软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(11):2610−2627 [doi: 10.3724/SP.J.1001.2013.04469]

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

一种基于数据流的软子空间聚类算法*

朱 林[1,2], 雷景生[1], 毕忠勤[1], 杨 杰[2]

[1](上海电力学院 计算机科学与技术学院,上海 200090)

[2](上海交通大学 图像处理与模式识别研究所,上海 200240)

通讯作者: 朱林, E-mail: cslinzhu@gmail.com, http://www.shiep.edu.cn/

摘 要: 针对高维数据的聚类研究表明,样本在不同数据簇往往与某些特定的数据特征子集相对应.因此,子空间聚类技术越来越受到关注.然而,现有的软子空间聚类算法都是基于批处理技术的聚类算法,不能很好地应用于高维数据流或大规模数据的聚类研究中.为此,利用模糊可扩展聚类框架,与熵加权软子空间聚类算法相结合,提出了一种有效的熵加权流数据软子空间聚类算法——EWSSC(entropy-weighting streaming subspace clustering).该算法不仅保留了传统软子空间聚类算法的特性,而且利用了模糊可扩展聚类策略,将软子空间聚类算法应用于流数据的聚类分析中.实验结果表明,EWSSC 算法对于高维数据流可以得到与批处理软子空间聚类方法近似一致的实验结果.

关键词: 子空间聚类;数据流聚类;可扩展聚类;模糊聚类;文本聚类

中图法分类号: TP181 **文献标识码**: A

中文引用格式: 朱林,雷景生,毕忠勤,杨杰.一种基于数据流的软子空间聚类算法.软件学报,2013,24(11):2610−2627. http://www.jos.org.cn/1000-9825/4469.htm

英文引用格式: Zhu L, Lei JS, Bi ZQ, Yang J. Soft subspace clustering algorithm for streaming data. Ruan Jian Xue Bao/Journal of Software, 2013,24(11):2610−2627 (in Chinese). http://www.jos.org.cn/1000-9825/4469.htm

Soft Subspace Clustering Algorithm for Streaming Data

ZHU Lin[1,2], LEI Jing-Sheng[1], BI Zhong-Qin[1], YANG Jie[2]

[1](School of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China)

[2](Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai 200240, China)

Corresponding author: ZHU Lin, E-mail: cslinzhu@gmail.com, http://www.shiep.edu.cn/

Abstract: A key challenge to most conventional clustering algorithms in handling many real life problems is that data points in different clusters are often correlated with different subsets of features. To address this problem, subspace clustering has attracted increasing attention in recent years. However, the existing subspace clustering methods cannot be effectively applied to large-scale high dimensional data and data streams. In this study, the scalable clustering technique to subspace clustering is extend to form soft subspace clustering for streaming data. An entropy-weighting streaming subspace clustering algorithm, EWSSC is proposed. This method leverages on the effectiveness of fuzzy scalable clustering method for streaming data by revealing the important local subspace characteristics of high dimensional data. Substantial experimental results on both artificial and real-world datasets demonstrate that EWSSC is generally effective in clustering high dimensional streaming data.

Key words: subspace clustering; data stream clustering; scalable clustering; fuzzy clustering; document clustering

聚类分析作为一种重要的数据处理技术,是目前机器学习和人工智能领域的研究热点,被广泛应用于数据挖掘、模式识别、计算机视觉、信息检索和生物信息学的研究中.聚类算法的目的是将一组未知类标的数据样

* 基金项目: 国家自然科学基金(61273258, 61272437, 61073189); 上海市自然科学基金(13ZR1417500); 上海市教育委员会科研创新项目(14YZ131)

收稿时间: 2013-03-18; 修改时间: 2013-07-12, 2013-08-02; 定稿时间: 2013-08-27

本进行划分,希望找到数据集中隐藏的潜在结构;并按照某种相似性度量,使得具有相似性质的数据归于同一类,不相似性质的数据尽可能地分开[1,2].目前,基于不同的理论和针对不同的应用提出了多种各具特色的聚类算法,如基于划分的方法、基于层次的方法、基于网格的方法和基于密度的方法等[3,4].随着信息技术的快速发展,人们在现实生活中接触到越来越多的高维数据.最新研究结果表明,对于文本和基因等高维数据,样本在不同数据簇中往往会与某些特定的数据特征子集相对应[5−7].因此,研究人员希望将数据的原始特征空间分割为不同的特征子集,从不同特征子空间的角度考察各个数据簇划分的意义.子空间聚类就成为目前高维数据聚类分析的研究热点[8−12].

子空间聚类实际上是将传统的特征选择技术与聚类算法相结合,在对数据集进行聚类划分的过程中,得到各个数据簇对应的特征子集或特征权重,从而为各个数据簇找到其对应的特征子空间[5−7].子空间聚类技术可以从数据集的不同子空间中发现相应的数据簇,也从不同数据簇中发现其对应的子空间.根据现有的研究结果,子空间聚类算法可以分为硬子空间聚类和软子空间聚类两种形式[7,13].其中,硬子空间聚类主要是指对于各个数据簇,从全部特征集合中选取某些特征子集组成其相应子空间;软子空间聚类又称为特征加权聚类,是指在聚类过程中对数据簇的各个特征赋予一个特征加权系数,在聚类过程中得到不同数据簇对应数据特征的重要性.具体而言,根据搜索方式的不同,硬子空间聚类算法可分为自底向上的子空间搜索方法和自顶向下的子空间搜索方法[6,7];对于软子空间聚类算法而言,根据特征加权不确定性表示方式的不同,可以分为模糊加权软子空间聚类(fuzzy weighting subspace clustering,简称 FWSC)[11,14]和熵加权软子空间聚类(entropy weighting subspace clustering,简称 EWSC)[12]两种.文献[9,12]指出,EWSC 对于文本和基因等高维数据都可以获得更好的聚类结果.但我们也观察到,现有的软子空间聚类算法都是基于批处理技术的聚类方法,在实际应用中,人们往往会在不同的时刻以数据流的形式得到部分数据子块;同时,受到内存大小的限制,大规模数据有时也会出现不能全部载入内存的情况[15−17],因此,针对高维数据流或者大规模数据,需要提出有效的基于数据流的软子空间聚类算法.

在过去十多年中,人们在现实生活接触到越来越多的数据流和大规模数据.因此,针对流数据的聚类方法研究成为目前数据挖掘和机器学习领域的重要课题[15−21].利用可扩展聚类框架,将大规模数据或者数据流分割成多个数据子块,分别连续地对各个数据子集进行处理,是一种非常有效的流数据聚类处理技术.Bradley 等人最早提出了可扩展聚类算法 ScaleKM[20].进一步地,Farnstrom 等人在 ScaleKM 算法的基础上提出了一种针对大规模数据简化版本的 ScaleKM 算法,同样获得了良好的聚类结果[21].Hall 等人利用模糊隶属度函数,提出了两种“软划分”的可扩展聚类算法:单次遍历模糊聚类算法 SFCM 和在线模糊聚类算法 OFCM[17−19].受到上述思想的启发,本文利用模糊可扩展聚类策略,与熵加权软子空间聚类算法 EWSC 相结合,提出了一种基于数据流的软子空间聚类算法,即熵加权流数据软子空间聚类算法 EWSSC(entropy-weighting streaming subspace clustering).

本文所做工作的意义在于:

1) EWSSC 算法不仅保留了传统软子空间聚类算法的特性,而且利用模糊可扩展聚类策略,将软子空间聚类算法应用于流数据的聚类分析中.
2) 通过在数据流聚类过程中对后续到达的数据子块采用随机采样的策略,本文进一步提出了 EWSSC-fast 算法.EWSSC-fast 算法利用数据采样技术,减少了部分数据子块包含的冗余样本,有效地降低了 EWSSC 算法的时间复杂度.
3) 本文在人造数据集和真实数据集上分别对 EWSSC 算法进行了聚类结果的测试,并对熵加权指数以及模糊权重指数的取值进行了相应的参数选择的分析,同时还对数据子块 S 大小的选择问题进行了分析和讨论.
4) 实验结果表明:EWSSC 算法对于高维数据流可以得到与传统批处理软子空间聚类方法近似一致的实验结果;并且 EWSSC-fast 算法利用数据采样技术,降低了 EWSSC 算法的运行时间,能够得到与 EWSSC 算法几乎一致的聚类划分结果.

本文第 1 节对软子空间聚类算法以及模糊可扩展聚类算法的相关研究工作进行介绍.第 2 节给出具体的熵加权流数据软子空间聚类算法 EWSSC.第 3 节对提出的 EWSSC 算法进行实验比较和分析.第 4 节对全文进行

总结,并对未来的工作进行展望.

1 相关研究

1.1 软子空间聚类算法研究

软子空间聚类算法是指在聚类过程中,对每个数据簇的各个数据特征赋予相应的特征加权系数,在迭代过程中得到各个特征在对应数据簇中的重要性.与传统的硬子空间聚类方法相比,软子空间聚类算法对数据集的处理具有更好的适应性与灵活性.因此,软子空间聚类算法受到人们越来越多的关注[8−14,22].

具体而言,对于给定的包含 N 个样本的 D 维数据集 $X=\{x_1,x_2,\ldots,x_N\}\subset R^D$,$x_{jk}$ 表示第 j 个样本第 k 维的数值.人们希望利用软子空间聚类算法得到 C 个聚类中心 $V=\{v_i,1\leqslant i\leqslant C\}$和整个数据集的模糊隶属度矩阵 $U=\{u_{ij}|1\leqslant i\leqslant C,1\leqslant j\leqslant N\}$.定义 v_{ik} 表示第 i 个聚类中心第 k 维的数值,u_{ij} 表示第 j 个样本 x_j 属于第 i 个聚类中心 v_i 的模糊隶属度.同时,为了更好地发现各个数据簇相应的子空间结构,软子空间聚类算法对每个数据簇的各个特征赋予一个特征加权系数 w_{ik}.由此,定义 w_{ik} 表示第 k 个特征对于第 i 个数据簇的重要性,则 W 表示整个数据集的特征加权系数矩阵 $W=\{w_{ik}|1\leqslant i\leqslant C,1\leqslant k\leqslant D\}$[9,11,12].根据现有的研究结果,软子空间聚类算法按照其特征加权系数不确定性表示方式的不同,可以分为模糊加权软子空间聚类方法,如 AWA[23],FWKM[13],FWSC[11,14],以及熵加权软子空间聚类算法,如 EWSC[12],LAC[22],ESSC[9]等.

1.1.1 模糊加权软子空间聚类算法

通过引入模糊权重指数 m 和模糊加权指数 τ,定义一般化的模糊加权软子空间聚类算法(FWSC)[11]的目标函数:

$$\begin{cases} J_{\mathrm{FWSC}}=\sum_{j=1}^{N}\sum_{i=1}^{C}u_{ij}^{m}\sum_{k=1}^{D}w_{ik}^{\tau}(x_{jk}-v_{ik})^2 \\ \text{s.t. } 0\leqslant u_{ij}\leqslant 1,\sum_{i=1}^{C}u_{ij}=1;0\leqslant w_{ik}\leqslant 1,\sum_{k=1}^{D}w_{ik}=1 \end{cases} \tag{1}$$

利用 Lagrange 乘子优化方法最小化公式(1),得到 FWSC 算法模糊隶属度 u_{ij}、聚类中心 v_{ik} 和特征加权系数 w_{ik} 的迭代公式.FWSC 算法的具体细节可以参考文献[11].

1.1.2 熵加权软子空间聚类算法

同样地,Jing 等人将信息熵引入软子空间聚类方法中,利用熵表示第 k 个数据特征对于第 i 个数据簇的不确定程度,提出了熵加权软子空间聚类算法(EWSC)[12].通过引入熵加权指数 γ,EWSC 的目标函数可以表示成

$$\begin{cases} J_{\mathrm{EWSC}}=\sum_{j=1}^{N}\sum_{i=1}^{C}u_{ij}^{m}\sum_{k=1}^{D}w_{ik}(x_{jk}-v_{ik})^2+\gamma\sum_{i=1}^{C}\sum_{k=1}^{D}w_{ik}\log w_{ik} \\ \text{s.t. } 0\leqslant u_{ij}\leqslant 1,\sum_{i=1}^{C}u_{ij}=1;0\leqslant w_{ik}\leqslant 1,\sum_{k=1}^{D}w_{ik}=1 \end{cases} \tag{2}$$

利用 Lagrange 乘子优化方法最小化公式(2),得到 EWSC 算法模糊隶属度 u_{ij}、聚类中心 v_{ik} 和特征加权系数 w_{ik} 的迭代公式,EWSC 算法的具体细节可以参考文献[12].

文献[9,12]指出,与其他子空间聚类算法相比,熵加权软子空间聚类算法 EWSC 对于文本和基因等高维数据均可以得到更好的聚类结果.但观察已有的软子空间聚类算法可以发现,FWSC 和 EWSC 只有在获得全部数据样本到当前聚类中心的模糊隶属度以后,才迭代更新得到新的聚类中心和特征加权系数.而在实际应用中,人们往往会在不同的时刻以数据流的形式得到部分数据子块.为此,在过去十多年中,针对数据流或者大规模数据的聚类算法研究成为当前数据挖掘和机器学习领域的研究热点[16−21,24−26].一般而言,现有数据流聚类方法可以分为两大类[16]:第 1 类将传统批处理聚类算法与增量学习(incremental learning)或在线学习(online learning)策略相结合,提出增量聚类(incremental clustering)或者在线聚类(online clustering)算法[24,25];第 2 类利用可扩展聚类方法框架(scalable clustering framework),将大规模数据或数据流分割成若干个数据子块,分别连续地对各个数据子集进行处理[16−21].本文考虑利用可扩展聚类方法的策略解决高维数据流的聚类问题.

1.2 模糊可扩展聚类算法研究

可扩展聚类方法(scalable clustering)的基本思想是:将数据流或者大规模数据分割成若干个数据子块,分别连续地对各个数据子集进行处理.可扩展聚类方法的关键技术在于,如何更好地保持历史数据的统计特征,同时利用已有的统计信息对新的数据样本做更准确的预测.目前,基于不同的可扩展学习策略和各种不同的聚类方法,大量新颖的可扩展聚类算法被提出来[16,20,21,26].

Bradley 等人首先提出了一种针对大规模数据的可扩展聚类算法 ScaleKM[20].ScaleKM 在聚类过程中,对新到的数据子块样本进行分类,将重要的数据样本保留在内存中;对普通的数据样本进行压缩,记录其充分统计特征,如总和、方差等;同时,剔除不重要的数据样本.尽管 ScaleKM 算法可以很好地应用于大规模数据的聚类分析,但在算法的聚类过程中需要多次运用复杂的数据压缩技术,使得 ScaleKM 算法的运行速度整体上要慢于 *K*-Means 算法.进一步地,Farnstrom 等人提出了一种简单的数据压缩策略,即简化版本的 ScaleKM 方法,或称为单次遍历 *K*-Means 算法(simple single pass *K*-means)[21].新方法与传统的 ScaleKM 算法相比具有更快的运行速度,同时能够获得与 ScaleKM 算法近乎一致的聚类结果.在后续的研究中,可扩展聚类框架也被应用于处理数据流的聚类问题.Agrawal 等人最早提出了一种针对大规模演化数据的数据流聚类算法 CluStream[26].CluStream 将聚类过程分成两个部分:在线模块,主要负责从历史数据中收集详细的统计信息;离线模块,主要负责将收集到的统计信息进行分析,给出对于数据流各个数据簇结构的理解.CluStream 算法就利用这两个模块实现数据流聚类过程的迭代学习.进一步地,Zhong 提出了一种在线版本的球面 *K*-Means 算法(online spherical *K*-means,简称 OSKM)[24],并扩展成流数据 OSKM 算法(streaming OSKM,简称 SOSKM)[16].实验结果表明,SOSKM 算法可以有效地处理大规模文本数据流的聚类问题.

我们知道,竞争理论的学习规则可以分为 WTA(winner take all)与 WTM(winner take more)两种,分别称为硬竞争学习(hard competitive learning)和软竞争学习(soft competitive learning)[27,28].易知,上述 ScaleKM, CluStream 和 SOSKM 等算法都属于硬划分可扩展聚类(crisp scalable clustering)的情况,即每个样本点要么完全属于特定的聚类中心,要么完全不属于.值得注意的是,WTA 规则会出现这样的问题,即对于某个样本点,获胜节点只有 1 个,因此在某些初始节点的情况下,学习过程可能存在死节点(dead node)或不充分利用(under utilization)的现象[3,28,29].为此,研究人员放宽了 WTA 规则的限制,提出了 WTM 规则.通过引入模糊隶属度等方法,削弱了学习过程对初始节点的依赖.根据 WTM 规则,软竞争学习策略依据各个样本到多个聚类中心之间度量的差异对聚类中心进行调整.为了利用软竞争学习策略的优势,Hall 等人改进了已有可扩展聚类算法,引入模糊隶属度,提出了两种软划分的可扩展聚类方法:单次遍历模糊聚类算法(single-pass fuzzy *C*-means,简称 SFCM)和在线模糊聚类算法(online fuzzy *C*-means,简称 OFCM)[17−19].在 SFCM 和 OFCM 的聚类过程中,单个样本不完全属于某个特定的聚类中心,需要利用模糊隶属度矩阵计算过去数据样本的统计信息,与传统的基于硬划分可扩展聚类算法有明显的不同.实验结果表明,SFCM 和 OFCM 算法对于大规模数据或者数据流可以得到很好的聚类结果[17−19].

同样地,针对高维数据流或者大规模数据,也需要提出有效的基于数据流的软子空间聚类算法.为此,本文利用模糊可扩展聚类框架,与熵加权软子空间聚类算法 EWSC 相结合,提出了熵加权流数据软子空间聚类算法 EWSSC.EWSSC 算法不仅可以在高维数据流的聚类过程中准确地发现各个数据簇相应的局部子空间的结构特征,而且利用了模糊可扩展聚类策略,能够有效地对流数据进行聚类分析.

2 基于数据流的熵加权软子空间聚类算法

本节将模糊可扩展聚类框架(fuzzy scalable clustering framework)与熵加权软子空间聚类方法 EWSC 相结合,提出了熵加权流数据软子空间聚类算法(entropy-weighting streaming subspace clustering,简称 EWSSC). EWSSC 算法的基本思路是:将高维数据流分割成若干个数据子块,针对这些数据子集分别进行处理,每个数据子块的大小由数据流的速度和内存大小所决定.与文献[17−19]类似,EWSSC 算法首先对初次到达的数据子块利用 EWSC 算法进行划分,得到各个数据子块的模糊隶属度、聚类中心以及各个数据簇的特征加权系数;进一步地,将每个数据簇表示成加权聚类中心的形式,其权值通过对所有样本点到该聚类中心的模糊隶属度求和得

到;随后,刚得到的加权聚类中心与新到达的数据子块一起,利用加权 EWSC 算法再次进行聚类.EWSSC 算法不断重复上述迭代,直到数据流结束或者所有的数据样本被完全遍历为止.下面对 EWSSC 算法进行详细介绍.

2.1 聚类中心权值的计算

具体而言,考虑到包含 N 个样本的高维数据流,t 时刻得到了包含 N_t 个新样本的数据子块,新到数据样本的权值表示成 $q_j(t)=1,j=1,\ldots,N_t$;前一时刻已得到 C 个加权聚类中心 $v_i(t-1)$,其权值表示为 $p_i(t-1),i=1,\ldots,C$.对于第一个数据子块,$v_i(0)$初始化为空集,$p_i(0)=0$;利用 EWSC 算法,N_1 个数据样本 $x_j(1)$被划分到 C 个聚类中心 $v_i(1)$,其对应的中心权值 $p_i(1)$表示成

$$\begin{aligned} p_i(1) &= \sum_{j=1}^{N_1}(u_{ij})q_j(1)+\sum_{i'=1}^{C}(u_{ii'})p_{i'}(0) \\ &= \sum_{j=1}^{N_1}(u_{ij})q_j(1) \\ &= \sum_{j=1}^{N_1}u_{ij}, 1\leqslant i\leqslant C \end{aligned} \tag{3}$$

其中,u_{ij} 表示样本 $x_j(1)$属于聚类中心 $v_i(1)$的模糊隶属度,$1\leqslant i\leqslant C,1\leqslant j\leqslant N_1$.

随后,当第 2 个数据子块到达时,EWSC 算法将 N_2 个新到达的数据样本 $x_j(2)$和前一时刻所得到的 C 个加权聚类中心 $v_i(1)$再次划分成 C 个新的数据簇.由此,N_2+C 个样本被划分到 C 个新的聚类中心 $v_i(2)$,其权值 $p_i(2)$表示成

$$\begin{aligned} p_i(2) &= \sum_{j=1}^{N_2}(u_{ij})q_j(2)+\sum_{i'=1}^{C}(u_{ii'})p_{i'}(1) \\ &= \sum_{j=1}^{N_2}u_{ij}+\sum_{i'=1}^{C}(u_{ii'})p_{i'}(1),\ 1\leqslant i\leqslant C \end{aligned} \tag{4}$$

其中,$q_j(2)=1$;$p_{i'}(1)$是聚类中心 $v_{i'}(1)$的权值;u_{ij} 和 $u_{ii'}$分别是 $x_j(2)$和 $v_{i'}(1)$属于新的聚类中心 $v_i(2)$的模糊隶属度,$1\leqslant i\leqslant C,1\leqslant i'\leqslant C,1\leqslant j\leqslant N_2$.

类似地,当第 3 个数据子集到达时,N_3 个新到达的数据样本 $x_j(3)$和前一时刻得到的 C 个加权聚类中心 $v_i(2)$再次被划分到 C 个新的加权聚类中心 $v_i(3)$.以此类推,当第 t 个数据子块到达后,EWSC 算法将 N_t 个数据样本 $x_j(t)$和前一时刻得到的 C 个加权聚类中心 $v_i(t-1)$划分成 C 个新的数据簇,历史数据的统计信息被表示成加权聚类中心 $v_i(t)$的形式,其权值为

$$\begin{aligned} p_i(t) &= \sum_{j=1}^{N_t}(u_{ij})q_j(t)+\sum_{i'=1}^{C}(u_{ii'})p_{i'}(t-1) \\ &= \sum_{j=1}^{N_t}u_{ij}+\sum_{i'=1}^{C}(u_{ii'})p_{i'}(t-1), 1\leqslant i\leqslant C \end{aligned} \tag{5}$$

其中,$q_j(t)=1$;$p_{i'}(t-1)$是 $t-1$ 时刻加权聚类中心 $v_{i'}(t-1)$的权值; u_{ij} 和 $u_{ii'}$分别是 $x_j(t)$和 $v_{i'}(t-1)$属于新的聚类中心 $v_i(t)$的模糊隶属度,$1\leqslant i\leqslant C,1\leqslant i'\leqslant C,1\leqslant j\leqslant N_t$.

容易证明,$\sum_{i=1}^{C}p_i(1)=N_1,\sum_{i=1}^{C}p_i(2)=N_1+N_2$.由此类推,对于任意时刻聚类中心 $v_i(t)$的权值 $p_i(t)$,满足:

$$\sum_{i=1}^{C}p_i(t)=N_1+N_2+\ldots+N_t.$$

我们知道,$v_i(t)$是通过对 N_t 个新到达的数据样本 $x_j(t)$和 $t-1$ 刻获得的 C 个加权聚类中心 $v_i(t-1)$进行聚类划分而得到.前一时刻的加权聚类中心 $v_i(t-1)$是利用 $t-1$ 时刻到达的 N_{t-1} 个数据样本 $x_j(t-1)$和 $t-2$ 时刻获得的加权聚类中心 $v_i(t-2)$所得到的,等等.由此可见,计算第 t 时刻的聚类中心 $v_i(t)$,需要利用 $t-1$ 时刻的加权聚类中心 $v_i(t-1)$,同时也意味着利用到 N_{t-1} 个数据样本 $x_j(t-1)$以及利用到 $t-2$ 时刻的加权聚类中心 $v_i(t-2)$的统计信息,等等.所以,EWSSC 算法可以有效地总结数据流聚类过程中的历史数据特征,并将过去时刻数据子块的统计信息随着时间的延续不断进行传递.

2.2 加权EWSC算法

为了更好地利用模糊可扩展聚类框架,EWSSC 算法需要对 EWSC 算法的聚类中心 v_{ik} 和特征加权系数 w_{ik} 的迭代公式进行修改,使得算法在聚类过程中可以考虑以往聚类中心的权值.文献[17]中给出了加权 FCM 算法的目标函数及其迭代公式.与文献[17]类似,本文定义加权 EWSC 算法(weighted EWSC)的目标函数 J_{WEWSC}:

$$\begin{cases} J_{\mathrm{WEWSC}}=\sum_{j=1}^{N}\sum_{i=1}^{C}u_{ij}^{m}g_j\sum_{k=1}^{D}w_{ik}(x_{jk}-v_{ik})^2+\gamma\sum_{i=1}^{C}\sum_{k=1}^{D}w_{ik}\log w_{ik} \\ \text{s.t. } 0\leqslant u_{ij}\leqslant 1,\sum_{i=1}^{C}u_{ij}=1;0\leqslant w_{ik}\leqslant 1,\sum_{k=1}^{D}w_{ik}=1 \end{cases} \tag{6}$$

其中,g_j表示第 j 个数据样本 x_j 的权值大小.

利用 Lagrange 乘子优化方法最小化公式(6),得到加权 EWSC 算法模糊隶属度 u_{ij}、聚类中心 v_{ik} 和特征加权系数 w_{ik} 的迭代公式,如定理 1 所示.

定理 1. 给定 m>1 和γ>0,最小化加权 EWSC 算法的目标函数(6),当且仅当:

(1) 模糊隶属度 u_{ij} 的迭代公式为

$$\begin{cases} u_{ij}=\dfrac{(d_{ij})^{-1/m-1}}{\sum_{s=1}^{C}(d_{sj})^{-1/m-1}} \\ d_{ij}=\sum_{k=1}^{D}w_{ik}(x_{jk}-v_{ik})^2 \end{cases} \tag{7}$$

(2) 聚类中心 v_{ik} 的迭代公式为

$$v_{ik}=\frac{\sum_{j=1}^{N}g_j(u_{ij}^{m})x_{jk}}{\sum_{j=1}^{N}g_j(u_{ij}^{m})} \tag{8}$$

(3) 特征加权系数 w_{ik} 的迭代公式为

$$\begin{cases} w_{ik}=\dfrac{\exp(-\sigma_{ik}/\gamma)}{\sum_{s=1}^{D}\exp(-\sigma_{is}/\gamma)} \\ \sigma_{ik}=\sum_{j=1}^{N}g_j(u_{ij}^{m})(x_{jk}-v_{ik})^2 \end{cases} \tag{9}$$

定理 1 的证明过程可以参考文献[9,12].

具体而言,t 时刻 EWSSC 算法需要将 N_t 个数据样本 x_j 以及前一时刻得到的 C 个加权聚类中心 $v_i(t-1)$划分成 C 个新的数据簇,其中,x_j 的权值 q_j=1,j=1,…,N_t;加权聚类中心 $v_i(t-1)$的权值为 $p_i(t-1)$,i=1,…,C.根据公式(8)和公式(9),可以得到 t 时刻 EWSSC 算法的聚类中心 v_{ik} 的迭代公式为

$$\begin{aligned} v_{ik}(t)&=\frac{\sum_{j=1}^{N}g_j(u_{ij}^{m})x_{jk}}{\sum_{j=1}^{N}g_j(u_{ij}^{m})} \\ &=\frac{\sum_{j=1}^{N_t}q_j(u_{ij}^{m})x_{jk}+\sum_{i'=1}^{C}p_{i'}(t-1)(u_{ii'}^{m})v_{i'k}(t-1)}{\sum_{j=1}^{N_t}q_j(u_{ij}^{m})+\sum_{i'=1}^{C}p_{i'}(t-1)(u_{ii'}^{m})} \\ &=\frac{\sum_{j=1}^{N_t}u_{ij}^{m}x_{jk}+\sum_{i'=1}^{C}p_{i'}(t-1)(u_{ii'}^{m})v_{i'k}(t-1)}{\sum_{j=1}^{N_t}u_{ij}^{m}+\sum_{i'=1}^{C}p_{i'}(t-1)(u_{ii'}^{m})} \end{aligned} \tag{10}$$

t 时刻特征加权系数 w_{ik} 的迭代公式为

$$\begin{cases} w_{ik}(t)=\dfrac{\exp(-\sigma_{ik}/\gamma)}{\sum_{s=1}^{D}\exp(-\sigma_{is}/\gamma)} \\ \sigma_{ik}=\sum_{j=1}^{N}g_j(u_{ij}^{m})(x_{jk}-v_{ik})^2 \\ \quad=\sum_{j=1}^{N_t}q_j(u_{ij}^{m})(x_{jk}-v_{ik})^2+\sum_{i'=1}^{C}p_{i'}(t-1)(u_{ii'}^{m})(v_{i'k}(t-1)-v_{ik})^2 \\ \quad=\sum_{j=1}^{N_t}u_{ij}^{m}(x_{jk}-v_{ik})^2+\sum_{i'=1}^{C}p_{i'}(t-1)(u_{ii'}^{m})(v_{i'k}(t-1)-v_{ik})^2 \end{cases} \tag{11}$$

2.3 EWSSC算法的实现细节及流程图

EWSSC 算法在初始化过程中,特征加权系数 $w_{ik}(0)$服从均匀分布,满足等式约束 $\sum_{k=1}^{D}w_{ik}(0)=1,1\leqslant i\leqslant C$.对第 1 个数据子块的聚类中心利用 K-Means++算法进行初始化.即当第 1 个初始聚类中心被确定以后,新的聚

类中心从剩下的数据样本中按照其离已有的聚类中心度量间距的反比进行选择,使得所选的初始化聚类中心尽可能地相互分离,由此保证初始聚类中心的多样性[30].

同时,在 EWSSC 算法的具体执行过程中,为了充分利用过去数据子块的统计信息,并保持聚类结果的一致性,对于后续数据子集的聚类中心和特征加权系数的初始化,均利用了前一次数据子集聚类划分得到的中心和加权系数.当整个数据流结束或者数据集的全部样本被遍历以后,EWSSC 算法得到了最终的聚类中心 v_{ik} 和特征加权系数 w_{ik}.

基于上述的描述,图 1 给出了基于数据流的熵加权软子空间聚类算法流程.

熵加权流数据软子空间聚类算法 EWSSC 步骤.
输入:数据流 $X=\{x_1,x_2,\ldots\}\subset R^D$,聚类数目 C,数据子块大小 S,加权 EWSC 算法最大迭代次数 M.
初始化:对于 C 个历史聚类中心 $v_i(0)$初始化为空集,相应的中心加权系数 $p_i(0)$设置为 0,设置数据子块指数 $t=1$.
重复:
(I) 获取 S 个新的数据样本,设置新到达数据样本的加权系数为 1;当 $t=1$ 时,初始化特征加权系数 $w_{ik}(0)$, $1\leqslant i\leqslant C,1\leqslant k\leqslant D$,并利用 K-means++算法从新到达的数据样本中选择 C 个初始聚类中心 $v_i(0),1\leqslant i\leqslant C$;
当 $t>1$ 时,聚类中心和特征加权系数的初始化利用前一次数据子集聚类划分得到的中心和加权系数.
(II) 对 S 个新到达的数据样本和 C 个历史聚类中心 $v_i(t-1)$利用加权 EWSC 算法进行聚类划分.
(a) 设置迭代指数 $itr=1$,
(i) 利用公式(7)计算数据样本和历史聚类中心到各个聚类中心的模糊隶属度 u_{ij};
(ii) 根据公式(10)计算 C 个新的聚类中心 $v_i(t)$;
(iii) 根据公式(11)计算各个数据簇的特征加权系数 $w_{ik}(t)$.
(b) $itr=itr+1$.
(c) 重复步骤(a)和步骤(b),直到迭代指数 itr 达到最大次数 M 或者满足加权 EWSC 算法的停止条件.
(III) 对于每个数据簇,利用公式(5)计算其聚类中心对应的加权系数 $p_i(t),i=1,\ldots,C$.
(IV) $t=t+1$.
直到数据流结束或者整个数据集被处理完毕为止.
输出:最终的聚类中心 $V=\{v_i,1\leqslant i\leqslant C\}$和数据簇特征加权系数矩阵 $W=\{w_{ik}|1\leqslant i\leqslant C,1\leqslant k\leqslant D\}$.

Fig.1 Flowchart of the proposed Entropy-Weighting Streaming Subspace Clustering algorithm

图 1 熵加权流数据软子空间聚类算法流程

2.4 算法收敛性与计算复杂性分析

在文献[17]中,Hall 等人证明了 SFCM 和 OFCM 这两种软划分可扩展聚类算法的收敛性.对于熵加权流数据软子空间聚类算法 EWSSC 而言,当特征加权系数 w_{ik} 大小相同且各个数据子块均利用加权 FCM 算法进行聚类划分时,EWSSC 退化成 SFCM 的形式,由此可以利用文献[17]类似的证明过程来说明 EWSSC 算法的收敛性.

进一步地,本文讨论 EWSSC 算法的计算复杂性.首先给出如下定义:

- D:数据样本特征空间的维数;
- S:各个时刻到达的数据子块大小;
- C:整个数据流样本集包含的数据簇的个数;
- s:EWSSC 算法需要遍历的数据子块的个数.

对于新到达的数据子块,利用加权 EWSC 算法,单次迭代计算 S 个新到达的数据样本和 C 个加权聚类中心的模糊隶属度 u_{ij},聚类中心 v_{ik} 和特征加权系数 w_{ik} 所需的计算复杂度为 $O((S+C)CD)$.由于加权 EWSC 算法的最大迭代次数为 M,所以对单个数据子块进行聚类划分的计算复杂度为 $O((S+C)CDM)$.假设熵加权流数据软子空间聚类算法需要遍历 s 个数据子块,所以,EWSSC 算法最终的计算复杂度为 $O(s(S+C)CDM)$.根据文献[12]我们知道,EWSC 算法对于整个数据集或者数据流进行聚类划分所需的计算复杂度为 $O(sSCDM)$,其中,$s\times S$ 表示整个数据集的样本数目,因此,EWSSC 算法的计算复杂度要略高于 EWSC 算法.但在实际的测试比较中,如文献[19,31]所示,可扩展聚类算法在对每个数据子块进行聚类划分的过程中都利用了过去时刻数据子块聚类结果的先验信息进行初始化,所以数据流聚类算法对于各个数据子块聚类划分所需的迭代次数也往往会少于传统的批处理聚类方法.本文在后续文本数据流上的实验结果也表明:EWSSC 算法与批处理的 EWSC 算法相比,由于在迭代过程中利用了前一次数据子集聚类划分得到的聚类中心和特征加权系数进行初始化,减少了算法的

运行时间.

进一步地,本文考虑通过采样技术降低 EWSSC 算法的时间复杂度.易知,在 EWSSC 算法迭代的初期,由于缺乏充足的数据样本,聚类中心和特征加权系数容易出现较大的调整.随着时间的推移,数据子块的数目将逐渐增加,到达的数据流样本也越来越多,聚类中心和特征加权系数会趋于稳定.因此,在 EWSSC 算法的迭代过程,后续数据子块所包含的样本会出现数据冗余的情况.可以利用采样的方法,对后续到达的数据子集进行随机筛选,降低 EWSSC 算法的计算复杂度.为此,本文基于随机采样策略,提出快速 EWSSC 算法 EWSSC-fast.具体而言,对于第 t 时刻到达的 S 个新的数据样本(假设 EWSSC 算法需要遍历的数据子块的个数为 s),我们随机采样其中的 $\frac{s-t+1}{s}S$ 个样本,和前一时刻得到的 C 个加权聚类中心一起,利用加权 EWSC 算法将其划分成 C 个新的数据簇.本文后续的实验结果表明,利用该随机采样技术,不仅可以有效地减少 EWSSC 算法的运行时间,而且可以得到与 EWSSC 近似一致的聚类划分结果.

3 实验分析

本节针对熵加权流数据软子空间聚类算法进行测试分析,通过选取不同的人造数据集和真实数据集进行实验对比.首先介绍实验安排和测试环境;然后说明 3 种聚类结果的性能评价指标;进一步地,对于 EWSSC 算法,分别给出其在人造数据集和文本数据流上的聚类实验结果.

3.1 实验安排和测试环境

本文将 EWSSC 与 6 种聚类算法进行对比,包括两种数据流聚类算法:流数据在线球面 K-Means 算法 SOSKM[16]和单次遍历模糊聚类算法 SFCM[17,19];两种软子空间聚类算法 EWSC[12]和 FWSC[11];以及两种批处理聚类算法 SPKM(batch spherical K-means)[32,33]和 FCM[34].进一步地,本文在文本数据流上对第 2.4 节提出的快速 EWSSC 算法 EWSSC-fast 进行了实验分析,以测试利用随机采样技术降低 EWSSC 算法时间复杂度的有效性.

本文选取了两组实验数据集进行对比测试.对于所有数据,各个特征均进行归一化处理,使得数据集的各维特征都在[0 1]区间.同时,为了保证实验比较的公平性,本文对所有聚类算法均进行 20 次的重复实验,对各个算法测试结果的平均值和方差进行比较.所有的实验都运行在 Intel Xeno(R) CPU 2.53-GHz 的工作台上,利用 MATLAB 软件进行仿真.

3.2 评价指标

本文采用 3 种聚类评价指标进行实验结果比较:聚类准确率 CA(clustering accuracy)[35]、互信息 NMI[36]和 Rand 指数 RI[37].

- CA 用于统计在所有的数据样本中,聚类算法正确划分的样本所占的比率,通常被定义为

$$CA=\sum_{l=1}^{C}n_l\Big/N \tag{12}$$

其中,n_l 表示数据样本中被正确划分为类标 l 的样本个数,N 是数据集包含的全部数据样本的总数.

- NMI 计算聚类划分结果和实际样本类标进行两两配对后得到的平均互信息大小,定义如下:

$$NMI=\frac{\sum_{i=1}^{C}\sum_{j=1}^{C}n_{ij}\log((N\cdot n_{ij})/(n_i\cdot n_j))}{\sqrt{\sum_{i=1}^{C}n_i\log(n_i/N)\cdot\sum_{j=1}^{C}n_j\log(n_j/N)}} \tag{13}$$

其中,n_{ij} 表示聚类结果为 i 而实际类标为 j 的数据样本个数,n_i 表示聚类结果为 i 的样本总数,n_j 表示实际类标为 j 的样本总数,N 是数据集包含的全部数据样本的总数.

- RI 度量了对数据集进行聚类划分和数据集实际划分之间,两种划分结果的一致性,通常被定义为

$$RI=\frac{n_{00}+n_{11}}{N(N-1)/2} \tag{14}$$

其中,n_{00} 表示数据集构成的全部样本对,每对样本属于不同的实际类标,同时被划分到不同聚类结果的样本对

个数;n_{11} 表示全部样本对,每对样本具有相同的实际类标,同时被划分到相同聚类结果的样本对个数;N 是数据集包含的全部样本的总数.

易知,聚类准确率 CA、互信息 NMI 和 Rand 指数 RI 的取值范围均在[0 1]区间,值越高表示聚类结果越准确,值越小表示聚类划分与实际划分的差距越大[38].当聚类结果和实际类标完全一致时,上述 3 种聚类评价指标均为 1.

3.3 人造数据集

本节按照文献[9]生成了人造数据集,以测试 EWSSC 算法的性能.人造数据集包括 4 200 个样本,7 个数据簇,每个数据簇包含 600 个样本,每个样本 50 维.其中,不同数据簇包含了不同的子空间特征结构,各个数据簇的特征分布如图 2 所示.对于每个数据簇,数据特征在相关子空间上服从高斯分布,在不相关子空间上服从均匀分布.易知,10 维~15 维特征和 45 维~50 维特征分别对于数据簇 1 和数据簇 7 更加重要,应具有更高的特征权值.因此,子空间聚类算法将更适用于该人造数据集的聚类划分.下面,本节在人造数据集上首先对 EWSSC 算法参数的选择以及数据块大小的设置进行实验分析.

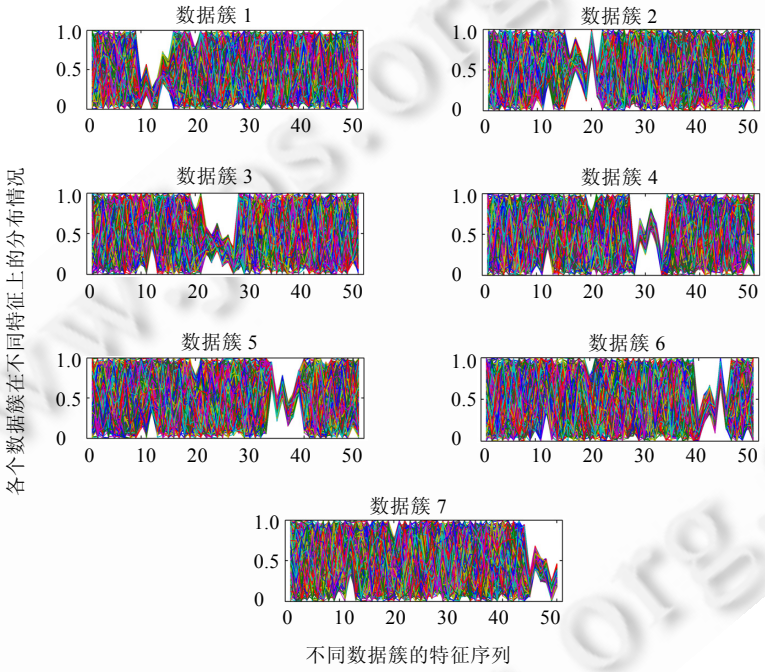
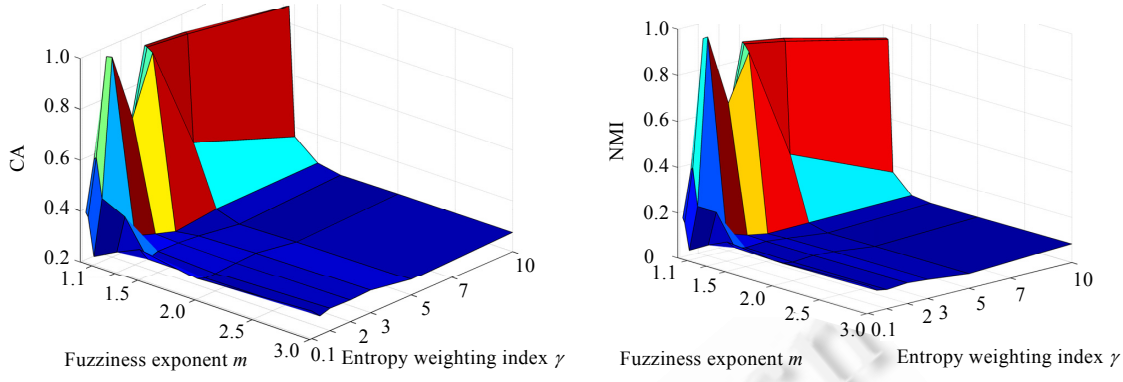


Fig.2 Synthetic dataset containing 7 clusters embedded in different subspaces

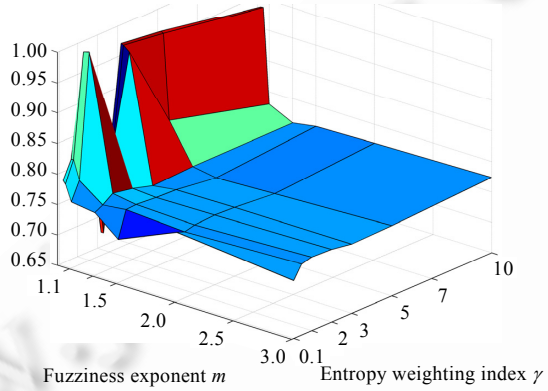
图 2 人造数据集所包含的 7 个数据簇对于不同子空间的分布情况

3.3.1 参数的选择与设置

对于模糊聚类和软子空间聚类算法而言,算法参数的选择一直是机器学习和数据挖掘领域的开放问题.本文将生成的人造数据集按全部样本总数的 10%分割成若干个数据子块,针对此数据集对 EWSSC 算法参数的选择进行分析.EWSSC 算法含有两个参数需要设置:模糊权重指数 m 和熵加权指数 γ.图 3 分别给出了 m 取值在{1.03,1.05,1.1,1.3,1.5,2,3}以及 γ 取值在{0.1,0.5,1,2,3,5,10}时,CA,NMI 和 RI 的平均值随算法参数变化的网格图.从图 3 中可以看出,EWSSC 算法的聚类结果随着 m 和 γ 的取值呈现出一定的变化趋势.当 m 的取值在 1.03~1.1 之间,同时 γ 的取值在 1~10 之间时,EWSSC 算法的实验结果相对稳定,可以获得较好的聚类划分结果.



(a) CA (b) NMI



(c) RI

Fig.3 Performance variation w.r.t. different parameters

图 3 聚类结果在不同算法参数情况下的变化比较

在后面的实验测试中,本文将针对各个聚类算法采用相应的参数设置.具体而言,对于 EWSSC 算法,其模糊隶属度的模糊权重指数 m 设置成 1.05;对于 EWSSC 和 EWSC 算法,其熵加权指数 γ 设置成 5;对于 FWSC 算法,其模糊加权指数 τ 设置成 2.对于所有的批处理聚类算法,设置各种算法的最大迭代次数为 100.

3.3.2 数据子块大小的选择

进一步地,为了分析数据子块大小对 EWSSC 算法聚类结果的影响,本文将整个人造数据集分割成若干数目不同的数据流,对其进行聚类划分结果的比较.图 4 分别给出了 CA,NMI 和 RI 的平均值在人造数据集上随数据子块个数的增加聚类结果的变化情况.从图 4 聚类评价指标的变化曲线中可以看出,EWSSC 算法的聚类划分结果随着数据子块个数的增加呈现下降的趋势.当数据子块数目较少或者数据子块较大时,可以得到更好的聚类结果.值得注意的是,当数据子块过大时,EWSSC 算法也将失去处理数据流问题的意义.因此,图 4 的实验结果表明,当人造数据集的数据子块数目小于等于 20,即数据子块的大小超过整个数据集的 5%时,EWSSC 算法的实验结果相对稳定,可以获得较好的聚类划分结果.当然,数据子块大小的合理选择还与数据集的规模有关.在今后的研究中,我们将针对不同大小情况下的高维数据集进行测试,权衡数据子块大小和最终聚类性能的联系,进一步展开相关方面的理论和实验说明.

在后续的实验测试中,对于各种数据流聚类算法,本文将各个数据子块的大小分别设置为整个数据集的 5%,10%和 20%这 3 种情况进行对比分析.4 种批处理聚类算法将在整个数据集上进行测试.

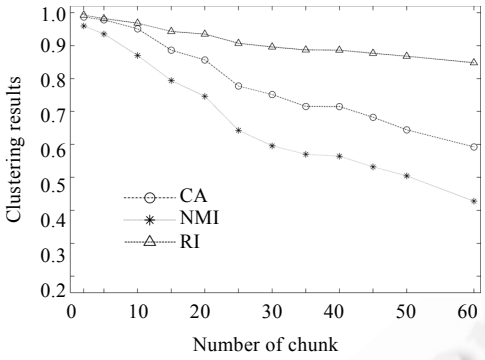


Fig.4 Performance variation w.r.t. different number of chunk

图 4 聚类结果在不同数据子块大小情况下的变化比较

3.3.3 不同聚类算法的实验对比

表 1 给出了 7 种不同的聚类算法在人造数据集上的 CA,NMI 和 RI 的平均值和标准差.从表 1 中可以看出:

1) 熵加权流数据软子空间聚类算法 EWSSC 对于数据流的聚类划分获得了良好的实验结果,EWSSC 基本可以获得与批处理软子空间聚类算法 EWSC 几乎一致的聚类划分.
2) 数据流聚类算法的划分结果随着数据子块大小的变化呈现一定的趋势,随着单个数据子块所占数据集百分比的增大(数据子块变大),聚类结果也得到了相应的提高.

表 1 的最后一列给出了 7 种聚类算法各自所需的运行时间.我们发现,由于该人造数据集具有比较明显的子空间结构特征,因此在大多数情况下,数据流聚类算法与批处理聚类算法所需的运行时间比较接近.

Table 1 Clustering results obtained for synthetic dataset

表 1 针对人造数据集所得的聚类结果

数据子块大小 (所占数据集百分比)		CA		NMI		RI		Time (s)
		Mean	Std	Mean	Std	Mean	Std	
EWSSC	5%	0.854 1	0.028 6	0.777 1	0.021 3	0.945 6	0.006 8	4.509 7
	10%	0.953 9	0.004 4	0.879 2	0.008 9	0.974 3	0.002 3	5.979 1
	20%	**0.981 2**	0.007 5	0.949 9	0.015 6	**0.985 0**	0.004 1	4.332 4
EWSC		0.975 5	0.070 3	**0.954 1**	0.073 4	0.981 9	0.023 0	4.679 7
FWSC		0.972 2	0.003 5	0.944 3	0.002 1	0.975 6	0.003 4	2.658 1
SOSKM	5%	0.445 0	0.041 6	0.254 7	0.036 3	0.788 9	0.014 4	1.137 3
	10%	0.616 0	0.032 5	0.435 3	0.034 3	0.847 3	0.009 7	1.121 5
	20%	0.706 5	0.017 9	0.548 5	0.022 6	0.882 5	0.006 0	1.115 5
SPKM		0.724 8	0.006 3	0.592 5	0.008 8	0.893 9	0.001 8	1.563 9
SFCM	5%	0.485 5	0.030 2	0.322 2	0.022 1	0.739 7	0.008 6	3.970 9
	10%	0.580 4	0.039 2	0.427 5	0.036 3	0.774 5	0.011 8	5.762 4
	20%	0.633 0	0.010 8	0.488 7	0.011 6	0.793 8	0.003 3	5.611 8
FCM		0.407 3	0.037 3	0.254 9	0.030 8	0.708 4	0.025 1	0.422 9

表 2 列举了在人造数据集上,上述 3 种软子空间聚类算法 EWSSC,EWSC 和 FWSC 对各个数据簇子空间结构的检测情况.加粗的数字表示被错误检测的子空间特征.易知,EWSSC 算法同样可以得到良好的数据子空间检测结果,特别在数据子块大小为 20%的情况下,获得了最高的检测准确率.

进一步地,本文对各种聚类算法的实验结果进行了显著性分析.通过利用基于 5%显著性水平的 t 检验方法,得到了 EWSSC 算法与其他聚类算法之间的 P 值大小.表 3 给出了 EWSSC 算法关于 CA,NMI 和 RI 聚类结果的 P 值.作为零假设,本文认为两种算法的聚类结果没有显著性区别,备择假设则认为聚类结果之间存在明显的显著性区别.从表 3 中我们可以看出:大多数实验结果的 P 值均小于 0.05(5%显著性水平),表明 EWSSC 算法与其他聚类算法相比存在着明显的不同;而 EWSSC 与 EWSC 算法在数据子块较大的情况下,实验结果的显著性

区别相对较小,说明 EWSSC 和 EWSC 算法可以获得几乎一致的聚类划分结果.

Table 2 Subspace detection results of different algorithms for synthetic dataset

表 2 针对人造数据集不同聚类算法的子空间结构检测结果

对应子空间的特征序列		各个数据簇对应子空间的特征检测序列														
		EWSSC (5%)			EWSSC (10%)			EWSSC (20%)			EWSC			FWSC		
数据簇 1	10~15	14	15	13	14	13	15	10	11	15	10	15	12	10	15	12
		11	12	10	10	11	12	14	13	12	14	11	13	14	11	13
数据簇 2	16~21	18	17	21	18	16	21	16	18	21	16	18	21	16	18	21
		19	**24**	16	20	19	17	19	20	17	19	20	17	20	19	17
数据簇 3	22~27	25	24	23	22	25	23	22	25	26	22	26	23	22	26	23
		27	26	22	26	24	27	23	27	24	24	25	27	24	25	27
数据簇 4	28~33	28	31	29	32	33	30	32	30	28	32	33	28	32	33	28
		30	33	32	29	28	31	33	31	29	**38**	29	31	30	29	31
数据簇 5	34~39	36	37	**10**	34	38	36	34	36	37	36	35	34	36	35	34
		38	35	45	37	**19**	39	38	35	39	38	39	37	38	39	37
数据簇 6	40~45	44	43	42	45	43	42	45	43	41	45	42	40	45	42	40
		45	40	41	40	44	41	40	44	42	43	44	41	44	43	41
数据簇 7	45~50	**10**	48	49	46	45	47	48	49	45	45	46	50	45	46	**35**
		47	50	45	50	48	49	46	47	50	48	49	47	48	49	47
子空间结构检测准确率		**39/42**			**41/42**			**42/42**			**41/42**			**41/42**		

Table 3 *P*-Values produced by *t*-test comparing EWSSC

表 3 EWSSC 与其他聚类算法进行 *t* 检验得到的 *P* 值

数据子块大小(%)		*P* 值									
		EWSC	FWSC	SOSKM			SPKM	SFCM			FCM
				(5%)	(10%)	(20%)		(5%)	(10%)	(20%)	
AC	5	1.5646e−04	4.8125e−14	1.2434e−14	5.5268e−11	3.3966e−08	8.9301e−08	2.2238e−12	2.1606e−07	7.7722e−07	6.2336e−16
	10	same	1.7755e−19	1.9811e−18	5.5472e−17	7.4865e−19	5.0545e−25	2.9112e−18	7.2100e−14	1.8408e−21	7.7728e−20
	20	same	2.4693e−08	6.6872e−19	1.1058e−17	1.3431e−19	2.2402e−24	7.0852e−19	8.7025e−15	8.1234e−22	2.9825e−20
NMI	5	1.2975e−07	8.2116e−20	2.9863e−17	2.4474e−13	1.2199e−10	8.4582e−10	8.1366e−16	1.4201e−09	9.3458e−10	3.7582e−18
	10	0.001 5	7.3605e−20	8.3200e−21	2.0095e−18	7.3263e−19	8.9303e−23	9.4501e−22	4.4286e−16	1.0296e−21	5.3240e−22
	20	same	1.8894e−09	2.9792e−21	3.5869e−19	1.4807e−19	6.6232e−22	5.8305e−22	3.5138e−17	2.2568e−21	2.6463e−22
RI	5	5.9997e−07	1.7760e−18	1.8378e−15	4.7798e−13	9.0804e−10	1.2862e−08	1.9435e−14	1.7896e−08	3.5406e−08	6.3743e−13
	10	3.3751e−04	1.7253e−19	1.2522e−18	1.9401e−18	4.4483e−19	6.1015e−24	7.8630e−20	5.2940e−15	1.8012e−21	2.6388e−15
	20	0.003 0	1.9455e−08	3.1620e−19	3.4305e−19	9.0557e−20	1.3267e−22	2.0335e−20	3.5255e−16	2.5165e−21	6.0737e−16

3.4 文本数据流

本节对 EWSSC 算法在真实数据集上进行聚类结果的比较.本文从 20-Newsgroups 语料库[39]中选取了 5 组不同结构的文本数据集进行对比实验.20-Newsgroups 语料库包含了从 20 种新闻主题中抽取的 20 000 篇不同内容的消息,平均每个主题包含了近 1 000 篇新闻.在具体的操作中,本文利用 Bow toolkit[40]对 20-Newsgroups 语料库进行预处理,获得了 18 846 篇新闻,其中每篇新闻都表示成 26 214 维的 tf-idf 特征向量.为了保证实验数据集的多样性,本文从 20-Newsgroups 语料库的不同新闻主题中进行抽取,表 4 列举了选取的 5 组文本数据集的基本结构特征,其中删除了部分 tf 较低的词汇项.例如,数据集 A4 包含了从 comp.graphics,rec.sport.hockey, sci.crypt 和 talk.religion.misc 主题中得到的 3 591 篇文本;数据集 A8 包含了从 comp.windows.x,rec.autos, rec.sport.baseball 等主题中得到的 7 661 篇文本.容易看出,A2 和 A4 数据集的文本由于选取于不同的新闻类别,具有更加明显的语义差别;B2 和 B4 的数据簇因为包含了更多相似或重叠的词汇,语义信息更加一致,所以增加了聚类算法划分的难度.

Table 4 Text datasets used in the experiment

表 4 实验中所用文本数据集

数据集	来源主题	文档数目	特征项	文档类别数
A2	alt.atheism comp.graphics	799 973	229	2
B2	talk.politics.guns talk.politics.mideast	910 940	247	2
A4	comp.graphics rec.sport.hockey sci.crypt talk.religion.misc	973 999 991 628	404	4
B4	alt.atheism rec.sport.baseball talk.politics.guns talk.politics.misc	799 994 910 775	435	4
A8	alt.atheism comp.graphics comp.windows.x rec.autos rec.sport.baseball sci.med sci.space talk.politics.mideast	799 973 988 990 994 990 987 940	543	8

3.4.1 不同聚类算法的实验对比

为了得到数据流聚类算法在文本数据流上的实验结果,本文将 5 组不同结构的数据集按照样本总数的 5%, 10%和 20%随机划分成多个数据子块.对于批处理聚类算法,则在整个样本集上进行聚类划分.表 5~表 9 给出了 8 种聚类算法(包含 EWSSC-fast)在文本数据流上,CA,NMI 和 RI 的平均值和标准差.

Table 5 Clustering results obtained for text dataset A2

表 5 针对文本数据集 A2 所得的聚类结果

数据子块大小(所占数据集百分比)		CA		NMI		RI		Time (s)
		Mean	Std	Mean	Std	Mean	Std	
EWSSC	5%	0.965 4	0.001 1	0.807 9	0.003 6	0.942 6	0.001 3	1.492 9
	10%	0.968 7	0.000 8	0.808 5	0.003 8	0.942 8	0.001 5	1.758 8
	20%	**0.970 7**	0.001 0	**0.808 9**	0.004 1	**0.943 0**	0.001 4	2.196 9
EWSSC-fast	5%	0.964 3	0.001 3	0.804 4	0.002 8	0.940 5	0.001 7	1.270 6
	10%	0.968 4	0.001 5	0.808 1	0.006 7	0.942 2	0.002 8	1.443 4
	20%	0.970 3	0.000 4	0.808 7	0.000 6	0.942 8	0.000 2	1.706 0
EWSC		0.965 6	0.000 9	0.797 7	0.004 4	0.939 2	0.001 7	8.387 5
FWSC		0.693 8	0.123 3	0.188 2	0.184 5	0.602 3	0.112 6	2.311 5
SOSKM	5%	0.949 8	0.001 1	0.784 3	0.005 4	0.921 4	0.002 1	2.268 8
	10%	0.952 2	0.001 1	0.795 9	0.005 4	0.934 9	0.002 1	2.381 1
	20%	0.956 1	0.000 8	0.796 5	0.004 2	0.935 8	0.001 6	2.576 5
SPKM		0.961 9	0.001 0	0.804 3	0.004 8	0.937 3	0.001 8	3.781 1
SFCM	5%	0.758 9	0.006 2	0.490 4	0.016 3	0.735 2	0.011 0	0.174 7
	10%	0.771 5	0.000 5	0.598 8	0.002 8	0.744 1	0.001 0	0.329 8
	20%	0.796 3	0.000 6	0.601 6	0.002 7	0.774 2	0.001 1	0.149 5
FCM		0.505 0	0.054 2	0.365 4	0.062 0	0.778 3	0.023 2	0.164 7

Table 6 Clustering results obtained for text dataset B2

表 6 针对文本数据集 B2 所得的聚类结果

数据子块大小(所占数据集百分比)		CA		NMI		RI		Time (s)
		Mean	Std	Mean	Std	Mean	Std	
EWSSC	5%	0.937 0	0.004 0	0.680 8	0.009 3	0.881 9	0.007 0	2.017 2
	10%	0.937 4	0.001 2	**0.684 4**	0.003 7	0.882 6	0.002 1	2.327 6
	20%	**0.938 1**	0.001 6	0.682 7	0.006 0	**0.883 8**	0.002 8	3.180 1
EWSSC-fast	5%	0.934 3	0.006 9	0.676 4	0.012 8	0.877 1	0.012 2	1.775 7
	10%	0.937 1	0.001 5	0.680 6	0.003 1	0.881 9	0.002 8	1.985 8
	20%	0.9379	0.002 3	0.682 5	0.007 1	0.883 0	0.004 0	2.768 7
EWSC		0.930 4	0.004 8	0.652 1	0.016 8	0.870 4	0.008 2	7.846 6
FWSC		0.567 5	0.039 3	0.083 7	0.073 8	0.511 6	0.011 4	3.147 6
SOSKM	5%	0.929 5	0.003 1	0.680 0	0.009 8	0.876 3	0.005 5	2.788 6
	10%	0.930 0	0.001 4	0.680 6	0.005 9	0.877 1	0.002 5	2.966 6
	20%	0.929 8	0.001 0	0.681 2	0.003 2	0.876 9	0.001 7	3.310 0
SPKM		0.9304	0.001 6	0.683 2	0.005 1	0.877 9	0.002 8	8.992 3
SFCM	5%	0.788 9	0.006 2	0.490 4	0.016 3	0.735 2	0.011 0	0.174 7
	10%	0.788 6	0.002 7	0.489 7	0.007 2	0.734 8	0.004 7	0.159 3
	20%	0.789 2	0.002 6	0.492 0	0.006 5	0.735 7	0.004 5	0.189 8
FCM		0.746 2	0.023 8	0.425 7	0.062 5	0.717 3	0.039 0	0.087 2

Table 7 Clustering results obtained for text dataset A4

表 7 针对文本数据集 A4 所得的聚类结果

数据子块大小(所占数据集百分比)		CA		NMI		RI		Time (s)
		Mean	Std	Mean	Std	Mean	Std	
EWSSC	5%	0.914 5	0.002 6	0.757 8	0.004 1	0.918 9	0.002 7	6.438 4
	10%	0.916 3	0.001 9	0.754 7	0.004 2	0.920 7	0.001 6	8.464 1
	20%	0.916 8	0.001 3	0.750 9	0.003 4	0.920 9	0.001 2	11.165 0
EWSSC-fast	5%	0.908 1	0.007 4	0.750 8	0.012 6	0.913 4	0.007 2	4.845 9
	10%	0.918 9	0.004 1	**0.767 6**	0.006 8	0.922 7	0.003 8	7.004 9
	20%	**0.920 7**	0.002 9	0.766 0	0.004 3	**0.924 5**	0.002 9	8.705 3
EWSC		0.846 8	0.075 3	0.687 8	0.064 5	0.881 2	0.042 9	18.607 7
FWSC		0.368 2	0.028 8	0.143 9	0.062 1	0.429 8	0.076 9	9.602 8
SOSKM	5%	0.906 0	0.005 1	0.746 2	0.010 0	0.910 2	0.004 1	11.046 2
	10%	0.900 6	0.001 5	0.752 5	0.003 6	0.914 2	0.001 3	11.735 8
	20%	0.911 0	0.000 6	0.753 8	0.002 0	0.914 5	0.000 6	12.820 4
SPKM		0.910 4	0.001 6	0.749 4	0.004 0	0.914 1	0.001 2	15.875 5
SFCM	5%	0.661 2	0.031 3	0.493 7	0.034 9	0.670 8	0.028 3	1.768 8
	10%	0.700 5	0.003 0	0.538 9	0.005 3	0.707 6	0.002 7	1.647 6
	20%	0.700 7	0.003 0	0.538 9	0.004 3	0.708 1	0.002 4	2.040 4
FCM		0.622 7	0.073 4	0.458 2	0.064 8	0.693 5	0.050 9	0.645 8

Table 8 Clustering results obtained for text dataset B4

表 8 针对文本数据集 B4 所得的聚类结果

数据子块大小(所占数据集百分比)		CA		NMI		RI		Time (s)
		Mean	Std	Mean	Std	Mean	Std	
EWSSC	5%	**0.813 4**	0.027 7	0.593 3	0.019 9	0.855 3	0.013 9	7.265 0
	10%	0.810 6	0.026 8	0.599 2	0.021 0	0.857 2	0.012 1	9.908 3
	20%	0.798 1	0.041 6	0.579 7	0.032 7	0.848 7	0.017 8	13.709 8
EWSSC-fast	5%	0.769 4	0.038 8	0.562 3	0.023 6	0.828 9	0.018 4	5.6211
	10%	0.775 3	0.040 9	0.589 0	0.046 2	0.841 3	0.021 0	7.282 3
	20%	0.806 6	0.042 5	0.616 5	0.032 8	**0.858 6**	0.018 5	10.502 6
EWSC		0.756 4	0.031 0	0.551 0	0.020 8	0.822 3	0.012 7	19.348 7
FWSC		0.360 4	0.035 4	0.109 1	0.052 0	0.435 2	0.079 4	10.393 1
SOSKM	5%	0.780 8	0.022 0	0.624 9	0.022 8	0.833 9	0.016 5	11.614 4
	10%	0.794 2	0.002 1	0.658 0	0.004 1	0.848 6	0.002 3	12.446 3
	20%	0.795 0	0.003 0	**0.661 9**	0.003 3	0.848 8	0.001 7	13.812 5
SPKM		0.767 8	0.024 9	0.612 7	0.033 9	0.840 4	0.008 9	24.584 5
SFCM	5%	0.505 7	0.029 4	0.428 8	0.056 1	0.572 0	0.032 1	1.952 5
	10%	0.537 7	0.031 2	0.436 7	0.051 1	0.609 5	0.019 5	3.508 1
	20%	0.515 8	0.057 3	0.427 6	0.060 1	0.603 2	0.028 7	4.539 8
FCM		0.491 6	0.031 9	0.427 2	0.052 4	0.592 8	0.024 5	0.616 5

Table 9 Clustering results obtained for text dataset A8

表 9 针对文本数据集 A8 所得的聚类结果

数据子块大小 (所占数据集百分比)		CA		NMI		RI		Time(s)
		Mean	Std	Mean	Std	Mean	Std	
EWSSC	5%	0.694 0	0.036 5	0.571 2	0.022 2	0.889 0	0.008 9	17.635 4
	10%	0.729 8	0.028 6	0.606 7	0.011 7	**0.898 8**	0.005 8	23.422 7
	20%	**0.736 1**	0.037 3	**0.607 7**	0.006 9	0.890 3	0.011 2	29.843 4
EWSSC-fast	5%	0.693 9	0.018 4	0.570 7	0.010 3	0.886 9	0.005 4	11.873 2
	10%	0.692 2	0.025 3	0.594 0	0.017 9	0.889 8	0.005 1	15.096 0
	20%	0.718 4	0.020 5	0.606 0	0.013 8	0.892 5	0.005 5	20.111 9
EWSC		0.508 8	0.037 8	0.497 9	0.023 6	0.761 5	0.033 8	35.920 3
FWSC		0.195 3	0.022 9	0.109 9	0.062 7	0.434 0	0.132 6	34.206 2
SOSKM	5%	0.605 7	0.042 5	0.496 6	0.045 1	0.853 0	0.013 6	39.528 1
	10%	0.644 4	0.036 3	0.543 0	0.020 8	0.870 6	0.011 0	42.352 7
	20%	0.724 8	0.015 3	0.603 2	0.009 3	0.890 6	0.005 2	47.394 8
SPKM		0.715 0	0.029 1	0.604 8	0.018 8	0.890 0	0.007 8	54.901 4
SFCM	5%	0.366 9	0.052 3	0.248 3	0.066 9	0.793 6	0.021 7	10.982 3
	10%	0.349 5	0.021 7	0.184 4	0.025 8	0.801 6	0.006 4	7.620 2
	20%	0.309 7	0.012 8	0.152 5	0.012 8	0.768 0	0.003 6	6.977 7
FCM		0.427 2	0.030 0	0.363 6	0.023 4	0.770 5	0.025 6	3.256 6

实验结果表明:

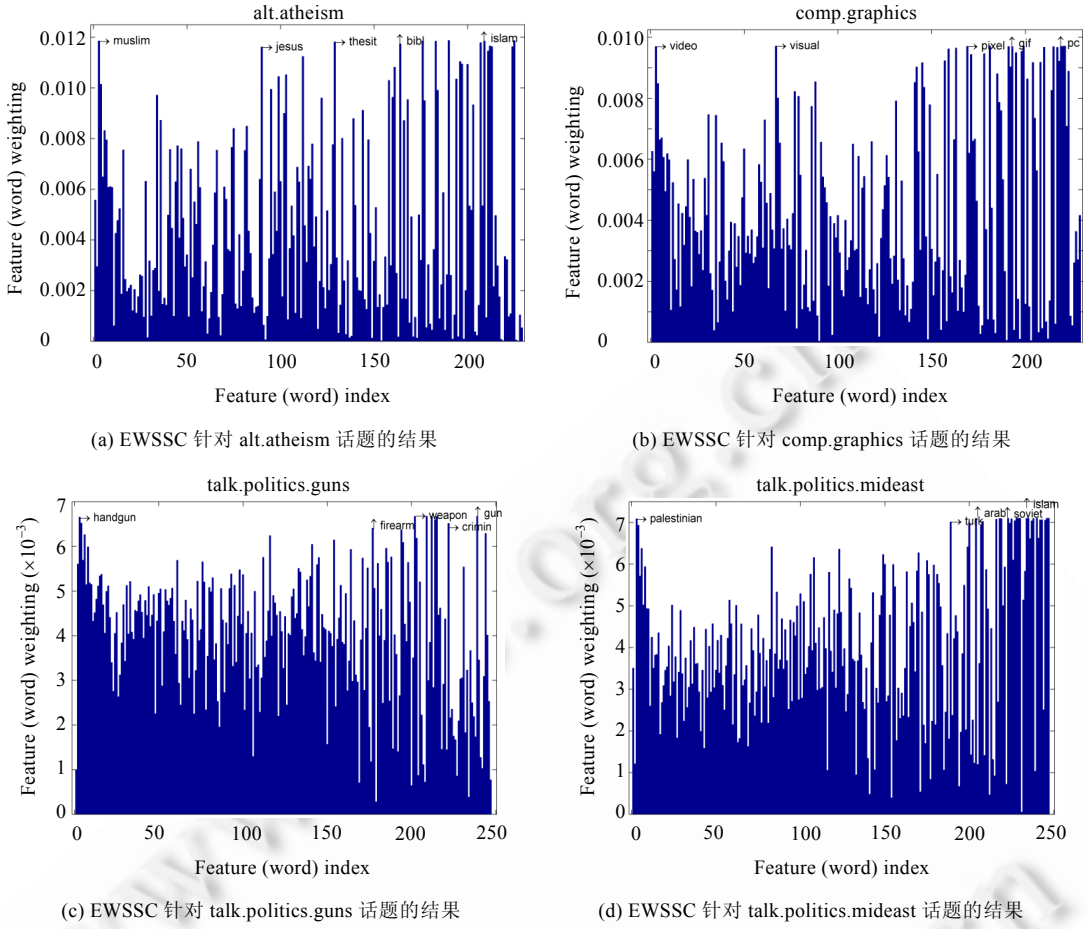
1) 在全部的 5 种文本数据流的实验中,EWSSC 和 EWSSC-fast 算法都得到了很好的实验结果,同样地,基于熵加权框架的软子空间聚类算法 EWSC 也可以进行良好的聚类划分.
2) EWSSC-fast 算法对后续到达的数据子块采用随机采样的方法,得到了与 EWSSC 算法近似一致的聚类划分结果,并且有效地减少了 EWSSC 算法的运行时间.
3) EWSSC 和 EWSSC-fast 算法获得了与 EWSC 算法几乎一致甚至更好的聚类结果,表明数据流软子空间聚类算法可以有效地处理流数据的聚类问题.
4) SOSKM 和 SPKM 算法的聚类结果虽然略低于 EWSSC,EWSSC-fast 和 EWSC 算法,但在大多数的文本数据集的实验中也获得了次好的划分结果;由此说明,基于球面 *K*-Means 框架的方向性聚类算法也适合于针对 tf-idf 模型的文本向量进行聚类分析.
5) 基于模糊加权理论的软子空间聚类算法 FWSC 并没有得到非常好的聚类结果,主要在于没有选择到合适的算法参数.我们将在后续的研究中进一步分析其针对高维数据流的参数选择问题.

此外,表 5~表 9 的最后一列给出了各种聚类算法所需的运行时间.容易看出,EWSSC-fast 算法利用采样技术减少了 EWSSC 算法的运行时间,从而有效地降低了 EWSSC 算法的时间复杂度.数据流聚类算法由于在聚类过程中利用了过去数据子块聚类划分得到的先验信息,减少了后续数据子块迭代收敛所需的运行次数,因此在文本数据集的实验中,比传统的批处理聚类算法节省了更多的运行时间.

3.4.2 不同文档集合代表的话题分析

进一步地,本节给出了针对 A2 和 B2 文本数据集,EWSSC 算法在数据子块大小为 20%时得到的各个数据簇词汇特征权值的分布情况.如图 5 所示,图中水平坐标表示 A2 和 B2 文本数据集的全部词汇特征序列,垂直坐标表示利用 EWSSC 算法得到的各个词汇特征的权值大小.由此可见,EWSSC 算法可以方便地获得各个数据簇最重要的词汇特征子集,并利用这些重要词汇特征集合得到该数据簇的相关主题.例如,在 alt.atheism 无神论的话题中,muslim,jesus,thesit,bibl 和 islam 等词汇特征得到了较高的权值;而在 comp.graphics 计算机图形学的话题中,video,visual,pixel,gif 和 pc 等词汇特征得到了较高的权值.因此,通过 EWSSC 算法得到各个数据簇词汇特征的权值,可以更好地理解各个文档集合所表示的话题.

总而言之,通过上述针对人造数据集和真实数据集的实验对比和分析,我们发现:结合模糊可扩展聚类策略和熵加权软子空间聚类方法,EWSSC 算法得到了与现有批处理软子空间聚类算法近乎一致的聚类结果.同时,实验结果也表明了 EWSSC 算法在流数据的聚类问题中可以得到很好的应用.



(a) EWSSC 针对 alt.atheism 话题的结果 (b) EWSSC 针对 comp.graphics 话题的结果

(c) EWSSC 针对 talk.politics.guns 话题的结果 (d) EWSSC 针对 talk.politics.mideast 话题的结果

Fig.5 Feature weighting distributions of keywords of text dataset A2 and B2 by EWSSC

图 5 EWSSC 算法对于 A2 和 B2 文本数据集所获得的关键词特征权值分布情况

4 结束语

在实际应用中,通常需要针对高维数据流或大规模数据进行聚类.为此,文本利用模糊可扩展聚类框架与现有的软子空间聚类算法相结合,提出了熵加权流数据软子空间聚类算法 EWSSC.该算法不仅可以准确地检测高维数据的局部子空间结构,而且可以利用模糊可扩展聚类策略,有效地解决流数据的聚类问题.本文在人造数据集和真实数据集的测试结果表明了 EWSSC 算法的有效性.

在后续的研究工作中,我们将围绕以下几个方面展开研究.比如,将对软子空间聚类算法的参数选择进行更加详细的讨论,提出针对高维数据流的比较合理和有效的参数选择方法;同时,将针对数据子块大小的选择问题进行更加详细的研究,权衡数据子块大小和最终聚类性能的联系,并进一步展开相关方面的理论和实验说明.我们还将利用其他更有效的采样技术或者近似推理方法来降低基于数据流的软子空间聚类算法的计算复杂度,从而更好地在实际问题中展开应用.

致谢 在此,我们向对本文的工作给予支持和建议的同行表示感谢.同时,对审稿人提出的有益建议表示感谢.

References:

[1] Jain AK. Data clustering: 50 years beyond *K*-means. Pattern Recognition Letters, 2010,31(8):651−666. [doi: 10.1016/j.patrec.2009.09.011]

[2] Sun JG, Liu J, Zhao LY. Clustering algorithms research. Run Jian Xue Bao/Journal of Software, 2008,19(1):48−61 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/48.htm [doi: 10.3724/SP.J.1001.2008.00048]

[3] Zhu L, Chung FL, Wang S. Generalized fuzzy *C*-means clustering algorithm with improved fuzzy partitions. IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, 2009,39(3):578−591. [doi: 10.1109/TSMCB.2008.2004818]

[4] Zhang M, Yu J. Fuzzy partitional clustering algorithms. Run Jian Xue Bao/Journal of Software, 2004,15(6):858−868 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/858.htm

[5] Muller E, Gunnemann S, Assent I, Seidl T. Evaluating clustering in subspace projections of high dimensional data. In: Proc. of the VLDB Endowment. 2009. 1270−1281.

[6] Kriegel HP, Kröger P, Zimek A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans. on Knowledge Discovery from Data, 2009,3(1):1−58. [doi: 10.1145/1497577.1497578]

[7] Parsons L, Haque E, Liu H. Subspace clustering for high dimensional data: A review. ACM SIGKDD Explorations Newsletter, 2004,6(1):90−105. [doi: 10.1145/1007730.1007731]

[8] Wang J, Wang ST, Deng ZH. A novel text clustering algorithm based on feature weighting distance and soft subspace learning. Chinese Journal of Computers, 2012,35(8):1655−1665 (in Chinese with English abstract).

[9] Deng ZH, Choi KS, Chung FL, Wang S. Enhanced soft subspace clustering integrating within-cluster and between-cluster information. Pattern Recognition, 2010,43(3):767−781. [doi: 10.1016/j.patcog.2009.09.010]

[10] Chen LF, Guo GD, Jiang QS. Adaptive algorithm for soft subspace clustering. Run Jian Xue Bao/Journal of Software, 2010,21(10):2513−2523 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3763.htm [doi: 10.3724/SP.J.1001.2010.03763]

[11] Gan G, Wu J. A convergence theorem for the fuzzy subspace clustering (FSC) algorithm. Pattern Recognition, 2008,41(6):1939−1947. [doi: 10.1016/j.patcog.2007.11.011]

[12] Jing L, Ng MK, Huang JZ. An entropy weighting *k*-means algorithm for subspace clustering of high-dimensional sparse data. IEEE Trans. on Knowledge and Data Engineering, 2007,19(8):1026−1041. [doi: 10.1109/TKDE.2007.1048]

[13] Jing L, Ng MK, Xu J, Huang JZ. Subspace clustering of text documents with feature weighting *k*-means algorithm. Advances in Knowledge Discovery and Data Mining, 2005,1(1):802−812. [doi: 10.1007/11430919_94]

[14] Gan G, Wu J, Yang Z. A fuzzy subspace algorithm for clustering high dimensional data. Advanced Data Mining and Applications, 2006,1(1):271−278. [doi: 10.1007/11811305_30]

[15] Guha S, Meyerson A, Mishra N, Motwani R, O'callaghan L. Clustering data streams: Theory and practice. IEEE Trans. on Knowledge and Data Engineering, 2003,15(3):515−528. [doi: 10.1109/TKDE.2003.1198387]

[16] Zhong S. Efficient streaming text clustering. Neural Networks, 2005,18(5-6):790−798. [doi: 10.1016/j.neunet.2005.06.008]

[17] Hall LO, Goldgof DB. Convergence of the single-pass and online fuzzy *C*-means algorithms. IEEE Trans. on Fuzzy Systems, 2011,19(4):792−794. [doi: 10.1109/TFUZZ.2011.2143418]

[18] Hore P, Hall LO, Goldgof DB. Creating streaming iterative soft clustering algorithms. In: Proc. of the Annual Meeting of the North American Fuzzy Information. 2007. 484−488. [doi: 10.1109/NAFIPS.2007.383888]

[19] Hore P, Hall LO, Goldgof DB. Single pass fuzzy *C*-means. In: Proc. of the IEEE Int'l Fuzzy Systems Conf. 2007. 1−7. [doi: 10.1109/FUZZY.2007.4295372]

[20] Bradley PS, Fayyad U, Reina C. Scaling clustering algorithms to large databases. In: Proc. of the Knowledge Discovery and Data Mining. 1998. 9−15.

[21] Farnstrom F, Lewis J, Elkan C. Scalability for clustering algorithms revisited. ACM SIGKDD Explorations Newsletter, 2000,2(1):51−57. [doi: 10.1145/360402.360419]

[22] Domeniconi C, Gunopulos D, Ma S, Yan B, Al-Razgan M, Papadopoulos D. Locally adaptive metrics for clustering high dimensional data. Data Mining and Knowledge Discovery, 2007,14(1):63−97. [doi: 10.1007/s10618-006-0060-8]

[23] Chan EY, Ching WK, Ng MK, Huang JZ. An optimization algorithm for clustering using weighted dissimilarity measures. Pattern Recognition, 2004,37(5):943−952. [doi: 10.1016/j.patcog.2003.11.003]

[24] Zhong S. Efficient online spherical *k*-means clustering. In: Proc. of the IEEE Int'l Joint Conf. on Neural Networks. 2005. 3180−3185. [doi: 10.1109/IJCNN.2005.1556436]

[25] Banerjee A, Ghosh J. Frequency-Sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres. IEEE Trans. on Neural Networks, 2004,15(3):702−719. [doi: 10.1109/TNN.2004.824416]

[26] Aggarwal CC, Han J, Wang J, Yu PS. A framework for clustering evolving data streams. In: Proc. of the Int'l Conf. on Very Large Data Bases. 2003. 81−92. [doi: 10.1016/B978-012722442-8/50016-1]

[27] Zhang YJ, Liu ZQ. Self-Splitting competitive learning: A new on-line clustering paradigm. IEEE Trans. on Neural Networks, 2002, 13(2):369−380. [doi: 10.1109/72.991422]

[28] Xu L, Krzyzak A, Oja E. Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. IEEE Trans. on Neural Networks, 1993,4(4):636−649. [doi: 10.1109/72.238318]

[29] Wei L M, Xie WX. Rival checked fuzzy *C*-means algorithm. Acta Electronica Sinica, 2000,28(7):63−66 (in Chinese with English abstract).

[30] Arthur D, Vassilvitskii S. *K*-Means++: The advantages of careful seeding. In: Proc. of the 18th Annual ACM-SIAM Symp. on Discrete Algorithms. 2007. 1027−1035.

[31] Cheng TW, Goldgof DB, Hall LO. Fast fuzzy clustering. Fuzzy Sets and Systems, 1998,93(1):49−56. [doi: 10.1016/S0165-0114(96)00232-1]

[32] Zhong S, Ghosh J. A unified framework for model-based clustering. Journal of Machine Learning Research, 2003,4(1):1001−1037.

[33] Dhillon IS, Modha DS. Concept decompositions for large sparse text data using clustering. Machine Learning, 2001,42(1):143−175. [doi: 10.1023/A:1007612920971]

[34] Bezdek JC. Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum Press, 1981.

[35] Nguyen N, Caruana R. Consensus clusterings. In: Proc. of the Int'l Conf. on Data Mining. 2007. 607−612. [doi: 10.1109/ICDM.2007.73]

[36] Strehl A, Ghosh J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research, 2003,3(1):583−617.

[37] Rand WM. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 1971,66(1):846−850. [doi: 10.2307/2284239]

[38] Iam-On N, Boongoen T, Garrett S, Price C. A link-based approach to the cluster ensemble problem. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2011,33(9):2396−2409. [doi: 10.1109/TPAMI.2011.84]

[39] 20-Newsgroups corpus. 1999. http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

[40] Mccallum AK. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. 1996. http://www.cs.cmu.edu/mccallum/bow

附中文参考文献:

[2] 孙吉贵,刘杰,赵连宇.聚类算法研究.软件学报,2008,19(1):48−61. http://www.jos.org.cn/1000-9825/19/48.htm [doi: 10.3724/SP.J.1001.2008.00048]

[4] 张敏,于剑.基于划分的模糊聚类算法.软件学报,2004,15(6):858−868. http://www.jos.org.cn/1000-9825/15/858.htm

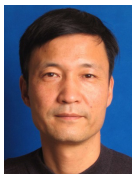
[8] 王骏,王士同,邓赵红.特征加权距离与软子空间学习相结合的文本聚类新方法.计算机学报,2012,35(8):1655−1665.

[10] 陈黎飞,郭躬德,姜青山.自适应的软子空间聚类算法.软件学报,2010,21(10):2513−2523. http://www.jos.org.cn/1000-9825/3763.htm [doi: 10.3724/SP.J.1001.2010.03763]

[29] 魏立梅,谢维信.对手抑制式模糊 *C*-均值算法.电子学报,2000,28(7):63−66.

朱林(1983－),男,安徽安庆人,博士,讲师,主要研究领域为数据挖掘,人工智能及应用.
E-mail: cslinzhu@gmail.com

毕忠勤(1977－),男,博士,副教授,主要研究领域为不确定数据管理,云计算,符号计算.
E-mail: zqbi@shiep.edu.cn



雷景生(1966－),男,博士,教授,主要研究领域为数据库与数据挖掘,智能 Web 信息处理技术.
E-mail: jshlei@126.com

杨杰(1964－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为模式识别,智能系统.
E-mail: jieyang@sjtu.edu.cn