一种改进的多视图聚类集成算法

邓 强 杨 燕 王 浩
(西南交通大学信息科学与技术学院 成都 610031)

摘 要 近年来,针对大数据的数据挖掘技术和机器学习算法研究变得日趋重要。在聚类领域,随着多视图数据的大量出现,多视图聚类已经成为了一类重要的聚类方法。然而,大多数现有的多视图聚类算法受算法参数设置、数据样本等影响,具有聚类结果不稳定、参数需要反复调节等缺点。基于多视图K-means算法和聚类集成技术,提出了一种改进的多视图聚类集成算法,其提高了聚类的准确性、鲁棒性和稳定性。其次,由于单机环境下的多视图聚类算法难以对海量的数据进行处理,结合分布式处理技术,实现了一种分布式的多视图并行聚类算法。实验证明,并行算法在处理大数据时的时间效率有很大提升,适合于大数据环境下的多视图聚类分析。

关键词 多视图聚类,聚类集成,分布式计算,并行化
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.01.012

Improved Multi-view Clustering Ensemble Algorithm

DENG Qiang YANG Yan WANG Hao
(School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China)

Abstract In recent years, data mining and machine learning algorithms for big data become increasingly important. In the clustering, with the appearance of multi-view data, multi-view clustering has become an important clustering method. However, many existing multi-view clustering algorithms are easily affected by parameter setting and dataset itself, so the clustering results are usually unstable. To overcome this problem, we presented a new multi-view clustering ensemble algorithm based on the multi-view K-means clustering algorithm in this paper. This algorithm uses ensemble technique to improve the multi-view K-means algorithm performance, increasing the accuracy, robustness, and stability of clustering results. It is well known that one single computer cannot process too much data, because one computer has the limited computation resources. To improve the efficiency of multi-view clustering, we implemented a distributed multi-view clustering ensemble algorithm based on distributed processing technology. Experimental results show that the proposed approach has higher efficiency when processing large dataset, and it is suitable for multi-view clustering in big data environment.

Keywords Multi-view clustering, Clustering ensemble, Distributed Computation, Parallelization

1 引言

多视图聚类是一种对多视图数据进行聚类的方法。在传统聚类算法中,假设数据只有一个视图,但是实际上,多视图数据是广泛存在的。例如,对于网页数据,其中的网页链接作为一个视图,文本内容作为另一个视图;对于图像数据,可以通过不同的特征进行描述,如SIFT特征、HOG特征等,不同的特征就可以作为数据的不同视图。在多视图聚类中,聚类数据由多个视图构成,具有一致性和互补性[1]的特点,所有视图共享同一标签信息。多视图聚类就是要找到一个满足所有视图的最优划分。在传统的聚类方法中,数据作为一个整体进行处理,在本文中将这类方法称为单视图聚类。

近年来,国内外许多研究者对多视图聚类的相关研究进行关注,并提出了多种多视图聚类算法。文献[2]首次提出了多视图聚类问题,并提出了Co-EM多视图聚类算法,该算法结合协同训练(Co-training)和EM算法解决多视图聚类问题。文献[3]提出了多视图协同正则化聚类算法。基于协同训练思想的多视图算法存在的缺点是,协同训练适合只有两个视图的情况。有研究者从不同视图的差异性出发,提出了对多视图进行加权的聚类算法,文献[4]提出了加权的K-means多视图聚类算法。文献[5]将核方法应用到多视图聚类中,提出了基于核方法的多视图K-means聚类和多视图谱聚类算法,该方法赋予每个视图一个权重,通过优化学习,得到一个新的核矩阵,取得了更好的聚类效果。

在许多多视图聚类算法中,聚类结果容易受到聚类参数和数据样本的影响,这极大地影响了算法的稳定性,限制了算法的使用场景。为了克服这一缺点,在文献[6,7]中将聚类集成应用于多视图聚类中,提升了多视图聚类的稳定性。聚类

到稿日期:2015-09-21 返修日期:2015-11-29 本文受国家自然科学基金(61170111,61572407,61134002),国家科技支撑计划课题(2015BAH19F02),四川省科技支撑计划项目(2014SZ0207)资助。

邓 强(1991-),男,硕士生,主要研究方向为数据挖掘;杨 燕(1964-),女,教授,博士生导师,主要研究方向为数据挖掘、计算智能、集成学习等,E-mail:yyang@swjtu.edu.cn(通信作者);王 浩(1993-),男,博士生,主要研究方向为数据挖掘、机器学习。

集成是一种采用集成学习提高聚类稳定性、鲁棒性和聚类精度的方法。文献[8]首次提出了聚类集成概念，并基于图划分思想提出了 CSPA，HGPA，MCLA 3 种聚类集成算法，显著提高了聚类精度。文献[9]提出了一种基于链接的相似度矩阵生成算法。在此基础上，本文提出了一种改进的多视图聚类集成算法，该算法同时采用单视图聚类算法和多视图聚类算法作为基聚类算法，通过基于链接的方法进行集成。

近年来，出现了许多并行化聚类算法的研究成果[10-15]。并行化聚类是提升聚类效率的有效途径。然而，目前关于并行化多视图聚类算法和并行化聚类集成算法的研究还较少。本文结合 Spark 大数据处理平台，基于分布式计算，实现了一种分布式多视图聚类集成算法。该算法可在 Spark 集群上并行地处理数据，显著提高了聚类的效率；同时，能够对大数据集进行有效处理。

2 基本原理

2.1 多视图 K-means 聚类

K-means 聚类是一种经典的聚类算法，具有简单、高效的特点，被广泛应用于实际的聚类问题中。已有相关研究将 K-means聚类拓展至多视图聚类中。文献[4]提出了一种具有鲁棒性的多视图 K-means 算法。

在多视图 K-means 算法中，假设多视图数据集 $X=[x^1,x^2,\cdots,x^m]\in R^{d_v\times n}$，$1\leqslant v\leqslant m$，表示数据集的样本数为 n，视图个数为 m，每个视图的维度是 d_v。$U\in R^{n\times k}$ 是一个表示聚类结果的聚簇指示矩阵，k 等于聚类簇的个数。对于矩阵 U，每一行只有一个元素为 1，其余元素为 0。$C^{(v)}\in R^{d_v\times k}$，$1\leqslant v\leqslant m$ 表示每个视图的聚类中心。为了衡量视图的重要性，引入了一个视图权重向量 $(w^{(1)},w^{(2)},\cdots,w^{(m)})$，其中 $0\leqslant w^{(v)}\leqslant 1$。多视图 K-means 聚类的目标函数可以表示成以下形式：

$$J=\min_{u_{ij},c_i^{(v)},w^{(v)}}\sum_{v=1}^{m}(w^{(v)})^{\gamma}\sum_{i=1}^{k}\sum_{j=1}^{n}\mu_{ji}(x_j^{(v)}-c_i^{(v)})^2$$
$$\text{s.t. }\mu_{ji}\in\{0,1\},\sum_{i=1}^{k}\mu_{ji}=1,\sum_{v=1}^{m}w^{(v)}=1 \tag{1}$$

其中，$\mu_{ji}=1$ 表示样本 j 属于簇 i，$x_j^{(v)}$ 表示在视图 v 下的样本 j，$c_i^{(v)}$ 表示在视图 v 下的第 i 个簇的聚类中心。

为了最小化目标式(1)，采用迭代寻优进行求解。在式(1)中有 3 个参数需要求解：μ_{ji}，$c_i^{(v)}$，$w^{(v)}$，固定两个参数，求解第 3 个参数，通过不断迭代，直到算法收敛。求解过程如下：

首先随机初始化聚类中心和视图权重，求解 μ_{ji}。

$$\begin{cases}u_{ji}=1,D_j\leqslant D_s,D_s=\sum_{v=1}^{m}(w^{(v)})^{\gamma}sim(x_j^{(v)},c_i^{(v)})\\ u_{js}=0,s\neq i,1\leqslant s\leqslant k\end{cases} \tag{2}$$

然后计算聚类中心 $c_i^{(v)}$，更新公式如下：

$$c_i^{(v)}=\frac{\sum_{i=1}^{n}\mu_{ji}x_j^{(v)}}{\sum_{j=1}^{n}\mu_{ji}} \tag{3}$$

最后计算视图的权重 $w^{(v)}$，由于视图的权重和为 1，因此，可采用拉格朗日乘子法求解。求解过程如下：

首先，令

$$H^{(v)}=\sum_{i=1}^{K}\sum_{j=1}^{n}\mu_{ji}(x_j^{(v)}-c_i^{(v)})^2 \tag{4}$$

加入拉格朗日乘子项后，目标函数转化为如下形式：

$$L=\sum_{v=1}^{m}(w^{(v)})^{\gamma}H^{(v)}-\lambda(\sum_{v=1}^{m}w^{(v)}-1) \tag{5}$$

将式(5)相对于 $w^{(v)}$ 求偏导，可得：

$$w^{(v)}=(\frac{\lambda}{\gamma H^{(v)}})^{\frac{1}{\gamma-1}} \tag{6}$$

将约束项 $\sum_{v=1}^{m}w^{(v)}=1$ 代入式(6)，可以得到视图权重的更新公式：

$$w^{(v)}=\frac{(\gamma H^{(v)})^{\frac{1}{1-\gamma}}}{\sum_{v=1}^{M}(\gamma H^{(v)})^{\frac{1}{1-\gamma}}} \tag{7}$$

多视图 K-means 算法的关键在于设置参数 γ，不同的参数设置将直接导致不同的聚类结果。然而，参数 γ 的设置依赖于经验值。为降低这种单一算法对参数的依耐性，本文采用聚类集成方法进行改进。

2.2 聚类集成

假设 $X=\{x_1,x_2,\cdots,x_n\}$ 表示输入的数据样本，对数据进行 r 次聚类，得到一组聚类划分 $P=\{\pi_1,\pi_2,\cdots,\pi_r\}$，$\pi_i$ 表示第 i 个聚类划分。则聚类集成可表示为如下形式：

$$P=\{\pi_1,\pi_2,\cdots,\pi_r\}\rightarrow\pi^* \tag{8}$$

在基于链接的相似度矩阵生成算法中，以聚类划分 P 作为输入，然后输出新的相似度矩阵 CTS(Connected-Triple Based Similarity Matrix)。CTS 的表示形式如下：

$$S_l(x_i,x_j)=\begin{cases}1, & \text{if } C(x_i)=C(x_j)\\ S_{WT}(C(x_i),C(x_j))\times DC, & \text{otherwise}\end{cases} \tag{9}$$

$$CTS(x_i,x_j)=\frac{1}{r}\sum_{l=1}^{r}S_l(x_i,x_j) \tag{10}$$

在式(9)、式(10)中，r 表示聚类划分的个数，$C(x_i)$ 表示样本 x_i 所属簇，$S_l(x_i,x_j)\in R^{n\times n}$ 表示样本的成对相似度矩阵。S_{WT} 是一个衡量两个簇相似性的度量，DC 是权重因子。CTS 的具体计算方法可参见文献[9]，文本不再赘述。通过 CTS 表示的相似度矩阵，将簇与簇的相似性转化为对象间的相似性，减少了原有相似度矩阵中的零值，提高了集成算法的准确性。

3 多视图聚类集成

聚类集成是提高聚类质量的有效方法，文本提出了一种多视图聚类集成算法，简称 LMKCE。算法框架如图 1 所示。

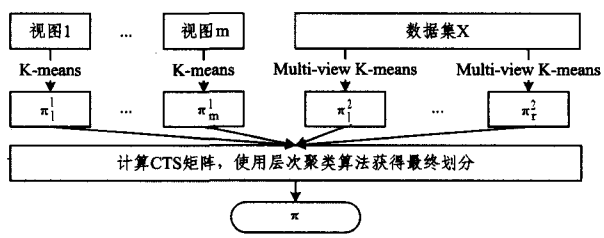


图 1 多视图聚类集成框架

假设多视图数据的各个视图相互独立，每个视图可以看作是一个单独的数据集合。首先分别对每个视图进行聚类，得到一组聚类划分 $P^1=\{\pi_1^1,\pi_2^1,\cdots,\pi_m^1\}$，这里的 m 等于视图的个数；然后采用多视图 K-means 算法聚类，并设置不同的初始参数，得到另一组聚类分量 $P^2=\{\pi_1^2,\pi_2^2,\cdots,\pi_r^2\}$。合并聚类分量 $P=P^1\cup P^2$，将 P 作为集成算法的输入，得到最后

的聚类结果。这样做有两个优点：1）获得了更多具有差异性的基聚类结果；2）由于多视图具有一致性和互补性特点，在对单个视图进行聚类时，获得了每个视图独有的信息，在进行多视图聚类时，获得了视图间的一致性信息。以下是多视图聚类集成算法流程描述。

算法 1　Link-based Multi-View K-Means Clustering Ensemble

输入：多视图的样本数据$\{X^{(1)},\cdots,X^{(m)}\}$，$X^{(v)}\in R^{d^{(v)}\times n}$，簇数 K，权重参数 γ，收敛阈值 t，最大迭代次数 I，运行次数 r

输出：数据对应的标签 L

1. 分别对每个视图聚类，得到聚类分量 P^1
2. 循环执行多视图 K-means 算法，得到聚类分量 P^2，循环过程如下：
   2.1. 初始化权重向量 $W=[w^{(1)},w^{(2)},\cdots,w^{(m)}]$，$w^{(v)}=1/m$，随机选择 K 个点作为初始聚类中心。
   2.2. 运行多视图 K-means 算法聚类直到收敛或达到最大迭代次数。
   2.3. 运行次数加 1。
   2.4. 如果运行次数小于 r，则对权重参数 γ 进行微调，然后转步骤 1，否则退出循环。
3. 合并 P^1 和 P^2，生成相似度矩阵 CTS。
4. 采用聚合的层次聚类算法对 CTS 进行划分，得到最终的聚类结果 π。

4　分布式多视图聚类集成算法

在目前的数据挖掘中，数据规模越来越大。面对日益复杂和海量的数据，一个算法能否对大数据进行处理，是该算法能否进行海量数据挖掘的关键。为了提高 LMKCE 算法的运行效率，使其具备处理大数据的能力，本文基于 Spark 大数据处理平台，实现了一种分布式的多视图聚类集成算法。该算法分为两个部分：1）分布式多视图 K-means 算法（DMKC 算法）；2）分布式聚类集成算法（DMKCE 算法）。

4.1　分布式多视图 K-means 算法

在 Spark 环境下，数据采用 RDD（Resilient Distributed Datasets）表示。多视图数据表示为如图 2 所示的数据结构。m 代表视图数，样本的每一个视图用向量 vector 表示。

$$\{\text{vector}^{(1)},\text{vector}^{(2)},\cdots,\text{vector}^{(m)}\}$$
$$\{\text{vector}^{(1)},\text{vector}^{(2)},\cdots,\text{vector}^{(m)}\}$$
$$\vdots$$

图 2　多视图数据 RDD 表示

在进行聚类的过程中，数据样本被分为多个分片（partitions），分布在不同的计算节点上，不必在节点间传输。这样的实现减少了节点间 I/O 的时间，同时充分利用了各节点的计算资源，提高了集群效率。聚类中心和视图权重则需要在各个计算节点上保持一致，以保证每轮迭代计算正确。在 DMKC 算法实现中，利用了 Spark 的广播机制，该机制允许在 Spark 集群中共享数据。因此通过广播聚类中心和视图权重可以使算法参数信息在集群上保持一致。

算法 2 对 DMKC 算法的实现流程进行描述。

算法 2　Distributed Multi-View K-Means Clustering Algorithm

输入：多视图数据$\{X^{(1)},\cdots,X^{(m)}\}$，$X^{(v)}\in R^{d^{(v)}\times n}$，簇数 K，权重参数 γ，收敛阈值 t，最大迭代次数 I

输出：样本数据标签 L，多视图的聚类中心 V，视图权重 W

1. 初始化：从 HDFS（Hadoop 分布式文件系统）上加载多视图数据，并表示为 RDD 形式；随机初始化聚类中心，初始化视图权重。
2. 迭代更新聚簇指示矩阵 U、聚类中心 V、视图权重 W，直到算法收敛或者达到最大迭代次数。循环过程如下：
   2.1. 采用 Spark 广播机制，广播聚类中心和视图权重。
   2.2. 阶段 1：分布式操作，计算每个视图中样本数据到聚类中心的距离，然后按照视图权重进行加权求和，选择距样本最近的聚类中心作为类标。最后，将结果合并成〈键，值〉对的形式。
   2.3. 阶段 2：合并有相同键的键值对集合，将结果返回给 Driver 节点，并计算新的聚类中心和视图权重。
   2.4. 计算目标函数值，迭代次数加 1。
   2.5. 判断是否收敛或达到最大迭代次数，符合条件则退出循环。
3. 计算样本数据标签 L，输出聚类结果。

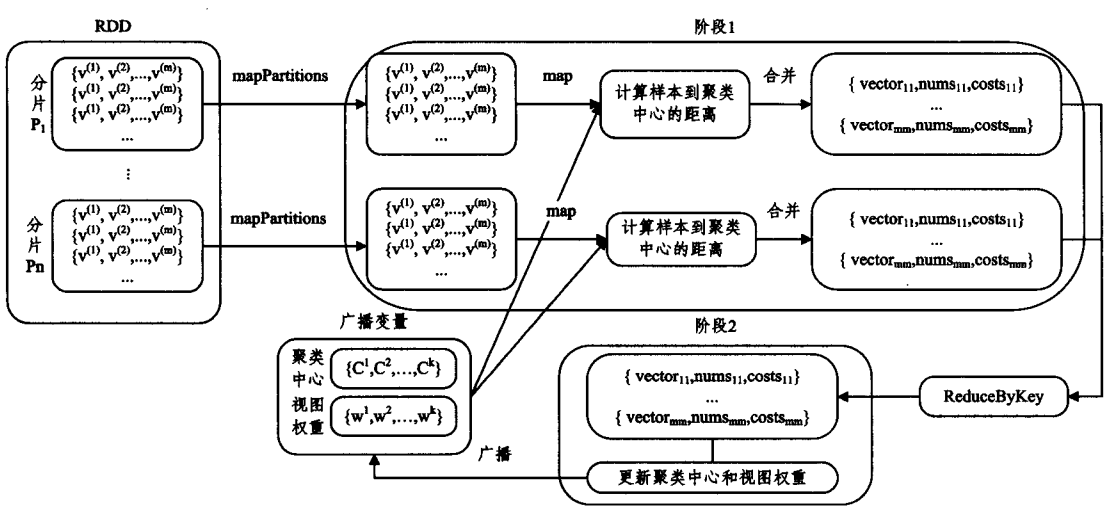


图 3　更新多视图聚类中心和视图权重过程

图 3 展示了 DMKC 算法迭代处理的过程，可以看到算法分为两个主要阶段。通过阶段 1 更新样本的簇信息，通过阶段 2 更新聚类中心和样本权重。阶段 2 是影响算法效率的关键部分，为了提高其性能，需要对阶段 1 的输出进行处理。

对每个数据分片，经过阶段 1 处理后，输出结果合并为如图 4 所示的键值对集合 $Item_{ij}=(key_i,value_j)\in R^{m\times k}$。在 $Item_{ij}$ 中 $key_i=(v_i,k_j)$，v_i 表示第 i 个视图，k_j 表示第 j 个簇；$value_j=(vector_{ij},nums_{ij},costs_{ij})$，$vector_{ij}$ 表示视图 i 中属于第 j 个簇的所有样本点的和，$nums_{ij}$ 表示视图 i 中属于第 j 个簇的样本点的个数，$costs_{ij}$ 表示视图 i 对第 j 个簇在目标函数

中的距离和。得到上述中间数据后，进行阶段 2 处理，并将结果返回到 Driver 节点，最后就可以按照式(3)和式(7)更新聚类中心和视图权重。

$$\begin{pmatrix} Item_{11} & \cdots & Item_{1k} \\ \vdots & \ddots & \vdots \\ Item_{ml} & \cdots & Item_{mk} \end{pmatrix}$$

$$Item_{i,j}=((v_i,k_j),(vector_{ij},nums_{ij},costs_{ij}))$$

图 4 阶段 1 输出的键值对的矩阵形式

4.2 分布式多视图聚类集成算法

在分布式 K-means 算法的基础上，本文提出进一步分布式多视图聚类集成算法(DMKCE)。该算法的实现流程如下：

算法 3 Distributed Multi-View K-Means Clustering Ensemble

输入：多视图的样本数据$\{X^{(1)},\cdots,X^{(m)}\}$，$X^{(v)}\in R^{d^{(v)}\times n}$，聚类的簇数 K，权重参数 γ，收敛阈值 t，最大迭代次数 I，基聚类算法运行次数 r

输出：数据标签 L

1. 分别对每个视图采用分布式 K-means 算法聚类，得到聚类划分 P^1。
2. 设置不同的初始参数，运行 r 次分布式多视图 K-means 算法，得到聚类划分 P^2。
3. 合并聚类分量 $P=P^1+P^2$，采用分布式算法生成相似度矩阵 CTS。
4. 使用分布式 K-means 聚类算法对 CTS 进行划分，最后将结果输出到本地。

在 DMKCE 中，聚类分量表示为 $\pi_j=\{(x_1,label_1),\cdots,(x_n,label_n)\}$，即每条数据包括样本的 id 和标签。为了得到新的相似度矩阵，按图 5 所示的方式对聚类划分 P 进行处理，得到每个簇包含的数据样本。然后，通过联结操作得到相似度矩阵 CTS。对于 CTS 矩阵，可以采用任意的分布式聚类算法进行划分即可。在本文中，使用了分布式 K-means 算法获得最终划分。

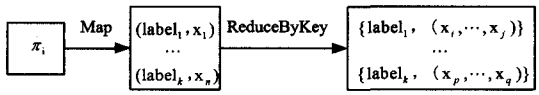


图 5 聚类分量的处理过程

5 实验

5.1 多视图聚类集成算法对比实验

本文在多个标准数据集上进行了实验。分别是 UCI digits 数据集，该数据集有 2000 条数据、6 个视图，包含 0－9 共 10 种字体类别，每个类别有 200 个样本；3 Sources 文本数据集，该数据集从 3 个数据源收集，总共包含 948 篇新闻文章，被手动标记为 6 个不同的主题类别，其中有 169 篇文章在 3 个数据源中同时出现，在本文中，使用这 169 条数据进行实验，每一个数据源当作一个独立的视图；Reuters 多语言数据集，该数据集包含了由 5 种不同语言书写的文本，在文献[16]中，将该数据分为 6 个类别，在实验中，将每种语言的文本数据当作一个视图；WebKB 数据集，该数据集包含 4 个子数据集(Cornell，Texas，Washington 和 Wisconsin)，本文只使用了 Texas 数据集，其中包含两个视图，一个视图是网页文本和词的关系数据，另一个视图是网页的链接关系数据。数据集的汇总信息如表 1 所列。

表 1 实验数据集信息汇总

数据集	样本个数	视图数	簇数
UCI digits	2000	6	10
3 Sources	169	3	6
Reuters	1200	5	6
WebKB	187	2	5

表 2 聚类集成实验结果

数据集	评价指标	算法					
		SC	RMKMC	RMSC	MvKKMCE	MvSpecCE	LMKCE
UCI digits	NMI	0.6466	0.8115	0.6466	0.6677	0.6535	**0.9010**
	F-measure	0.5953	0.7329	0.5557	0.5898	0.6046	**0.9113**
	RI	0.9183	0.9396	0.9183	0.9126	0.9192	**0.9823**
3Sources	NMI	0.4718	0.4327	0.4260	0.4600	0.5147	**0.5530**
	F-measure	0.4832	0.5185	0.4079	0.3969	**0.6023**	0.5261
	RI	0.7761	0.7813	0.6072	0.7327	0.8095	**0.8132**
Resuters	NMI	0.3029	0.3028	0.3554	0.3052	0.0501	**0.3679**
	F-measure	0.3458	0.3723	**0.4111**	0.3472	0.2824	0.3830
	RI	0.7241	0.7682	**0.7911**	0.7245	0.2178	0.7479
WebKB	NMI	0.3022	0.1938	0.1475	0.2721	0.1215	**0.4174**
	F-measure	0.3968	0.4774	0.3955	0.5583	0.6341	**0.6614**
	RI	**0.6474**	0.6172	0.6224	0.5009	0.6434	0.6439

实验对比算法采用了经典聚类算法、多视图聚类算法以及多视图聚类集成算法。对比算法的简要信息描述如下：

(1)谱聚类算法(Spectral Clustering，SC)。将多视图数据集组合为一个单视图数据集，采用谱聚类进行聚类。

(2)RMKMC 算法。该算法是文献[4]中提出的鲁棒性多视图 K-means 算法。

(3)RMSC 算法[17]是基于矩阵低秩和稀疏表示的一种多视图子空间聚类算法。

(4)MvKKMCE 算法和 MvSpecCE 算法是两种分别基于 K-means 和谱聚类的多视图聚类集成算法。

算法有效性评价指标采用了 *NMI*(Normalized Mutual Information)，*F-measure* 和 *RI*(Rand Index)，共 3 个指标。这 3 个指标的值域均为 0 到 1，值越大表示聚类质量越好。在实验过程中，每个算法都运行 30 次，实验结果取平均值。

表 2 是多视图聚类集成算法对比实验结果。最右边一列是本文提出的 LMKCE 算法。从实验结果上可以看到，在多数情况下，LMKCE 算法取得了更好的实验结果。在 UCI digits 数据上对比其他算法有明显的提升，其次，在这 4 个数据集上都有更高的 *NMI* 值，这说明该算法是稳定且有效的。在 Reuters 数据集上，LMKCE 算法性能与 RMSC 算法接近，但仍高于其他算法。Reuters 数据集的维度高，且包含大量零值，是一个稀疏的文本数据集。本文提出的算法在这类数据集上表现不佳，仍有改进之处。

5.2 分布式算法性能实验

算法分布式性能实验在 Spark 集群上进行。该集群包含 12 台高性能计算机,其中 1 台作为 Master 节点,10 台作为 Worker 节点,1 台作为 Driver 节点。实验数据存放在 Hadoop 的分布式存储 HDFS 上。集群的基本信息如表 3 所列。

表 3　实验集群信息

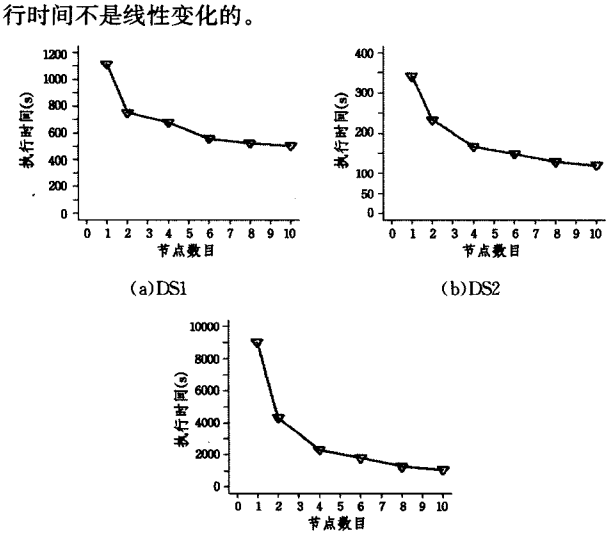
节点名称	硬件配置		
	CPU	内存	硬盘
Master	Inter Xeon E5506 4 核	12GB	1TB
Worker	Inter Xeon E5506 4 核	12GB	1TB
Driver	Inter Xeon E5506 4 核	12GB	1TB

实验数据采用了不同规模大小的数据样本 SUSY 和 YouTube Multiview Video Games Dataset 进行实验。SUSY 数据集包含 500 万条数据,在实验中,通过随机选择属性生成一个两视图的数据集。YouTube Multiview Video Games Dataset 是 YouTube 提供的真实的多视图数据集,包含 97935 条游戏视频数据,被分为 80%的训练数据、10%的测试数据和 10%的验证数据。在实验中,选择了 10%的验证数据集和 80%的测试数据集,去掉标签数据。以下称这 3 个数据集分别为 DS1,DS2,DS3。数据集的详细信息如表 4 所列。

表 4　分布式算法性能实验测试数据集信息汇总

数据集	样本个数	视图数	簇数	大小
DS1	5000000	2	2	573MB
DS2	12177	13	31	1.16GB
DS3	97935	13	31	9.33GB

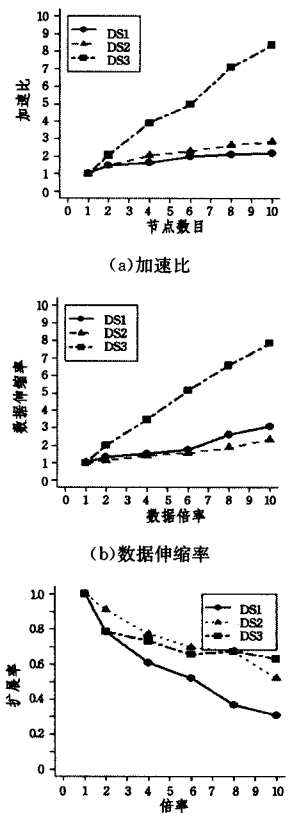
分布式聚类集成算法在各数据集上的执行时间如图 6 所示。从中可以明显看出,算法在单个节点上运行时消耗时间最长,随着节点个数的增加,算法的执行时间逐渐缩短。最开始增加节点,算法运行效率提升最明显,当节点增加到一定程度后,运行时间不再缩短。这是因为在分布式算法中会对输入数据进行分块,然后在每块数据上进行计算。当增加节点时,数据块被分配到了不同的节点上并行地进行处理,提升了程序效率。由于数据的最大分块数是一个固定的常数,当节点数目达到一定数目后,再增加节点,新增的节点不会再分配到数据,也就不会有计算,算法的运行效率也不会再有提升。由于集群的节点间的通信和 I/O 需要耗费一定时间,因此运行时间不是线性变化的。



(a)DS1　(b)DS2　(c)DS3

图 6　分布式算法在不同数据集上的执行时间

图 7(a)是 3 个数据集下的加速比。加速比衡量了在数据集规模不变的情况下,通过增加计算节点,计算程序在一个节点上的运行时间除以程序在多个节点运行时间的值的变化关系。从图中可以看到,对于数据集 DS1 和 DS2,加速比较小,在节点数增加到 6 时就趋于收敛了。而对于 DS3,加速比较大,在节点数到 10 时,仍然没有收敛。这说明对于不同数据集,并行算法的加速效率是不同的。当数据集较小时,并行算法的加速效率十分有限,而且只需要较少的节点即可。随着数据集规模的增大,算法加速效率显著提升,需要的计算节点也会随之增加。



(a)加速比　(b)数据伸缩率　(c)扩展率

图 7　分布式并行算法并行性能指标

分布式算法的数据伸缩率如图 7(b)所示。数据伸缩率指的是在保持计算节点数目不变的情况下,扩大数据,测试算法本身的复杂度。在实验中,分别将 3 个数据集分割为 6 个成倍增长的子数据集,然后在 10 节点上运行,统计运行时间。从图 7(b)中可以看到,DS1 和 DS2 的数据伸缩率较小,这是因为这两个数据集都较小,数据分块少,数据块被平均分配到各个节点,能够较快地完成计算。对于 DS1 来说,数据规模较大,伸缩率呈线性增长趋势,说明算法对大数据集有良好的并行性能。

图 7(c)展示了分布式算法的可扩展性。扩展率指的是在扩大数据的同时,增加计算机的数目,比较数据和节点同比增长下算法运行效率的变化情况。从图中可以看到,当数据集较小时,扩展率下降明显。对于 DS1 和 DS2,随着节点增加,扩展率迅速下降,这说明计算集群的计算资源充足,有很好的并行效果。对于 DS3,当同比增长到 6 倍后,扩展率开始持平,说明集群的计算资源得到了全部使用,算法有良好的可扩展性,能够有效处理大规模的数据。

结束语 本文首先提出了一种新的多视图聚类集成算法，克服了单一多视图 K-means 聚类受初始参数影响大的缺点，通过集成不同的聚类划分，充分利用多视图数据的互补性和一致性，提高了聚类的准确率、稳定性和鲁棒性。其次，实现的分布式多视图聚类集成算法有良好的并行性能，能够有效地对大规模多视图数据进行聚类，为下一步在大数据背景下进行聚类分析奠定了基础。

参 考 文 献

[1] KUMAR A, DAUMÉ H. A co-training approach for multi-view spectral clustering[C]// Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011: 393-400.

[2] Bickel S, Scheffer T. Multi-View Clustering [C]// ICDM. 2004: 19-26.

[3] KUMAR A, RAI P, DAUME H. Co-regularized multi-view spectral clustering[M]// Advances in Neural Information Processing Systems. 2011: 1413-1421.

[4] CAI X, NIE F, HUANG H. Multi-view k-means clustering on big data[C]// Proceedings of the Twenty-Third international Joint Conference on Artificial Intelligence. AAAI Press, 2013: 2598-2604.

[5] TZORTZIS G, LIKAS A. Kernel-based weighted multi-view clustering[C]// Proceedings of the 12th IEEE International Conference on Data Mining (ICDM). 2012: 675-684.

[6] XIIE X, SUN S. Multi-view clustering ensembles [C]// Proceedings of the IEEE 2013 International Conference on Machine Learning and Cybernetics (ICMLC). 2013: 51-56.

[7] MIZAEI H. A novel multi-view agglomerative clustering algorithm based on ensemble of partitions on different views[C]// 2010 20th International Conference on Pattern Recognition (ICPR). 2010: 1007-1010.

[8] STREHL A, GHOSH J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions[J]. The Journal of Machine Learning Research, 2003, 3: 583-617.

[9] IAM-ON N, BOONGOEN T, GARRETT S. Refining pairwise similarity matrix for cluster ensemble problem with cluster relations[M] // Discovery Science. Springer Berlin Heidelberg, 2008: 222-233.

[10] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51 (1): 107-113.

[11] ZHAO W, MA H, HE Q. Parallel k-means clustering based on mapreduce[M]// Cloud Computing. Springer Berlin Heidelberg, 2009: 674-679.

[12] CHEN W Y, SONG Y, BAI H, et al. Parallel spectral clustering in distributed systems[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(3): 568-586.

[13] LU Wei-ming, DU Chen-yang, Wei Bao-gang, et al. Distributed affinity propagation clustering based on map reduce[J]. Journal of Computer Research and Development, 2012, 49 (8): 1762-1772. (in Chinese)
鲁伟明，杜晨阳，魏宝刚，等. 基于 MapReduce 的分布式近邻传播聚类算法[J]. 计算机研究与发展，2012，49(8)：1762-1772.

[14] ZHAO Wei-dong, MA Hui-fang, FU Yan-xiang, et al. Research on Parallel k-means Algorithm Design Based on Hadoop Platform[J]. Computer Science, 2011, 38(10): 166-168. (in Chinese)
赵卫中，马慧芳，傅燕翔，等. 基于云计算平台 Hadoop 的并行 k-means 聚类算法设计研究[J]. 计算机科学，2011，38(10)：166-168.

[15] TANG Dong-ming. Affinity propagation clustering for big data based on Hadoop[J]. Computer Engineering and Applications, 2015, 51(4): 29-34. (in Chinese)
唐东明. 基于 Hadoop 的仿射传播大数据聚类分析方法[J]. 计算机工程与应用，2015，51(4)：29-34.

[16] AMINI M R, USUNIER N, GOUTTE C. Learning from multiple partially observed views- an application to multilingual text categorization[M]// Advances in Neural Information Processing Systems (NIPS). 2009: 28-36.

[17] XIA R, PAN Y, DU L, et al. Robust multi-view spectral clustering via low-rank and sparse decomposition[C]// AAAI Conference on Artificial Intelligence. 2014: 2149-2155.

(上接第 41 页)

[16] ZHANG C S, WANG F. A multilevel approach for learning from labeled and unlabeled data on graphs [J]. Pattern Recognition, 2010, 43(6): 2301-2315.

[17] YU H, SUN C, YANG W, et al. AL-ELM: One uncertainty-based active learning algorithm using extreme learning machine [J]. Neurocomputing, 2015, 166: 140-150.

[18] YONG Z, MENG J E. Sequential active learning using meta-cognitive extreme learning machine [J]. Neurocomputing, 2016, 173: 835-844.

[19] GU Y, JIN Z, CHIU S C. Active learning combining uncertainty and diversity for multi-class image classification [J]. IET Computer Vision, 2015, 9(3): 400-407.

[20] LONG B, BIAN J, CHAPELLE O, et al. Active learning for ranking through expected loss optimization[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(5): 1180-1191.

[21] HUANG S J, JIN R, ZHOU Z H. Active Learning by Querying Informative and Representative Examples [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36 (10): 1936-1949.

[22] HU L S, LU S X, WANG X Z. A new and informative active learning approach for support vector machine [J]. Information Sciences, 2013, 244(7): 142-160.

[23] HUANG G B, ZHU Q Y, SIEW C K. Extreme learning machine: Theory and applications [J]. Neurocomputing, 2006, 70 (1-3): 489-501.

[24] LIANG N Y, HUANG G B, SARATCHANDRAN P, et al. A fast and accurate on-line sequential learning algorithm for feedforward networks [J]. IEEE Transactions on Neural Networks, 2006, 7(6): 1411-1423.