一种集成簇内和簇间距离的加权 *k*-means 聚类方法

黄晓辉 王 成 熊李艳 曾 辉
(华东交通大学信息工程学院 南昌 330013)

摘 要 聚类分析是数据挖掘与分析最重要的方法之一.它把相似的数据对象归类到一个簇,把不同的数据对象尽可能分到不同的簇.其中 *k*-means 聚类算法,由于其简单性和高效性,被广泛运用于解决各种现实问题,例如文本演化分析、图像聚类、社区发现等.然而在聚类过程中,大部分现有的类 *k*-means 算法主要考虑簇内距离,而忽略了簇间距离的作用.本文结合特征加权方法,提出了一种新的集成簇内和簇间距离的加权 *k*-means 方法(a weighting *k*-means clustering approach by integrating Intra-Cluster and Inter-Cluster distances,KICIC)来解决高维数据聚类问题.虽然现有少数类 *k*-means 算法通过最大化簇中心与全局中心距离来融入簇间信息,但不同于这类方法,KICIC 通过在子空间内最大化簇中心与其他簇数据对象的距离来融合簇内和簇间距离进行聚类.基于此思路,本文首先为 KICIC 算法设计了一个目标函数,然后通过优化求解目标函数得到算法参数的更新迭代公式,并在此基础上设计了 KICIC 算法.最后,在 6 个真实数据集上的实验结果表明,对比现有类 *k*-means 算法,KICIC 算法在大部分情况下都有获得更好的聚类结果.

关键词 *k*-means;聚类分析;特征加权;熵调整;数据挖掘
中图法分类号 TP181 **DOI 号** 10.11897/SP.J.1016.2019.02836

A Weighting *k*-Means Clustering Approach by Integrating Intra-Cluster and Inter-Cluster Distances

HUANG Xiao-Hui WANG Cheng XIONG Li-Yan ZENG Hui
(*School of Information Engineering, East China Jiaotong University, Nanchang* 330013)

Abstract Clustering analysis is one of the most important methods for data mining and analysis. It aims at partitioning a data set into clusters such that the objects in a cluster have high similarity and the objects in different clusters are low similarity. Namely, the higher the similarity among the objects, the greater the probability that the objects are divided into the same cluster. Due to the simplicity and effectiveness, *k*-means clustering algorithm is widely used to solve various problems in real applications, such as text evolutionary analysis, image clustering, community detection, etc. However, most existing *k*-means type clustering methods consider only intra-cluster compactness and overlook inter-cluster separation, which results in a kind of deterioration of performance in some cases. Some existing methods measure the inter-cluster separation by computing the sum of the distances between the global center and the center of each cluster. However, bad results may occur in these algorithms when the global center locates in a certain cluster, e.g., the number of objects is totally imbalance in each cluster. Hence, how to make the best use of inter-cluster separation is a key challenge in the clustering process. By incorporating the subspace techniques with feature weighting methods, this paper proposes a novel *k*-means

收稿日期:2018-06-08;在线出版日期:2019-05-22.本课题得到国家自然科学基金(61562027)、江西省社会科学"十二五"(2015 年)规划项目(15XW12)、江西省自然科学基金项目(20181BAB202024,20192ACBL21006)、江西省教育厅项目(GJJ170413,GG170379)资助.
黄晓辉,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为数据挖掘、机器学习.E-mail:hxh016@hotmail.com.**王 成**(通信作者),硕士研究生,主要研究方向为数据挖掘、机器学习.E-mail:wangchengo@126.com.**熊李艳**,教授,中国计算机学会(CCF)会员,主要研究领域为数据库系统、自然语言处理.**曾 辉**,副教授,主要研究方向为数据挖掘、机器学习.

type method (KICIC) which is able to integrate intra-cluster compactness and inter-cluster separation to solve the clustering problem of high-dimensional data. Different to the existing algorithms that maximize the sum of the distances between the centers of clusters and the global center as inter-cluster separation, our proposed method maximizes inter-cluster separation by maximizing the sum of the distances between the centers of clusters and the data objects of other clusters in subspaces. Specifically, the inter-cluster separation in KICIC is to maximize the distances between the center of each cluster and the data objects which are not belong to this cluster in subspaces. Motivated by this idea, we firstly develop an objective function for KICIC by integrating intra-cluster compactness and inter-cluster separation. Then, the iterative rules of KICIC are derived by optimizing the objective function using Lagrange multiplier method. Based on the iterative rules, we design the KICIC algorithm. Subsequently, the convergence and complexity analysis of the algorithm are investigated analytically. At last, we analyze the clustering results, the convergence speed and the robustness of the proposed algorithm in experiments. The experimental results on six real data sets show that KICIC outperforms state-of-the-art k-means type algorithms in most of cases with respects to four metrics: *Accuracy* (*Acc*), *Adjust Rand Index* (*ARI*), *Fscore* (*Fsc*) and *Normal Mutual Information* (*NMI*). For example, on the dataset banknote, the proposed method KICIC achieves the best results in four metrics. On *Acc*, *ARI*, *Fsc* and *NMI*, the results of KICIC are at least 4%, 6%, 5%, 5% higher than these of the traditional clustering algorithms which do not consider the inter-cluster separation in the clustering process. Compared with the methods integrating inter-cluster separation, our proposed method also achieves 2%, 8%, 4% and 7% *Acc*, *ARI*, *Fsc* and *NMI* improvement, respectively. In order to conquer the clustering problem of multi-cluster structure in high-dimensional datasets, we plan to modify the objective function of KICIC by adding more constraint conditions to overcome this obstacle.

Keywords　k-means; clustering analysis; feature weighting; entropy regularization; data mining

1　引　言

在现实世界中高维数据无处不在，例如图像数据[1]、文本数据[2]、生物信息数据[3]等. 有效地分析这些高维数据往往会产生大量的高价值知识. 然而在高维数据中，往往存在大量的冗余和噪音信息，这将导致很多算法不能够获得很好的聚类性能. 高维数据的簇结构往往隐藏在相对低维的空间结构中. 因此，如何有效地降低数据冗余、去除噪音信息已成为挖掘高维数据簇结构的关键.

聚类分析是一类用于发现数据集中簇结构的无监督学习方法. 该类方法通常通过同时最小化簇内距离和最大化簇间距离来发现数据集中的簇结构. 聚类分析算法大致可以分为基于划分的方法、基于层次的方法、基于密度的方法、基于网格的方法和基于模型的方法[4]. 其中，k-means 算法是一种简单而有效的基于划分的聚类方法. 该方法通过两个迭代步骤进行聚类：(1) 根据现有的数据对象划分计算簇中心；(2) 根据新的簇中心与数据对象的相似度，重新划分数据对象的簇结构. 该方法的基本思想是把数据对象分成若干个簇，使得簇内距离尽量小，但是忽略了最大化簇间距离. 为了提高聚类性能，研究人员通过最大化每个簇中心与全局中心的距离来最大化簇间距离，如算法 ESSC[5]. 然而，最大化每个簇中心与全局中心的距离并不等价于最大化簇间距离，即最大化任意两个簇之间的距离. 事实上，当数据集中部分簇相距较近，即边界较模糊时，通过最大化每个簇中心与全局中心来最大化簇间距离可能会导致更差的结果. 如图 1 所示，图中有 4 个簇，分别为 C_1，C_2，C_3，C_4，它们的簇中心分别为 V_1，V_2，V_3，V_4，$global\_center$ 为全局中心. 用传统的方法最大化簇间距离可能导致簇中心 V_2 偏向 V_2' 的位置，因为 V_2' 比 V_2 离全局中心更远. 这样使得簇 C_1 中心和簇 C_2 的中心更加靠近，降低 C_1 和 C_2 的聚类正确率. 本文提出一种新的最大化簇间距离的方法. 该方法

通过在子空间内最大化每个簇中心与不属于本簇数据对象的距离来最大化簇间距离，这样可以真正使得任意两个簇中心之间相互远离，达到最大化簇间距离的效果，从而提高聚类性能.

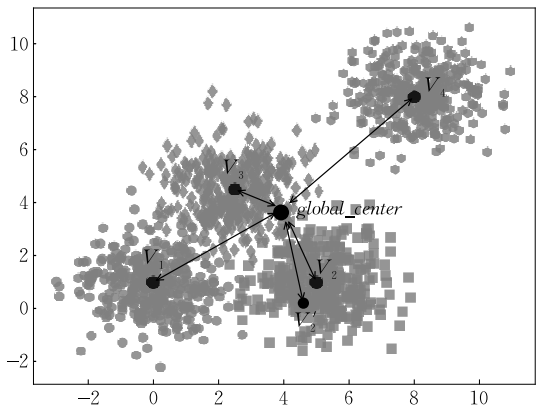

图 1 簇间距离图

此外，传统 k-means 算法平等地利用数据中所有的特征进行聚类分析. 事实上，在高维数据聚类中，有些特征具有较重要的作用，而有些特征的作用不大甚至是噪音特征. 传统的 k-means 方法不能有效地区分出不同特征在聚类中所起的作用. 例如在对新闻文本进行聚类时，不同类型主题的新闻对应的主要特征维度肯定不同，如体育主题中，篮球、足球、田径等关键词比较重要，而在财经主题中，股票、经济、财政等关键词比较重要. 传统 k-means 方法不能够有效地区分不同关键词在聚类过程中的作用，因此，聚类效果往往不好.

结合特征加权方法，本文提出了一种能够同时最小化簇内距离和最大化簇间距离的加权 k-means 聚类方法：KICIC 算法. 现有能够集成簇内和簇间距离的算法主要是通过最大化簇中心与全局中心的距离进行聚类. 不同于现有方法，KICIC 通过在子空间内最大化每个簇中心与其他簇数据对象的距离来融合簇内和簇间距离进行聚类. 基于此思路，本文首先为 KICIC 算法设计了一个目标函数，然后通过优化该目标函数得到了算法的更新迭代公式. 在更新迭代公式的基础上，本文设计了一个迭代算法：KICIC，并证明了算法的收敛性. 最后，本文在 6 个真实数据集上对比了 KICIC 算法与现有类 k-means 算法的性能，证明 KICIC 算法在大部分情况下都能获得更好的性能. 本文的主要贡献如下：

(1) 本文提出了一种新的 k-means 聚类框架，在该框架中提出了一种新的最大化簇间距离的方法. 与现有通过最大化每一个簇中心与全局中心的距离来最大化簇间距离的方法不同，本文所提出的最大化簇间距离的方法是在子空间内最大化每一个簇中心与其他簇中数据对象的距离.

(2) 结合传统特征加权 k-means[6-7] 和新的聚类框架，本文为新算法(KICIC)设计了一个目标函数，该目标函数能够有效地结合簇内和簇间距离，使得每个簇中心不但能代表本簇的数据对象，也能尽可能远离不属于本簇的数据对象.

(3) 通过优化求解目标函数，本文获得了 KICIC 算法的迭代更新规则，并证明了算法的收敛性.

(4) 在真实数据集的实验上，研究了算法参数的选择，最后通过对比实验表明 KICIC 算法能够在大部分情况下获得更好聚类结果.

本文在第 2 节中对各种类 k-means 聚类算法进行简要的综述；在第 3 节中提出 KICIC 算法的目标函数，并给出算法的迭代更新规则及其证明；在第 4 节中，本文研究 KICIC 算法的参数选择，并对比各种不同版本的 k-means 算法性能；最后第 5 节总结本文的工作.

2 相关工作

目前，类 k-means 聚类算法可以简要地分成两类：经典 k-means 聚类算法和特征加权 k-means 聚类算法. 本节从这两个方面进行相关文献的综述. 最后在分析相关工作的基础上，总结本文所提出方法的特点. 对于更多与 k-means 和聚类相关的方法介绍，可参考综述文献[8-10].

2.1 经典 k-means 聚类算法

尽管经典 k-means 算法被提出已经超过 60 年[11]，但它仍然是在现实世界中运用最广泛的聚类算法之一. 经典 k-means 算法通过迭代更新簇中心和数据对象分配矩阵来获得数据的聚类结构. 在运行算法之前，需要通过人工指定两方面的参数：初始化中心和簇数目. k-means 通常只能保证找到局部最优解，初始化中心的选择将影响算法最终找到最优解的好坏. 因此，一些研究人员提出不同的方案来寻找更好的初始化中心[12-16]. Arthur 和 Vassilvitskii 提出了一种 k-means++ 算法[12]，该方法选取初始中心的原则是不同初始中心之间的距离尽量大，这样可以使得初始中心尽量分布在不同的簇中. 为了提高 k-means++ 算法在大规模数据中选择中心的效率，Bachem 等人利用马尔可夫蒙特卡洛方法提出了一种快速的近似 k-means++ 的初始中心选择

方法.但是该方法需要假设数据服从某一种分布，然而，在现实应用中这种假设并不一定成立.在文献[16]中，Bachem 等人在不需要任何数据分布假设的前提下提出了一种快速获得初始中心的方法.同时，该方法可以扩展到任意距离度量上.在文献[15]中，Shahnewaz 等人通过凸包(Convex hull)理论估计局部数据的密度来获得初始化簇中心.

此外，在现实应用中数据集中的簇数目也很难提前获得.因此，很多研究人员提出各种自适应的方法来获得簇数目[17-18].在文献[17]中，Dan 和 Moore 提出通过优化某种准则，如赤池信息量准则(Akaike Information Criterion)、贝叶斯信息准则(Bayesian Information Criterion)，来自动地获得簇数目.在文献[18]中，Tibshirani 等人利用间隔统计量(Gap Statistic)来估计数据集中簇数目，该方法不仅可以用来帮助 k-means 估计簇数目，还可以用于其它算法中，如层次聚类.

不同的相似度度量方法也会衍生出不同的 k-means 算法.传统 k-means 算法通常采用欧氏距离作为度量，然而在一些现实应用中，欧氏距离度量效果并不理想.例如在文本分析中，Cos 距离度量(球形 k-means[19]或椭球 k-means[20])通常比欧氏距离度量效果好.在文献[21]中，Kashima 等人提出基于曼哈顿距离的 k-means 算法用于聚类比例数据.

经典 k-means 算法通常在聚类过程中平等地对待所有的特征.然而在现实应用中，不同特征在聚类过程中的重要性可能不同，平等对待所有特征将会降低聚类算法的性能.

2.2　特征加权 k-means 聚类算法

由于传统的 k-means 算法无法区分不同特征在聚类过程中的重要性，因此研究人员提出了不同形式的特征加权方法来衡量不同特征在聚类过程中的重要性[6-7,22].在文献[6]中，Huang 等人提出了一种自动特征加权方法(WKME).该方法在整个数据集中为每一个特征计算一个权重，用来代表该特征在聚类过程中的重要性.然而在现实应用中，相同的特征在不同的簇中可能具有不同的作用.因此，研究人员提出了矩阵特征加权方法[7,22]，即为同一特征在不同的簇中计算不同的权重.文献[22]中，Chan 等人采用与文献[6]相同的特征加权方式为每一个簇计算一个特征向量.在文献[7]中，Jing 等人提出了一种基于熵调整的加权 k-means 算法(EWKM)来聚类高维稀疏数据，该方法利用熵调整让更多的特征获得权重.在文献[23]中，Huang 等人提出了一种基于 $L2$-Norm 调整的矩阵加权方法，该方法通过 $L2$ 调整可以使得部分噪音特征的权重为零，从而排除其在聚类中的作用.在文献[24]中，王骏等人将特征加权与软子空间相结合，提出了一种用于高维文本数据的软子空间模糊聚类新方法.为了提高传统 k-means 聚类算法的聚类准确性，王宏杰等人在文献[25]中提出一种结合初始中心优化和特征加权的改进 k-means 聚类算法.在文献[26]中，提出了一种最小最大加权 k-means 方法，该方法根据簇中数据对象的方差赋予不同簇不同的权重.

以上方法在聚类过程中，都只考虑了簇内距离而忽略了簇间距离.研究人员提出结合簇内和簇间距离聚类的 k-means 算法[5,27-28].在文献[5]中，Deng 等人提出结合簇内与簇间距离的软子空间聚类算法.通过扩展 EWKM[7]，Huang 等人提出了一种基于判别子空间的 k-means 聚类算法[28].在文献[27]中，Huang 等人提出了一个与线性判别分析[29]类似的框架.在此框架下，作者提出了三种类 k-means 聚类算法：无特征加权 k-means，向量特征加权 k-means 和矩阵特征加权 k-means.

2.3　新方法的特点

本文所提出的方法 KICIC 与 EWKM[7]、文献[22-23]中所提方法的特征加权方式类似，都是针对每个簇计算一个特征向量来表示簇中每个特征在聚类该簇所起的作用.然而这些方法在聚类过程中，只考虑了簇内距离而忽略簇间距离的作用.因此，本文在以上方法的基础之上，进一步集成簇间距离来提高算法的聚类性能.

ESSC[5]与文献[27-28]中所提方法也集成了簇内与簇间距离进行聚类.文献[5,27]通过最大化每个簇中心与全局中心的距离来最大化簇间距离，而在本文所提出的方法中不需要引入全局中心.文献[28]中，需要通过引入更多的参数来度量任意两个簇间的相对权重.以上方法最大化簇间距离的基本思想是最大化不同簇中心的距离，而本文所提出方法的思路是在当前特征子空间下，最小化簇内数据对象与簇中心距离的同时，最大化簇外数据对象与本簇中心的距离，这是与现有方法不同的思路.

3　基于簇内和簇间距离的加权 k-means 聚类算法(KICIC)

本节将详细介绍 KICIC 方法.首先，根据最小化簇内距离同时最大化簇间距离的思想设计了

KICIC 方法的目标函数.然后,通过最优化目标函数获得算法的更新规则,并证明了算法的收敛性.最后,根据更新规则给出了算法的执行过程.

3.1 目标函数

设 $X=\{X_1,X_2,\cdots,X_n\}$ 为一个包含 n 个数据对象的数据集,其中第 i 个数据对象表示为 $X_i=\{x_{i1},x_{i2},\cdots,x_{im}\}$,$m$ 为数据对象特征的数目.数据对象分配矩阵 $\boldsymbol{U}$ 是一个 $n\times k$ 的 0-1 矩阵,$u_{ip}=1$ 表示第 i 个特征被分到第 p 个簇.$\boldsymbol{Z}=\{\boldsymbol{Z}_1,\boldsymbol{Z}_2,\cdots,\boldsymbol{Z}_k\}$ 为 k 个簇中心向量,其中 $\boldsymbol{Z}_p=\{z_{p1},z_{p2},\cdots,z_{pm}\}$ 为第 p 个簇中心.$\boldsymbol{W}=\{\boldsymbol{W}_1,\boldsymbol{W}_2,\cdots,\boldsymbol{W}_k\}$ 为 k 个权重向量,其中 $\boldsymbol{W}_p=\{w_{p1},w_{p2},\cdots,w_{pm}\}$ 代表第 p 个簇的特征权重,w_{pj} 为第 p 个簇中第 j 个特征的权重.通过结合簇间距离与簇内距离的信息,KICIC 算法的目标函数可形式化为

$$\begin{aligned}P(\boldsymbol{W},\boldsymbol{U},\boldsymbol{Z})=&\sum_{p=1}^{k}\sum_{i=1}^{n}u_{ip}\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2-\\&\eta\sum_{p=1}^{k}\sum_{i=1}^{n}(1-u_{ip})\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2+\\&\gamma\sum_{p=1}^{k}\sum_{j=1}^{m}w_{pj}\log w_{pj}\end{aligned}\tag{1}$$

服从约束条件

$$\begin{cases}\sum_{p=1}^{k}u_{ip}=1,\ u_{ip}\in\{0,1\}\\ \sum_{j=1}^{m}w_{pj}=1,\ 0\le w_{pj}\le 1\end{cases}\tag{2}$$

其中,η 是权衡簇内簇间距离的参数,γ 是控制权重 $\boldsymbol{W}$ 分布的参数.在目标函数(1)中,第一项的目的是使得样本在特定子空间中簇内距离最小;第二项作用是使得簇间距离尽可能大,更具体地说,就是使得不属于簇 $p(1\le p\le k)$ 的样本在簇 p 所表示的子空间(即 W_P)中,离簇 p 的中心尽可能远.在该目标函数中,增加 η 的值将增加簇间距离在聚类过程中的作用,当 $\eta=0$ 时,簇间距离对聚类过程将不起作用,KICIC 算法将退化为 EWKM 算法[7];第三项是通过权重的熵进行特征权重分布的调整,增加 γ 的值将促使每个子空间中有更多的特征参与聚类过程.当 $\gamma=0$ 时,算法将只给具有最小簇内和簇间距离之差的特征赋值为 1,而其他特征的权重为 0;而当 γ 趋于无穷大时,所有特征的权重将趋同.

3.2 KICIC 算法

为了更方便地优化求解目标函数(1),我们首先将目标函数(1)转换为如下形式

$$\begin{aligned}P(\boldsymbol{W},\boldsymbol{U},\boldsymbol{Z})=&(1+\eta)\sum_{p=1}^{k}\sum_{i=1}^{n}u_{ip}\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2-\\&\eta\sum_{p=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2+\\&\gamma\sum_{p=1}^{k}\sum_{j=1}^{m}w_{pj}\log w_{pj}\end{aligned}\tag{3}$$

在目标函数(3)中,把数据对象分配矩阵 $\boldsymbol{U}$ 合并到一项中,以便求解.在该目标函数中,需要求解三个参数矩阵:数据对象分配矩阵 $\boldsymbol{U}$,簇中心矩阵 $\boldsymbol{Z}$ 和特征权重矩阵 $\boldsymbol{W}$.常用的优化求解目标函数 P 的方法是固定其中两个参数矩阵,求解第三个参数矩阵.

公式 1. 固定权重矩阵 $\boldsymbol{W}$ 和簇中心矩阵 $\boldsymbol{Z}$,$P(\boldsymbol{W},\boldsymbol{U},\boldsymbol{Z})$ 可以最小化当且仅当

$$u_{ip}=\begin{cases}1, & \sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2\ge\sum_{j=1}^{m}w_{p'j}(x_{ij}-z_{p'j})^2\\ 0, & \text{其他}\end{cases}\tag{4}$$

当固定权重矩阵 $\boldsymbol{W}$ 和簇中心矩阵 $\boldsymbol{Z}$ 时,目标函数(3)中的第二项和第三项与分配矩阵 $\boldsymbol{U}$ 无关,同时 η 也与 $\boldsymbol{U}$ 无关.因此在求解 $\boldsymbol{U}$ 时,目标函数(3)可简化为

$$P(\boldsymbol{W},\boldsymbol{U},\boldsymbol{Z})=\sum_{p=1}^{k}\sum_{i=1}^{n}u_{ip}\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2\tag{5}$$

固定 $\boldsymbol{W}$ 和 $\boldsymbol{Z}$,最优化目标函数(5)可得到式(4),详细的证明过程可参考文献[30-31].从直觉上来看,式(4)是把数据对象分配到带权距离最小的簇中.

公式 2. 固定数据对象分配矩阵 $\boldsymbol{U}$ 和特征权重矩阵 $\boldsymbol{W}$,$P(\boldsymbol{W},\boldsymbol{U},\boldsymbol{Z})$ 可以最小化当且仅当

$$z_{pj}=\frac{(1+\eta)\sum_{i=1}^{n}u_{ip}x_{ij}-\eta\sum_{i=1}^{n}x_{ij}}{(1+\eta)\sum_{i=1}^{n}u_{ip}-\eta n}\tag{6}$$

证明. 当固定 $\boldsymbol{U}$ 和 $\boldsymbol{W}$ 时,目标函数(3)可以简化为

$$\begin{aligned}P(\boldsymbol{W},\boldsymbol{U},\boldsymbol{Z})=&(1+\eta)\sum_{p=1}^{k}\sum_{i=1}^{n}u_{ip}\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2-\\&\eta\sum_{p=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2\end{aligned}\tag{7}$$

针对目标函数(7)关于 z_{pj} 求导得

$$\begin{aligned}\frac{\partial P(\boldsymbol{W},\boldsymbol{U},\boldsymbol{Z})}{\partial z_{pj}}=&2(1+\eta)\sum_{i=1}^{n}u_{ip}w_{pj}(z_{pj}-x_{ij})-\\&2\eta\sum_{i=1}^{n}w_{pj}(x_{ij}-z_{pj})^2\end{aligned}\tag{8}$$

令 $\frac{\partial P(\boldsymbol{W},\boldsymbol{U},\boldsymbol{Z})}{\partial z_{pj}}=0$,可求解得到式(6).

公式 3. 固定数据对象分配矩阵 $\boldsymbol{U}$ 和簇中心

矩阵 $\boldsymbol{Z}$，$P(\boldsymbol{W},\boldsymbol{U},\boldsymbol{Z})$可以最小化当且仅当

$$w_{pt}=\frac{\exp\left(\frac{-D_{pt}}{\gamma}\right)}{\sum_{j=1}^{m}\exp\left(\frac{-D_{pj}}{\gamma}\right)} \tag{9}$$

其中

$$D_{pt}=(1+\eta)\sum_{i=1}^{n}u_{ip}(x_{it}-z_{pt})^2-\eta\sum_{i=1}^{n}(x_{it}-z_{pt})^2 \tag{10}$$

证明.　基于拉格朗日乘数法得到如下无约束的优化问题：

$$\begin{aligned}\min F_1(w_{pj},\delta_p)=&\sum_{p=1}^{k}\Big[(1+\eta)\sum_{i=1}^{n}u_{ip}\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2-\\&\eta\sum_{i=1}^{n}\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2+\gamma\sum_{j=1}^{m}w_{pj}\log w_{pj}\Big]-\\&\sum_{p=1}^{k}\delta_p\Big(\sum_{p=1}^{m}w_{pj}-1\Big)\end{aligned} \tag{11}$$

其中，$[\delta_1,\cdots,\delta_k]$是一个包含所有拉格朗日乘子的向量；由于每个权重值 w_{pj} 之间都是相互独立的，所以该优化问题可以拆分为 k 个独立的问题进行求解：

$$\begin{aligned}\min F_{1p}(w_{pj},\delta_p)=&\Big[(1+\eta)\sum_{i=1}^{n}u_{ip}\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2-\\&\eta\sum_{i=1}^{n}\sum_{j=1}^{m}w_{pj}(x_{ij}-z_{pj})^2+\gamma\sum_{j=1}^{m}w_{pj}\log w_{pj}\Big]-\\&\delta_p\Big(\sum_{p=1}^{m}w_{pj}-1\Big)\end{aligned} \tag{12}$$

其中，$l=1,\cdots,k$；通过分别对 δ_p，w_{pj} 求偏导数，并令其为 0，可以得到如下等式：

$$\frac{\partial F_{1p}}{\partial\delta_p}=\sum_{j=1}^{m}w_{pj}-1=0 \tag{13}$$

$$\begin{aligned}\frac{\partial F_{1p}}{\partial w_{pt}}=&(1+\eta)\sum_{i=1}^{n}u_{ip}(x_{it}-z_{pt})^2-\eta\sum_{i=1}^{n}(x_{it}-z_{pt})^2-\\&\delta_p+\gamma+\gamma\log w_{pt}=0\end{aligned} \tag{14}$$

从式子(14)中可以得到

$$\begin{aligned}w_{pt}&=\exp\left(\frac{-D_{pt}+\delta_p-\gamma}{\gamma}\right)\\&=\exp\left(\frac{\delta_p-\gamma}{\gamma}\right)\exp\left(\frac{-D_{pt}}{\gamma}\right)\end{aligned} \tag{15}$$

其中，D_{pt} 是用来衡量在第 p 个簇中，样本点的第 t 个维度与中心点第 t 个维度之间的紧凑度. 同时将式子(15)代入到式(13)，可以到如下等式：

$$\begin{aligned}\sum_{j=1}^{m}w_{pj}&=\sum_{j=1}^{m}\exp\left(\frac{\delta_p-\gamma}{\gamma}\right)\exp\left(\frac{-D_{pj}}{\gamma}\right)\\&=\exp\left(\frac{\delta_p-\gamma}{\gamma}\right)\sum_{j=1}^{m}\exp\left(\frac{-D_{pj}}{\gamma}\right)=1\end{aligned} \tag{16}$$

进一步整理(16)，可以得到

$$\exp\left(\frac{\delta_p-\gamma}{\gamma}\right)=\frac{1}{\left[\sum_{j=1}^{m}\exp\left(\frac{-D_{pj}}{\gamma}\right)\right]} \tag{17}$$

将式(17)代入到式(16)可得

$$w_{pt}=\frac{\exp\left(\frac{-D_{pt}}{\gamma}\right)}{\sum_{j=1}^{m}\exp\left(\frac{-D_{pj}}{\gamma}\right)} \tag{18}$$

KICIC 算法的流程如算法 1. 该算法的主体框架由三个迭代步骤组成，分别求出数据对象分配矩阵 $\boldsymbol{U}$，簇中心 $\boldsymbol{Z}$ 与特征权重 $\boldsymbol{W}$. 重复迭代直到目标函数(1)的值不再降低或者分配矩阵 $\boldsymbol{U}$ 不再改变为止. 目标函数(1)的值不再降低或者分配矩阵 $\boldsymbol{U}$ 不再改变是等价的.

算法 1.　KICIC 算法.

输入：数据集 X，簇数目 k，参数 γ，η

输出：数据对象分配矩阵 $\boldsymbol{U}$，簇中心 $\boldsymbol{Z}$ 和特征权重 $\boldsymbol{W}$

初始化：随机初始簇中心 $\boldsymbol{Z}$ 和特征权重 $\boldsymbol{W}$

REPEAT

　　固定 $\boldsymbol{W}$ 和 $\boldsymbol{Z}$，利用式(4)求解数据对象分配矩阵 $\boldsymbol{U}$

　　固定 $\boldsymbol{W}$ 和 $\boldsymbol{U}$，利用式(6)求解簇中心 $\boldsymbol{Z}$

　　固定 $\boldsymbol{U}$ 和 $\boldsymbol{Z}$，利用式(9)求解特征权重 $\boldsymbol{W}$

UNTIL 目标函数(1)收敛或分配矩阵 $\boldsymbol{U}$ 不再改变

3.3　收敛性及复杂度分析

3.3.1　收敛性

当参数设置合理时，KICIC 算法将在有限次迭代后达到收敛. 由于将任意一堆样本点分成 k 簇的分法是有限的，也就是说分配矩阵 $\boldsymbol{U}$ 的情况是有限的；所以，对于每一种可能的分配矩阵 $\boldsymbol{U}^t$ 来说，在整个迭代过程中只可能出现一次，直到收敛.

假设 $\boldsymbol{U}^{t_1}=\boldsymbol{U}^{t_2}$，$t_1\neq t_2$，其中 t_i 表示第 i 次迭代，对于给定的 $\boldsymbol{U}^{t_i}$，通过式(6)能够得出 $\boldsymbol{Z}^{t_i}$；因此，对于 $\boldsymbol{U}^{t_1}$，$\boldsymbol{U}^{t_2}$，分别能够得出 $\boldsymbol{Z}^{t_1}$，$\boldsymbol{Z}^{t_2}$（且两者相等，因为 $\boldsymbol{U}^{t_1}=\boldsymbol{U}^{t_2}$）. 进一步可以通过式(9)可以得出，$\boldsymbol{W}^{t_1}$，$\boldsymbol{W}^{t_2}$（由于 $\boldsymbol{U}^{t_1}=\boldsymbol{U}^{t_2}$，$\boldsymbol{Z}^{t_1}=\boldsymbol{Z}^{t_2}$ 故二者同样相等）. 由此可以得到

$$P(\boldsymbol{U}^{t_1},\boldsymbol{Z}^{t_1},\boldsymbol{W}^{t_1})=P(\boldsymbol{U}^{t_2},\boldsymbol{Z}^{t_2},\boldsymbol{W}^{t_2}).$$

由于序列 $P(\cdot,\cdot,\cdot)$是算法严格单调递减而产生的，因此，算法 KICIC 将在有限次迭代之后收敛.

3.3.2　复杂度

与其他加权 k-means 算法类似，KICIC 算法也包含三个计算步骤：更新分配矩阵 $\boldsymbol{U}$、更新簇中心矩阵 $\boldsymbol{Z}$ 和更新特征权重矩阵 $\boldsymbol{W}$. 下面分别介绍更新三个矩阵所需要的计算复杂度.

(1) 更新数据对象分配矩阵. 在迭代计算 $\boldsymbol{U}$ 时，

对每个簇都需要计算其簇中心与每个样本点的带权距离，也即是计算如下公式的值

$$\sum_{j=1}^{m} w_{pj}(x_{ij}-z_{pj})^2.$$

因此，计算数据对象分配矩阵所需要的时间代价为 $O(mnk)$，其中 m 为数据对象特征数目，n 为数据集中数据对象的数目，k 为簇数目；

(2) 更新簇中心矩阵. 给定的分配矩阵 $\boldsymbol{U}$ 和特征权重 $\boldsymbol{W}$，根据式(6)，其所需要计算代价为 $O(mnk)$；

(3) 更新权重矩阵. 根据式(9)和(10)，其主要时间代价是针对每个数据对象，计算该数据对象与每个簇中心的距离，该步骤的时间复杂度同样为 $O(mnk)$.

综上所述，KICIC 算法单轮更新的时间复杂度为 $O(mnk)$，若整个收敛过程需要进行 h 次迭代，则算法的时间复杂度为 $O(hmnk)$.

4 实验结果与分析

本节将通过实验对比各算法在真实数据集上的性能. 接下来的这个部分，将分别对数据集的选择、实验的设置、评价指标的选择、参数的研究以及最终性能的表现和最大化簇间距离有效性的验证进行详细的阐述.

4.1 数据集选择

如图 2 所示，左右两个簇分别为 c_1，c_2. v_1，v_2 分别表示两个簇中心，在 c_2 中存在一个样本点 x_i，距离簇 c_1 的中心 v_1 最近，该距离表示为 D. 样本点 x_i 距离它所属簇 c_2 的中心 v_2 距离表示为 d. 当且仅当存在样本点 x_i，使得 $D<d$ 时，簇 c_1 与簇 c_2 相距较近，两簇的边界模糊(注：原始数据均经过标准化处理).

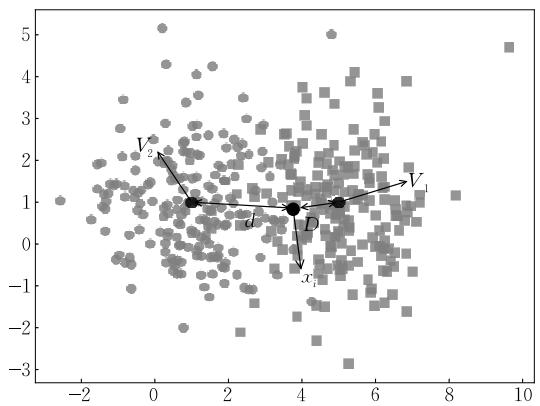

图 2 簇边界模糊图

根据表 1 可知，对于选择的 6 个数据集来说，相同点均存在有两个簇相距较近的情况，而不同点在于在不同的数据集中，簇之间的相交程度不一样. 同时，为了评价本文所提出方法的性能，我们将在 6 个真实数据集上进行比较研究，数据集的特征如表 2 所示，这些数据集可以从 UCI① 网站上下载.

表 1 簇间距离表

数据集	(D,d)
knowledge	(0.57,0.97)
waveform	(2.48,3.81)
wine	(2.58,3.56)
banknote	(1.01,1.27)
vertebra	(0.94,1.47)
messidor	(1.08,1.74)

表 2 数据集信息表

数据集名称	样本数	维度	簇数
banknote	1372	4	2
knowledge	403	5	4
vertebra	310	6	2
wine	178	13	3
messidor	1151	19	2
waveform	5000	21	3

4.2 实验设置

KICIC 方法是通过特征加权、集成簇内和簇间信息的方式，提出的一种改进 k-means 算法. 为了验证其性能，我们选择现有的各种 k-means 类型算法作为对比算法，包括经典 k-means 算法、WKME[6]、EWKM[7]、ESSC[5]，以及近年来最新发表的算法 AFKM[16]、MinMax[26]、SC[32]. 其中经典 k-means 算法不需要进行超参数的选择，其它算法都需要确定额外的超参数. 对于 WKME、AFKM、MinMax 和 SC 算法，此处均采用其论文中对应的默认参数. 对于 ESSC，EWKM，KICIC 算法，有一个共同的超参数 γ，该参数的作用是通过熵调整来改变特征权重的参数分布，本文将在后面小节来研究该参数的选择. 对于 ESSC 和 KICIC 算法，分别有参数 α 和 η 用来权衡簇内和簇间距离在聚类过程中所起的作用. 但是两种算法中簇间距离通过不同的方式影响聚类结果. 在 ESSC 中，簇间距离是通过每个簇中心与全局中心的距离来度量；而在 KICIC 中，簇间距离是通过在子空间内不属于本簇的数据对象离本簇中心的距离来度量. 本文将在 4.4 节对参数 γ，α 和 η 进行研究. 同时，在实验过程中，我们对所有的数据集均采取归一化的处理.

由于 k-means 类型的算法结果与初始化中心有关. 在实验中，我们首先随机初始化 100 个簇中心，然后用这 100 个簇中心初始化所有算法并进行聚

① http://archive.ics.uci.edu/ml/

类，最后对 100 次运行结果的平均性能指标和方差进行对比研究. 此外，初始化权重也对聚类结果有影响，在实验中，我们对不同的算法均采用随机的方式初始化权重.

4.3　评价指标

在本文中，我们采用聚类算法中常用的四种评价指标：*Accuracy*(*Acc*)，*AdjustRandIndex*(*ARI*)，*Fscore*(*Fsc*)以及 *Normal Mutual Information*(*NMI*). *Acc* 代表聚类结果的准确度，其取值范围为[0,1]；*ARI* 即调整兰德系数，其取值范围为[−1,1]；*Fscore* 是召回率和准确率的加权调和平均，其取值范围为[0,1]；*NMI* 即归一化互信息，用来衡量两个数据集分布的吻合程度，其取值范围为[0,1]. 上述四种评测指标都是值越大，说明聚类结果越好. 这四种指数的具体计算方法可参考文献[28].

4.4　参数研究

4.4.1　超参数 γ

在 ESSC、EWKM、KICIC 三个算法中，都采用了熵调整方法来增加特征在聚类过程中的参与度，并用参数 γ 来调节熵在目标函数中的作用. 由于 EWKM 只有 γ 一个参数，并且 ESSC 和 KICIC 算法都是从 EWKM 扩展而来. 因此我们通过 EWKM 算法来研究获得 γ 的取值. 图 3 到图 6 分别展示了 EWKM 算法在 6 个数据集上的 *Accuracy*、*Fscore*、*ARI* 和 *NMI* 的趋势图.

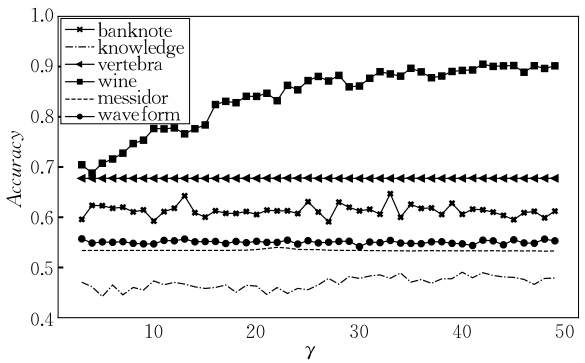

图 3　评价指标 *Accuracy* 随 γ 的变化趋势

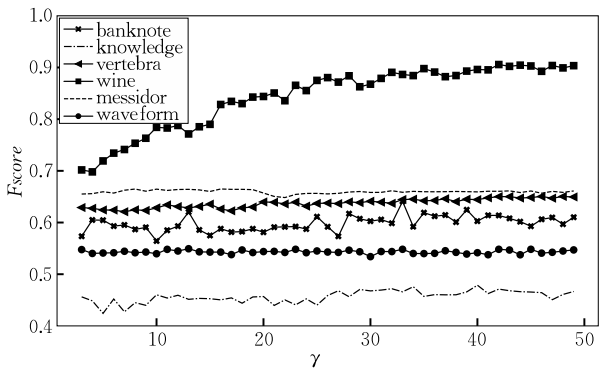

图 4　评价指标 *Fscore* 随 γ 的变化趋势

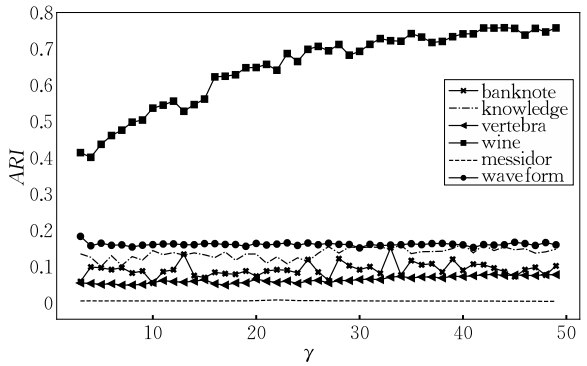

图 5　评价指标 *ARI* 随 γ 的变化趋势

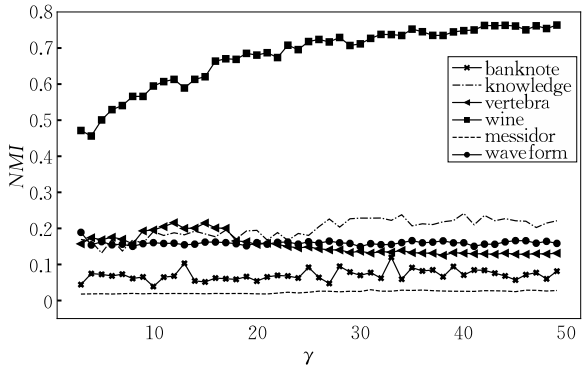

图 6　评价指标 *NMI* 随 γ 的变化趋势

从图 3 到图 6 中，我们可以发现 EWKM 在大部分数据集上对参数 γ 不是很敏感，例如数据集 banknote、knowledge、vertebra、messidor 和 waveform. 而在数据集 wine 上，EWKM 的性能随 γ 值的增加而提高. 当 $\gamma \geq 40$ 时，性能趋于稳定.

对于 banknote 数据集来说，在 *Accuracy* 和 *Fscore* 这两项指标中，随着 γ 的增大，其值主要是在 0.6 附近波动；在 *ARI* 和 *NMI* 这两项指标中，其值分别在 0.05 和 0.1 之间波动. 对于数据集 knowledge，在 *Accuracy* 和 *Fscore* 这两项指标中，其值主要是在 0.45 附近波动；在 *ARI* 和 *NMI* 这两项指标中，其值分别在 0.15 和 0.2 附近波动. 也就是说，banknote 和 knowledge 这两个数据集在随着 γ 增大的时候，虽然四个指标有着一定的波动，但总体来说波动范围不大，γ 任取其中一个值影响都不太大. 但是对于数据集 wine 来说，随着 γ 的不断增大，四个指标的变化幅度均超过了 30%.

从上面的图 3 到图 6 我们可以看到，当 γ 的取值大于 40 之后，随着 γ 的增大，EWKM 在数据集 wine 上的性能，不再有明显的波动. 因此，通过上面的研究分析，我们最终认为 $\gamma=40$ 对所有的数据集是一个比较合理的取值.

4.4.2 超参数 α

在算法 ESSC 中，超参数 α 是用来权衡簇内和簇间距离在聚类过程中所起的作用，其簇间距离是通过最大化每个簇中心与全局中心的距离来体现. 聚类主要依靠最小化簇内距离来形成各个簇，簇间距离只是起到一个调节的辅助作用，如果其作用太大，将会导致目标函数发散. 因此我们在选取参数 α 时，要避免其过大.

在通过 EWKM 算法确定好 γ 后，我们就可以针对 ESSC 研究参数 α 的取值. 在 ESSC 算法中，因为 α 值越小，算法对其敏感程度越大，所以我们通过在不同的区间段为 α 设置不同的步长来观察 α 在四个指标上对算法性能的影响. 如图 7 到图 10 所示，当 α 在[0.01，0.1]这个区间段时，其步长设置为 0.002；当 α 在(0.1，0.3]这个区间段时，其步长设置为 0.02. 同时根据图 7 到图 10 还可以发现，随着 α 的取值越来越大，在某些数据集上，算法已经开始出现了发散或者很难收敛的现象，这也说明簇结果主要由簇内聚类决定.

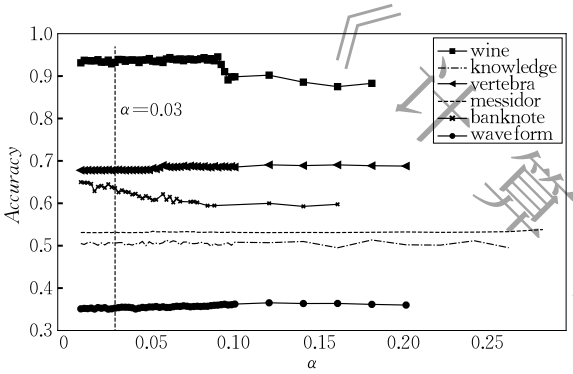

图 7 各数据集聚类 *Accuracy* 指标随 α 变化图

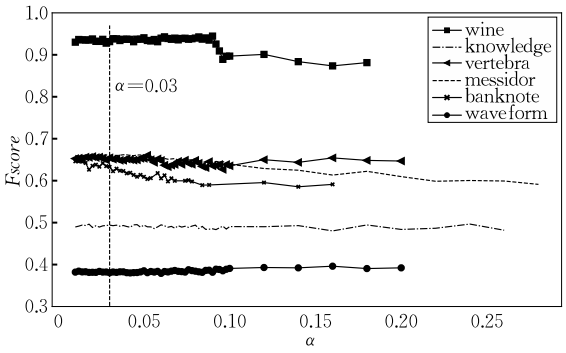

图 8 各数据集聚类指标 *Fscore* 随 α 变化图

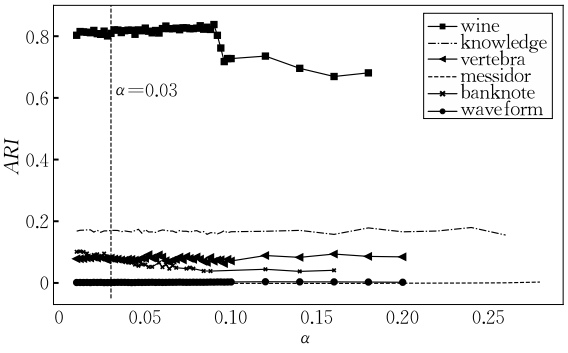

图 9 各数据集聚类指标 *ARI* 随 α 变化图

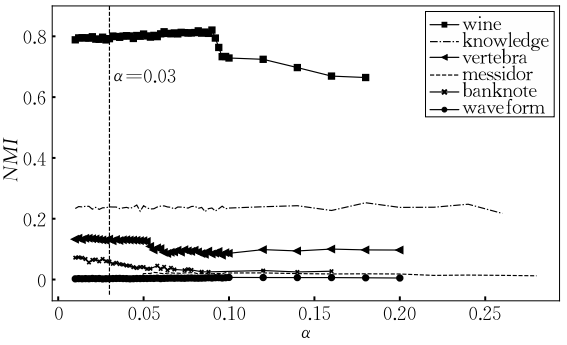

图 10 各数据集聚类指标 *NMI* 随 α 变化图

从图 7 可以看出，随着 α 的慢慢增大，当它超过某个值之后，算法 ESSC 在数据集 banknote 上的性能有所下降；同时，随着 α 的值越来越大，在数据集 wine 上的性能更是出现了严重的下降. 同样，在剩余的 3 项指标中我们也能得出同样的结论. 尤其是在数据集 wine 上的波动，在多项指标中几乎都达到了 10 个百分点. 因此，根据 6 个数据集在 4 个性能指标上的研究结果表明，参数 α 的取值选为 0.03 在大部分情况下都能获得相对较好的聚类结果.

4.4.3 超参数 η

同 ESSC 算法中的参数 α 类似，在算法 KICIC 中，超参数 η 同样用来中和调节簇内和簇间距离在整个聚类过程所起的作用. 但是算法 KICIC 的簇间距离是通过最大化子空间中簇中心与其它簇数据对象的距离来体现. 同样，由于在聚类过程中簇间距离只是具有一定的辅助作用，所以在选取参数 η 时也不能过大，以免目标函数发散.

对于参数 η 的研究方法与 α 的类似. 通过在不同的区间段为 η 设置不同的步长来观察 η 在四个指标上对算法性能的影响. 如图 11 到图 14 所示，当 η 在[0.01，0.1]这个区间段时，其步长设置为 0.002；当 η 在(0.1，0.3]这个区间段时，其步长设置为 0.02. 从图 11 可以看出，随着 η 逐渐增大，算法 KICIC 在数据集 vertebra、messidor 上的性能几乎没有变化. 对于数据集 wine、banknote、waveform 这三个数据集来说，随着 η 增大，算法 KICIC 的性能有所增加，且当 η 在 0.05 附近时，算法性能趋于稳定. 但此时算法在数据集 knowledge 上的性能却呈现出下降趋势，并且随着 η 增大，算法 KICIC 在数据集 knowledge 上几乎已不再收敛.

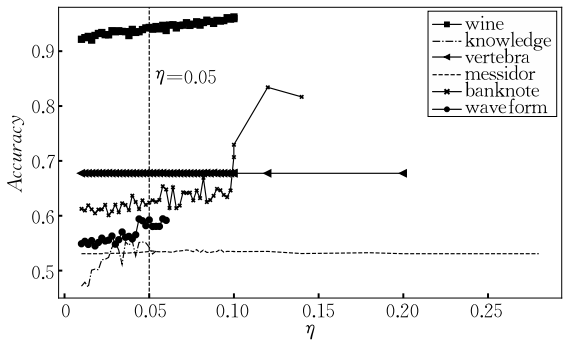

图 11　各数据集聚类 *Accuracy* 指标随 η 变化图

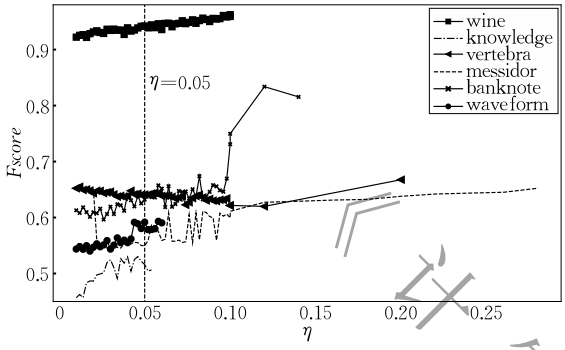

图 12　各数据集聚类指标 *Fscore* 随 η 变化图

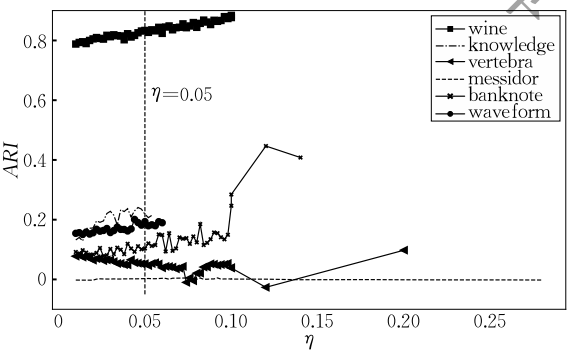

图 13　各数据集聚类指标 *ARI* 随 η 变化图

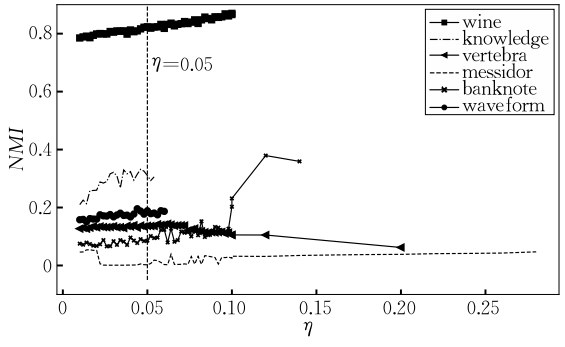

图 14　各数据集聚类指标 *NMI* 随 η 变化图

从图 12 可以发现，算法 KICIC 在数据集 wine、banknote、waveform、knowledge 上的 *Fscore* 值随 η 增大而增大，在 $\eta=0.05$ 附近趋于缓和. 然而，从算法 KICIC 在数据集 messidor、vertebra 上的表现来看，随着 η 增大至 0.02 时，性能出现了断崖式的下降，说明此时当 η 在 0.02 附近时，对于所有的数据集来说，其表现性能都不稳定.

从图 13 可以发现，对于数据集 wine、knowledge、waveform、banknote、messidor 来说，算法 KICIC 的性能总体上有所增加；而在数据集 vertebra 上的表现性能有所下降；但总体上来说在 0.005 附近都趋于稳定状态. 对于图 14 来说，算法 KICIC 在各数据集上的性能走势同图 13 类似，当且仅当 η 在 0.05 附近时，算法的表现性能最稳定.

通过对图 11 到图 14 的分析，当 $\eta=0.05$ 时，是一个比较客观且兼顾全局的取值，此时，算法性能在总体上趋于一个比较稳定的状态.

4.5　实验结果与分析

通过上一节中对算法在各个真实数据集上参数分析，我们得到 WKME、ESSC 和 KICIC 算法的参数取值为 $\gamma=40$，$\alpha=0.03$，$\eta=0.05$；表 3 是每个算法在 6 个真实数据集上运行 100 次后的平均值和标准差.

由表 3 我们可以得出，本文提出的 KICIC 算法在总体上优于其它 7 种算法. 其中，在数据集 banknote 和 knowledge 上，KICIC 在所有指标上都优于其他算法；同时，融入簇间距离的算法 KICIC 和 ESSC 整体上优于其他算法，这说明融入簇间距离能够在一定程度上提高算法的性能. 但是相比 ESSC，KICIC 在 banknote 上分别取得 2% 的精度提升、8% 的 *ARI* 提升、4% 的 *Fscore* 提升和 7% 的 *NMI* 提升；在 knowledge 上分别取得了 4% 的精度提升、5% 的 *ARI* 提升、3% 的 *Fscore* 提升和 8% 的 *NMI* 提升. 在数据集 waveform 和 wine 上，算法 KICIC 在性能指标 *Acc* 和 *Fscore* 上获得了最优的表现；其中在数据集 waveform 上，算法 KICIC 分别比对应第二优算法 WKME 和 SC 获得 4% 的精度提升和 3.4% 的 *Fscore* 提升. 而在数据集 vertebra 和 messidor 上，分别有一个性能指标获得最好的值，而在其他指标上，与对比算法获得相近的结果.

通过上述实验结果的分析以及从表 3 的结果来看，我们可以发现：当满足假设簇间最小距离小于簇内最远样本距离，即 $D<d$ 时，实验结果要总体上好于其它忽视簇间距离的算法. 例如由表 1 可知，在数据集 banknote 上存在两个簇相距较近的情况，此时对应在表 3 中的各项评价指标均高于第二优算法 4 个百分点以上. 但有少数情况下，算法 KICIC 的聚

表 3 在不同数据集上各种算法的结果对比

指标	算法	banknote	knowledge	waveform	wine	vertebra	messidor
Acc	*k*-means	0.5715(±0.02)	0.4600(±0.02)	0.5396(±0.04)	0.9443(±0.07)	0.6774(±0.000)	**0.5452(±0.004)**
	WKME	0.5821(±0.03)	0.4897(±0.06)	0.5504(±0.06)	0.9471(±0.07)	0.6774(±0.000)	0.5382(±0.007)
	EWKM	0.6143(±0.10)	0.4789(±0.09)	0.5498(±0.03)	0.9024(±0.08)	0.6777(±0.002)	0.5329(±0.001)
	ESSC	0.6307(±0.05)	0.5023(±0.03)	0.3507(±0.01)	0.9397(±0.01)	0.6778(±0.002)	0.5308(±0.000)
	AFKM	0.5588(±0.0007)	0.4780(±0.044)	0.5347(±0.024)	0.9399(±0.081)	0.6774(±0.00)	0.5416(±0.007)
	SC	0.5553(±0.00)	0.4987(±0.00)	0.526(±0.00)	0.8707(±0.00)	0.6774(±0.00)	0.5308(±0.00)
	MinMax	0.5575(±0.00)	0.4678(±0.041)	0.5154(±0.00)	0.9232(±0.09)	0.6774(±0.00)	0.5370(±0.01)
	KICIC	**0.6557(±0.12)**	**0.5427(±0.08)**	**0.5947(±0.07)**	**0.9512(±0.01)**	**0.6782(±0.005)**	0.5362(±0.008)
ARI	*k*-means	0.0225(±0.02)	0.1318(±0.02)	0.2603(±0.03)	0.8580(±0.11)	0.0983±0.081)	**0.0070(±0.002)**
	WKME	0.0301(±0.03)	0.1588(±0.06)	**0.2723(±0.06)**	**0.8632(±0.11)**	**0.1017(±0.005)**	0.0024(±0.004)
	EWKM	0.0971(±0.17)	0.1425(±0.14)	0.1596(±0.03)	0.7545(±0.12)	0.0649(±0.023)	0.0000(±0.001)
	ESSC	0.0789(±0.06)	0.1634(±0.03)	0.0011(±0.00)	0.8224(±0.04)	0.0752(±0.019)	−0.0080(±0.001)
	AFKM	0.0131(0.0003)	0.1412(0.04)	0.2561(0.023)	0.8496(0.12)	0.0887(0.030)	0.0046(0.0045)
	SC	−0.0010(±0.00)	0.1062(0.0010)	0.1698(±0.00)	0.6458(±0.00)	0.0053(±0.00)	0.0013(±0.00)
	MinMax	0.0125(±0.00)	0.1334(±0.03)	0.2451(±0.00)	0.7960(±0.08)	0.0703(±0.00)	0.0036(±0.006)
	KICIC	**0.1542(±0.19)**	**0.2183(±0.10)**	0.1942(±0.07)	0.8534(±0.04)	0.0637(±0.049)	0.0034(±0.003)
FSC	*k*-means	0.5718(±0.02)	0.1945(±0.02)	0.5395(±0.03)	0.9469(±0.06)	0.6694(±0.006)	0.5533(±0.027)
	WKME	0.5812(±0.03)	0.4769(±0.05)	0.5526(±0.05)	0.9482(±0.06)	**0.6720(±0.004)**	0.6029(±0.057)
	EWKM	0.6113(±0.11)	0.4652(±0.09)	0.5416(±0.03)	0.9046(±0.07)	0.6458(±0.028)	**0.6601(±0.008)**
	ESSC	0.6268(±0.05)	0.4898(±0.02)	0.3792(±0.01)	0.9390(±0.01)	0.6505(±0.014)	0.6597(±0.010)
	AFKM	0.5600(±0.0007)	0.4689(±0.0332)	0.5342(±0.022)	0.9431(±0.06)	0.6722(±0.011)	0.5814(±0.053)
	SC	0.5410(±0.00)	0.4355(0.2127)	0.5584(±0.00)	0.8694(±0.00)	0.5642(±0.00)	0.5349(±0.00)
	MinMax	0.5589(±0.00)	0.4633(±0.024)	0.5181(±0.00)	0.9220(±0.092)	0.6472(±0.00)	0.5442(±0.011)
	KICIC	**0.6620(±0.12)**	**0.5160(±0.07)**	**0.5925(±0.07)**	**0.9508(±0.01)**	0.6485(±0.025)	0.5505(±0.016)
NMI	*k*-means	0.0166(±0.01)	0.1945(±0.02)	0.3686(±0.01)	0.8474(±0.08)	0.1587(±0.005)	0.0088(±0.007)
	WKME	0.0210(±0.022)	0.2323(±0.08)	**0.3724(±0.03)**	**0.8503(±0.08)**	**0.1608(±0.003)**	0.0184(±0.013)
	EWKM	0.07094(±0.13)	0.2197(±0.15)	0.1641(±0.04)	0.7620(±0.09)	0.1287(±0.023)	0.0261(±0.014)
	ESSC	0.0555(±0.04)	0.2309(±0.04)	0.0025(±0.00)	0.8042(±0.02)	0.1325(±0.004)	0.0102(±0.016)
	AFKM	0.0109(0.0004)	0.2102(0.0508)	0.3663(0.01)	0.8414(±0.09)	0.1451(±0.04)	0.0164(±0.014)
	SC	0.0002(±0.00)	0.1819(±0.00)	0.1642(±0.00)	0.649(±0.00)	0.0024(±0.00)	0.0011(±0.00)
	MinMax	0.0098(±0.00)	0.2013(±0.04)	0.3511(±0.00)	0.7934(±0.033)	0.1554(±0.00)	0.0038(±0.004)
	KICIC	**0.1264(±0.16)**	**0.3103(±0.11)**	0.1901(±0.05)	0.8397(±0.03)	0.1349(±0.021)	**0.0445(±0.011)**

类结果不如部分现有算法，如数据集 vertebra，这可能是由过多的噪音维度所造成的，不同的噪音维度可能对各种算法具有不同的影响.

由此分析，在大部分情况下算法 KICIC 优于其他比较算法. 也即是说，通过在子空间内最大化簇中心与其他簇数据对象的方式来最大化簇间距离能够提高算法的聚类结果.

4.6 收敛速度

通过 3.3 节的分析，KICIC 算法与其它算法有着近似的时间复杂度. 在本小节中，我们进一步在实验上对比不同算法所花费的运行时间. 图 15 展示了 8 种算法在 6 个数据集上随机初始化中心，并运行 100 次得到的运行时间. 从图中可以看出，在相同数据集上，KICIC 算法与其他算法所花费时间没有数量级上的差别. 但是不同算法之间还是有一定差别，导致这个差别的原因可能是 KICIC、EWKM 和 ESSC 需要为每一个簇计算一个权重向量，WKME、MinMax 为整个数据集计算一个权重向量，而 *k*-means 不需要计算权重向量. 同时 AFKM 需要计算一个随机初始化中心点的种子，而 SC 需要在聚类前先生成一个聚类图.

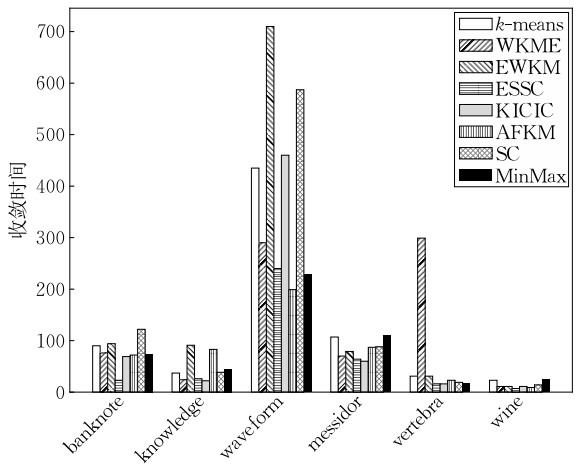

图 15 收敛时间图

4.7 鲁棒性

算法 KICIC 的两个超参数 γ 和 η 均会影响到

算法的性能.但由于 KICIC 是拓展自算法 EWKM，超参数 γ 是来自算法 EWKM 中的参数，该参数的主要作用是通过调节权重 $\boldsymbol{W}$ 的熵在目标函数中的作用来影响权重 $\boldsymbol{W}$ 的分布.该参数在原文中有详细的研究，本文也在相应的数据集中研究过超参数 γ 对聚类结果的影响，得出的结论是 γ 在 $40\leq\gamma\leq50$ 时，算法结果相对更平稳，如图 3 到图 6 所示.接下来，我们固定 $\gamma=40$ 来研究参数 η 对聚类结果的影响.在算法 KICIC 中，参数 η 的作用是用来调节簇内和簇间距离在聚类过程所起的作用；因此对于算法 KICIC 的鲁棒性可通过其对超参数 η 的敏感程度来分析.从图 11 到图 14 可知：KICIC 在 $0.04\leq\eta\leq0.07$ 时，算法 KICIC 在不同数据集上的聚类性能相对较稳定.

5　结束语

本文提出了一种新的集成簇内和簇间距离的加权 k-means 聚类算法.与传统通过最大化簇中心与全局中心来最大化簇间距离不同，本文主要通过在子空间内最大化簇中心与其他簇数据对象的距离来最大簇间距离.传统的最大化簇间聚类是使得每个簇中心远离全局中心，但是远离全局中心并不等价于任意两个簇相互远离.而本文所提出的方法能克服该问题.本文提出的最大化簇间距离的方法是使得每个簇中心不但能代表本簇的数据对象，也能尽可能远离不属于本簇的数据对象.根据此思想，本文首先为算法设计了一个目标函数，然后再通过优化求解目标函数获得算法的迭代规则.最后，在 6 个真实数据集上的实验结果表明 KICIC 算法能够在大部分情况下获得更优的聚类性能.

在现实应用中，有许多的高维数据同时具有各类簇结构特征，在未来的研究工作中，我们计划通过修改目标函数使得该方法能够适应于包含各种复杂簇结构的情况，以此来拓展算法的应用范围.

参　考　文　献

[1] Lin X, Li C T. Large-scale image clustering based on camera fingerprints. IEEE Transactions on Information Forensics & Security, 2017, 12(4): 793-808

[2] Li C, Yang C, Jiang Q. The research on text clustering based on LDA joint model. Journal of Intelligent & Fuzzy Systems, 2017, 32(5): 3655-3667

[3] Hu C W, Li H, Qutub A A. Shrinkage clustering: A fast and size-constrained clustering algorithm for biomedical applications. BMC Bioinformatics, 2018, 19(1): 19:1-11

[4] Han J, Kamber M, Pei J, Data Mining: Concepts and Techniques. San Mateo, USA: Morgan Kaufmann, 2011

[5] Deng Z, Choi K S, Chung F L, et al. Enhanced soft subspace clustering integrating within-cluster and between-cluster information. Pattern Recognition, 2010, 43(3): 767-781

[6] Huang J Z, Ng M K, Rong H, et al. Automated variable weighting in k-means type clustering. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2005, 27(5): 657-668

[7] Jing L, Ng M K, Huang J Z. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. IEEE Transactions on Knowledge & Data Engineering, 2007, 19(8): 1026-1041

[8] Jain A K. Data clustering: 50 years beyond k-means// Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Berlin, Germany, 2008: 3-4

[9] Xu R, Wunsch D. Survey of clustering algorithms. IEEE Transactions on Neural Networks, 2005, 16(3): 645-678

[10] Sun Ji-Gui, Liu Jie, Zhao Lian-Yu. Clustering algorithms research. Journal of Software, 2008, 19(1): 48-61(in Chinese) (孙吉贵，刘杰，赵连宇. 聚类算法研究. 软件学报，2008, 19(1): 48-61)

[11] Steinhaus H. Sur la division des corp materiels en parties. Bulletin of the Polish Academy of Science-Mathematics, 1956, 1(804): 801

[12] Arthur D, Vassilvitskii S. k-means++: The advantages of careful seeding//Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. New Orleans, USA, 2007: 1027-1035

[13] Bachem O, Lucic M, Hassani S H, et al. Approximate k-means++ in sublinear time//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 1459-1467

[14] Moseley B, Vattani A, Kumar R, Vassilvitskii S. Scalable k-means++. Very Large Data Bases, 2012, 5(7): 622-633

[15] Shahnewaz S M, Rahman M A, Mahmud H. A self acting initial seed selection algorithm for k-means clustering based on convex-hull//Proceedings of the International Conference on Informatics Engineering and Information Science. Berlin, Germany, 2011: 641-650

[16] Bachem O, Lucic M, Hassani H, et al. Fast and provably good seedings for k-means//Proceedings of the Advances in Neural Information Processing Systems. Barcelona, Spain, 2016: 55-63

[17] Dan P, Moore A W. X-means: Extending k-means with efficient estimation of the number of clusters//Proceedings of the 17th International Conference on Machine Learning. San Francisco, USA, 2000: 727-734

[18] Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society, 2001, 63(2): 411-423

[19] Dhillon I S, Modha D S. Concept decompositions for large sparse text data using clustering. Machine Learning, 2001, 42(1-2): 143-175

[20] Dzogang F, Marsala C, Lesot M, et al. An ellipsoidal *k*-means for document clustering//Proceedings of the IEEE, International Conference on Data Mining. Washington, USA, 2012: 221-230

[21] Kashima H, Hu J, Ray B, et al. *K*-means clustering of proportional data using L1 distance//Proceedings of the 19th International Conference on Pattern Recognition. Tampa, USA, 2008: 1-4

[22] Chan E, Ching W, Ng M, et al. An optimization algorithm for clustering using weighted dissimilarity measures. Pattern Recognition, 2004, 37(5): 943-952

[23] Huang X, Yang X, Zhao J, et al. A new weighting *k*-means type clustering framework with an *L*2-norm regularization. Knowledge-Based Systems, 2018, 151: 165-179

[24] Wang Jun, Wang Shi-Tong, Deng Zhao-Hong. 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)
(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)

[25] Wang Hong-Jie, Shi Yan-Wen. *K*-Means clustering algorithm based on initial center optimization and feature weighted. Computer Science, 2017, 44(b11): 457-459(in Chinese)
(王宏杰, 师彦文. 结合初始中心优化和特征加权的 *K*-Means 聚类算法. 计算机科学, 2017, 44(b11): 457-459)

[26] Tzortzis G, Likas A, Tzortzis G. The minmax *k*-means clustering algorithm. Pattern Recognition, 2014, 47(7): 2505-2516

[27] Huang X, Ye Y, Zhang H. Extensions of kmeans-type algorithms: A new clustering framework by integrating intracluster compactness and intercluster separation. IEEE Transactions on Neural Networks & Learning Systems, 2014, 25(8): 1433-1446

[28] Huang X, Ye Y, Guo H, et al. DSKmeans: A new kmeans-type approach to discriminative subspace clustering. Knowledge-Based Systems, 2014, 70: 293-300

[29] Fisher R A. The use of multiple measurements in taxonomic problems. Annals of Human Genetics, 1936, 7(2): 179-188

[30] Bezdek J C. A convergence theorem for the fuzzy ISODATA clustering algorithms. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1980, PAMI-2(1): 1-8

[31] Selim S Z, Ismail M A. *K*-means-type algorithms: A generalized convergence theorem and characterization of local optimality. IEEE Transactions on Pattern Anal Mach Intell, 1984, 6(1): 81-87

[32] Tarn C, Zhang Y, Feng Y. Sampling clustering. arXiv: 1806.08245. 2018: 1-10

HUANG Xiao-Hui, Ph. D., associate professor. His research interests are in the areas of data mining, and machine learning.

WANG Cheng, M. S. candidate. His major research interests focus on data mining and machine learning.

XIONG Li-Yan, M. S., professor. Her research interests include database system, natural language processing.

ZENG Hui, M. S., associate professor. His research interests include data mining and machine learning.

Background

In the real world, high-dimensional data is ubiquitous, such as image data, text data, biological information data, etc. The clustering problem of high-dimensional data sets is an important research issue in the field of data mining. In the last decades, the area of this research has attracted the attention of many researchers. The research of this field can be observed in various conferences like SIGKDD, IJCAI, ICML and ICDE.

Subspace clustering is one type of methods to solve high-dimensional clustering problem. This paper proposes a new *k*-means type of clustering algorithm which is able to integrate both intra-cluster compactness and inter-cluster separation. Different to traditional approaches which can integrate the information of intra-cluster compactness and inter-cluster separation by maximizing the distance between the center of each cluster and the global center, our proposed algorithm maximizes inter-cluster separation by maximizing the distance between the center of a cluster and the objects in other clusters in subspace. In this paper, we firstly develop an objective function of our proposed method and obtain the iterative rules by optimizing the function. At last, we verify the effectiveness of our proposed method in the experiments of six real-life data sets.

This work was supported by the National Natural Science Foundation of China under Grant No. 61562027, the Jiangxi Social Sciences "12th Five-Year" (2015) Planning Project No. 15XW12, the Natural Science Foundation of Jiangxi Province under Grant Nos. 20181BAB202024, 20192ACBL21006, and the Education Department of Jiangxi Province under Grant No. GJJ170413 and No. GG170379.