP, Carin L. Dynamic clustering via asymptotics of the dependent Dirichlet process mixture. In: Proc. of the Advances in Neural Information Processing Systems. 2013. 449−457.

[18] Theis FJ. Bayesian fuzzy clustering of colored graphs. In: Proc. of the 10th Int'l Conf. on Latent Variable Analysis and Signal Separation. Springer-Verlag, 2012. 528−535. [doi: 10.1007/978-3-642-28551-6_65]

[19] Feng H, Giles DEA. Bayesian fuzzy regression analysis and model selection: Theory and evidence [Ph.D. Thesis]. Department of Economics, University of Victoria, 2007.

[20] Marttinen P, Tang J, Baets B, Dawyndt P, Corander J. Bayesian clustering of fuzzy feature vectors using a quasi-likelihood approach. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2009,31(1):74−85. [doi: 10.1109/tpami.2008.53]

[21] Honda K, Ichihashi H. Regularized linear fuzzy clustering and probabilistic PCA mixture models. IEEE Trans. on Fuzzy Systems, 2005,13(4):508−516. [doi: 10.1109/tfuzz.2004.840104]

[22] Glenn TC, Zare A, Gader PD. Bayesian fuzzy clustering. IEEE Trans. on Fuzzy Systems, 2015,23(5):1545−1561. [doi: 10.1109/tfuzz.2014.2370676]

[23] Robert C, Casella G. Monte Carlo Statistical Methods. 2nd ed., Springer-Verlag, 2005.

[24] Havens TC, Bezdek JC, Leckie C. Fuzzy c-means algorithms for very large dataset. IEEE Trans. on Fuzzy Systems, 2012,20(6): 1130−1146. [doi: 10.1109/tfuzz.2012.2201485]

[25] Wang ST, Wang J, Chung FL. Kernel density estimation, kernel methods, and fast learning in large data sets. IEEE Trans. on Cybernetics, 2014,44(1):1−20. [doi: 10.1109/tsmcb.2012.2236828]

[26] Mei JP, Chen L. A fuzzy approach for multitype relational data clustering. IEEE Trans. on Fuzzy Systems, 2012,20(2):358−371. [do: 10.1109/tfuzz.2011.2174366]

[27] Plummer M, Best N, Cowles K, Vines K. CODA: Convergence diagnosis and output analysis for MCMC. R News, 2006,6(1):7−11.

[28] Zhu CJ. On the convergences of a posterior density determined by MCMC samplers. Journal of Math, 2002,22(3):348−348 (in Chinese with English abstract). [doi: 10.3969/j.issn.0255-7797.2002.03.019]

[29] Roberts GO, Smith AFM. Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. Stochastic Processes and Their Applications, 1994,49(2):207−216. [doi: 10.1016/0304-4149(94)90134-1]

[30] Conway JH, Guy RK. The Book of Numbers. Springer Science & Business Media, 2012.

[31] Dunham W. The Mathematical Universe: An Alphabetical Journey through the Great Proofs, Problems, and Personalities. New York: Wiley, 1994. 1. [doi: 10.5860/choice.32-5705]

[32] Rosenthal JS. Minorization conditions and convergence rates for Markov chain Monte Carlo. Journal of the American Statistical Association, 1995,90(430):558−566. [doi: 10.2307/2291067]

[33] Strehl A, Ghosh J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research, 2003,3:583−617. [doi: 10.1162/153244303321897735]

[34] Rand WM. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 1971, 66(336):846−850. [doi: 10.2307/2284239]

附中文参考文献:

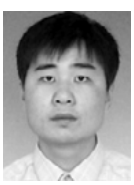
[2] 吴建宅,陈芳林,胡德文.基于检测的人体跟踪算法.国防科技大学学报,2014,36(2):113−117. [doi: 10.11887/j.cn.201402019]

[6] 张晨,金澈清,周傲英.一种不确定数据流聚类算法.软件学报,2010,21(9):2173−2182. http://www.jos.org.cn/1000-9825/3654.htm [doi: 10.3724/SP.J.1001.2010.03654]

[28] 朱崇军.MCMC 样本确定的后验密度的收敛性.数学杂志,2002,22(3):348−348. [doi: 10.3969/j.issn.0255-7797.2002.03.019]

刘解放(1982－),男,河南周口人,博士生,CCF 专业会员,主要研究领域为模式识别,数据挖掘.

蒋亦樟(1988－),男,博士,副教授,CCF 专业会员,主要研究领域为模式识别,系统建模.



王骏(1978－),男,博士,副教授,CCF 高级会员主要研究领域为智能计算,数据挖掘.

邓赵红(1981－),男,博士,教授,CCF 高级会员,主要研究领域为智能计算,系统建模.

王士同(1964－),男,教授,博士生导师,CCF 专业会员,主要研究领域为模式识别,人工智能.