一种基于 Divide-and-Merge 聚类算法的改进算法 *

黄智武 , 张东站 , 段江娇

(厦门大学信息科学与工程学院, 厦门 361005)

摘 要: BNAK-Divide-and-Merge 聚类算法是基于 David 等人提出的 Divide-and-Merge 算法的一种改进算法。Divide-and-Merge 算法是一种将自顶向下的分裂方法和自底向上的聚合方法相结合的聚类算法。虽然这个聚类算法已经通过众多实验表明其聚类的效率和质量,但是它在数据集很大的情况下分裂会很耗时间和空间资源,并且它需要阈值来确定聚类个数的方法也不是很理想。针对以上两个主要不足,对原算法进行改进。

关键词: 聚类算法; 分裂方法; 聚合方法; 时间和空间资源; 聚类个数

0 引 言

Divide-and-Merge 算法同时使用分裂和聚合方法解决了传统聚类算法不能实现重新修正聚类结果的问题,有着较明显的优势,不过仍然有以下三点不足:①在算法的分裂阶段所采用的谱聚类方法中, 通过寻找拉普拉斯矩阵的第二大特征向量, 再对其进行线性扫描从而找到最优二分割的方法并不是最佳的方法;②当算法运用到较大数据集时, 分裂阶段生成的二叉树将会变得很巨大,严重消耗了时间和空间资源;③在算法的合并阶段采用的方法所确定的聚类数并不理想。本文是基于 Divide-and-Merge 算法提出一个新的算法 BNAK-Divide-and-Merge,该算法旨在解决原算法的以上三点不足。通过相关实验表明本文的算法具有更高的效率和聚类质量,最后确定的聚类数也更加准确。

1 相关工作

如图 1 所示,Divide-and-Merge 聚类算法包括了分裂和聚合两个阶段。分裂阶段采用谱聚类算法[2]生成一棵树, 这棵树上的叶子都是由数据集的单个对象构成。紧接着在聚合阶段,从这棵树的叶子结点开始,将有些簇合并起来,从而得到比较理想的聚类结果。

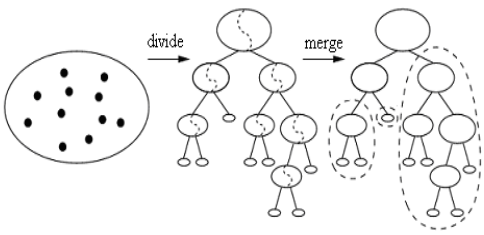

图 1 Divide-and-Merge 聚类算法示意图

谱聚类算法[12]作为一种根据顶点之间的权值对图进行划分的方法,其主要思想是:首先构造数据集的图模型建立对应的相似矩阵, 然后计算其拉普拉斯矩阵的前几个最大特征向量, 接着对这几个特征向量进行分析处理,从而将原来的多维空间转换成低维空间,最后在这个低维空间中再用传统经典的聚类方法进行聚类分析。本质上,它是一种基于矩阵特征向量提取新数据特征的方法。

Alpert 等人在文献[3]发现当拉普拉斯矩阵的所有特征向量全部被使用时, 最小割图划分问题等价于最大和向量划分问题。当所使用的特征向量数 d 小于特征向量总数 n 时,这个向量划分实例将是图划分的一个近似解。但是随着 d 的不断增大,这个实例所得的解将

* 基金项目:国家自然科学基金项目(No.50604012)
收稿日期:2010-04-09 修稿日期:2010-05-09
作者简介:黄智武(1985-),男,福建莆田人,硕士研究生,研究方向为数据挖掘

越来越接近于精确解。

Shi 等人在文献[4]鉴于随机游走的观点认为 $D^{-1}L$（其中 D 是相似矩阵的各行元素总和的向量矩阵）具有更加优越的理论基础，并且认为正则割谱聚类方法是一种比较好的谱聚类方法。Luxburg 等人在文献[5]也证明了当样本实例数目接近于无穷时，正则割谱聚类方法所得到的聚类结果会比其他谱聚类方法更好，所以他们建议首先使用正则割谱聚类算法。

Tan 等人在文献[6]提出个体簇的大小是由簇的有效性（比如簇的聚合度和分离度）决定。他们也认为一个具有高聚合度和分离度的簇会是个比较理想的簇，这种信息可以用来改进聚类的质量，也可以用来确定聚类簇个数。

2 BNAK-Divide-and-Merge 聚类算法

Divide-and-Merge 聚类算法有着几方面的缺陷，本文主要针对这三个不足，提出一个新算法 BANK-Divide-and-Merge，和原算法一样，它也有分裂和聚合两个阶段。分裂阶段的输入是一个已经给定相似度的数据集以及一个阈值 minSizeDivide，输出是一棵叶子结点由这个数据集子集构成的二叉树。这个阶段采用二路正则割谱聚类算法划分这个数据集，最后生成了这棵二叉树。这种分裂方法可以很好地应用于多维和高维数据集。例如 Iris 数据集[7]，20newsgroups 数据集[8]。而合并阶段也是对分裂阶段生成的树进行操作。但不同的是，我们用了另一种方法来确定聚类簇个数。这个阶段的输出是对象集的一个划分 $C_1,\cdots,C_k$，每个簇 C_i 都是分裂阶段生成的树的一个子结点。在这个阶段用动态规划的算法最优化 K-means 的目标函数从而得到最优的聚类结果。

2.1 分裂阶段

分裂阶段是 Divide-and-Merge 算法的第一个阶段，利用谱聚类算法对数据集进行分裂操作，生成一棵全部叶子结点由数据对象构成的二叉树。通过实验对其进行测试发现当对象集很大的时候，由于要彻底分裂数据集直到分裂树的每个叶子结点都是单对象为止，这个阶段产生的分裂树将会非常巨大，使得这个阶段的性能将会严重下降，又经证明彻底生成由单个对象样本构成的子结点并不是十分合理。所以，定义了一个阈值 minDividedSize 用来防止一些小结点的生成，从而极大提高了这一阶段的分裂效率。如图 2 所示，在算法的第二步，D 是一个对角矩阵，其对角线上的元素等于相似矩阵 $A\cdot A^T$ 对应的各行元素总和。

Input: An $n\times m$ matrix A and a threshold minDivideSize
Output: A tree whose leaves are subsets of the objects

1. If the size of A is not less than minDivideSize, then go to step 2, else stop.

2. Compute the Laplacian matrix of the data set: $L=D-A\cdot A^T$.

3. Compute the two smallest eigenvectors υ_1 and υ_2 of $D^{-1}L$, let $V=\{y_1,\cdots,y_n\}^T$ where $\upsilon=\{\upsilon_1,\upsilon_2\}$

4. Partition the samples $y_1,\cdots,y_n$ by k-means which k=2.

5. Let A_S,A_T be the submatrices of A. Recurse (Step 1-4) on A_S and A_T.

图 2 BNAK-Divide-and-Merge 的分裂阶段

Divide-and-Merge 算法的分裂阶段通过寻找拉普拉斯矩阵的第二大特征向量，再对其进行线性扫描从而找到数据集的最优二分割。因为由文献[3~5]已经证明可知，对比较多的拉普拉斯特征向量进行处理分析会比原算法只用一个特征向量的聚类效果好很多，再加上二分 K-means 聚类算法相对不太受质心初始化的影响，因此，通过不断衡量，最终选择使用拉普拉斯的二个特征向量进行分析处理。具体体现在在新算法中是先求出拉普拉斯矩阵前两个最小特征向量，再对这两个特征向量进行分析处理，将原来的高维空间转换为二维空间，最后对新数据集特征在这个二维空间中进行 K-means 聚类分析。通过大量的实验表明这种聚类方案与原方法相比能够较好地改进了分裂的精度。以上过程由下图中算法的第 3、4 步进行描述。第 5 步是用正则割将数据集 A 划分为 A_S 和 A_T 两个数据集然后递归对数据集按上述步骤对数据集不断进行分裂，直到树中的每个结点的大小小于所定义的阈值算法才结束分裂数据集。

2.2 合并阶段

Divide-and-Merge 聚类算法在这一阶段采用相关聚类算法对分裂阶段产生的层次结构进行合并操作，

并且最后动态确定了聚类的簇个数，相对其他一般的聚类方法更具实用性。但不足的是它必须先人为地设置一个阈值，导致确定的聚类个数不是很稳定。为了避免聚类个数受人为参数的影响，所以采用另一种不用事先设定参数的方法较好地确定了最后聚类的簇个数。合并阶段的算法如下图所示：

Input: The tree T which is produced by divide phase and whose leaves´ number is M

Output: A partition $C_1,\cdots,C_k$, where C_i is a node of T, $1\leqslant i\leqslant k$

1. for K=2 to M do
2. Compute

$$SSE(K)=\max_{x,y}\{SSE(K-1)+\Delta SSE(C_x,C_y)\}$$

where $\Delta SSE(C_x,C_y)$ is the changed value of SSE when C_x and C_y are merged

3. Merge clusters C_x and C_y.
4. repeat
5. Construct the curve K-SSE whose abscissa is the number of clusters (i.e.K) and ordinate is SSE.
6. Compute the best K and the partition $C_1,\cdots,C_k$ by observing the most obvious turning point of the K-TSS curve.

图 3 BNAK-Divide-and-Merge 合并裂阶段

文献[6]提到，簇的有效性可以通过聚合度和分离度两种量度标准进行衡量。其中，簇的聚合度(SSE)就是在每个簇内部的所有链接的权重之和。SSE 的定义如下：

$$SSE=\sum_{i=1}^{K}\sum_{x\in c_i}(x-c_i)^2 \qquad (1)$$

簇的分离度(SSB)是两两簇结点之间相互连接的权重总和。所以 SSB 是用来测量一个簇与其他簇的分离程度。SSB 的定义如下：

$$SSB=\sum_{i=1}^{K}\sum_{j=1,j\neq 1}^{K}(c_i-c_j)^2 \qquad (2)$$

在其些情况下，簇的聚合度与分离度有着很紧密的联系，所以有时计算它们的总和也具有实际的意义。定义 TSS 是计算每个点到数据的总均值的距离的平方和。具体地，可以证明总 SSE 和总 SSB 之和是一个常数，它等于总平方和(TSS)，可参考文献[6]。在 K 一定的情况下，最小化 SSE(凝聚度)等价于最大化 SSB(分离度)，所以选择了 SEE 度量来计算出比较理想的聚类个数 K。

通过大量的实验，发现通过观察 K-SEE 曲线的最明显拐点可以帮助确定比较理想的簇个数。随着簇个数的增加，直到在一个最明显的拐点之后 SEE 值会收敛于一个稳定值。其中 K-SEE 曲线的横坐标是簇的个数(即 K)，纵坐标是 TSS 度量。

3 实验结果与分析

3.1 实验数据

(1)Iris 数据集[7]包含 150 个 Iris 的信息，每 50 个构成 Iris 的一个种类，分别为以下 3 个种类：Setosa、Versicolour 和 Virginica.

(2)Wine 数据集[7]包括 178 个样本，每个样本有 13 个属性。这些样本可以划分成 3 个大类别，各自包含了 59、71 和 48 个样本。

(3)Glass 数据集[7]包含 214 个样本，每个样本有 9 个属性，总共可以分成 6 大类，每个类分别包括 70、17、76、13、9 和 29 个样本。

(4)20newsgroups 数据集[8]包含大约有 20,000 篇文章，可以将这些文章总共分成 20 个不同的新闻组。

3.2 实验结果

分别在 Iris、Wine 和 Glass 数据集上分别用 K-means、Divide-and-Merge (DAM) 和 BNAK-Divide-and-Merge(BNAKDAM)三个算法随机运行 20 次。从新闻组 talk.policics.mideast 和 talk.politics.misc 中随机选择 50 篇文章，同样用上面的 3 个算法随机运行 20 次。图表 1~4 分别是这 3 个算法的聚类结果之间的对比。其中，MDS、TNN、Time 和 Accuracy 分别指的是阈值(即 minDivideSize)，分裂阶段生成树结点的平均个数，平均运行时间，平均准确率。平均准确率定义如下：$Accuracy=\frac{\sum_{i=1}^{K^*}\max_{1\leqslant j\leqslant K}(e_{ij})}{N}$，其中 N 指的是数据集的大小，$K^*$ 指的是期望簇个数，K 指的是算法确定的聚类数，e_{ij} 指的是同时在第 i 个期望簇和在第 j 个实际簇的样本个数，$0\leqslant Accuracy\leqslant 1$。

表 1 在 Iris 数据集上的实验结果(MDS、TNN、Time 和 Accuracy)

Algorithm	MDS	Iris		
		TNN	Time(s)	Accuracy (%)
K-means	/	/	0.0094	83.667
DAM	2	299	21.609	88.667
BNAKDAM		299	20.246	97.333
BNAKDAM	10	42	0.79295	97.333
BNAKDAM	30	13	0.3235	97.333

表 2 在 Wine 数据集上的实验结果(MDS、TNN、Time 和 Accuracy)

Algorithm	MDS	Wine		
		TNN	*Time(s)*	*Accuracy (%)*
K-means	/	/	0.01025	69.663
DAM	2	355	35.9158	69.831
BNAKDAM		355	34.7642	72.427
BNAKDAM	10	71.5	1.2991	73.64
BNAKDAM	30	17.8	0.78495	73.034

表 3 在 Glass 数据集上的实验结果(MDS、TNN、Time 和 Accuracy)

Algorithm	MDS	Glass		
		TNN	Time(s)	Accuracy (%)
K-means	/	/	0.03025	57.85
DAM	2	427	50.8154	60.28
BNAKDAM		427	50.6012	66.382
BNAKDAM	10	69.6	2.125	66.822
BNAKDAM	30	23	0.814	66.981

表 4 在 20newsgroups 数据集上的实验结果(MDS、TNN、Time 和 Accuracy)

Algorithm	MDS	Talk.politics.mideast / talk.politics.misc		
		TNN	Time(s)	Accuracy (%)
K-means	/	/	0.312	63.311
DAM	2	99	84.046	68.235
BNAKDAM		99	81.641	70.126
BNAKDAM	10	14.5	8.4608	70.238
BNAKDAM	20	6.5	4.1327	70.137

3.3 实验分析

实验结果表明:①随着阈值(即 minDivideSize)的增大,分裂阶段生成的树结点总数以及算法的平均运行时间将会急剧地下降;②如图 4 所示,由 BNAK-Divide-and-Merge 确定的自然簇个数很好地吻和了原数据集的期望簇个数;③由上面四个表对 3 个算法进行聚类准确率比较表明 BNAK-Divide-and-Merge 在总体上会比 K-means 和 Divide-and-Merge 算法性能更好。

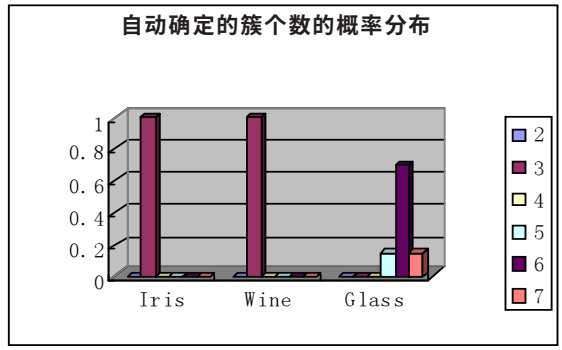

图 4 BNAK-Divide-and-Merge 自动确定的簇个数情况(MDS=30)

4 结 语

本文提出一种基于 David 等人的 Divide-and-Merge 聚类算法的改进算法 BNAK-Divide-and-Merge,通过各种实验表明它不仅很大提高了聚类的性能和效率,而且也很好地自动确定了聚类簇个数。

未来的研究可能会沿着以下几个方面继续展开:一方面将会继续研究一个更准确的数据集相似矩阵;另一方面将会寻找某个规则用来确定阈值的大小以最好地平衡分裂树的结点数大小和算法性能。

参考文献

[1]David Cheng, Ravi Kannan, A Divide-and-Merge Methodology for Clustering, In ACM New York, NY, USA,2006, Pages:1499-1525, 2007:37~65

[2]Ravi Kannan, Santosh Vempala, Adrian Vetta, On Clusterings: Good, Bad and Spectral, Journal of the ACM (JACM) Archive Volume 51, Issue 3 (May 2004) Table of Contents, Pages: 497~515

[3]Charles J. Alpert, So-Zen Yao. Spectral Partitioning: The More Eigenvectors, The Better, Design Automation, 1995. DAC '95. 32nd Conference

[4]Maila, M., Shi, J, A Random Walks View of Spectral Segmentation,International Conference on AI and Statistics (AISTAT), Key West, FL, January 4-7, 2001

[5]Von Luxburg, U., O. Bousquet M. Belkin: Limits of Spectral Clustering. Advances in Neural Information Processing Systems 17: Proceedings of the 2004 Conference, 857~864.(Eds.) Saul, L. K., Y. Weiss, L. Bottou, MIT Press, Cambridge,MA, USA (07 2005).

[6]Pang-Ning TAN. Michael Steinbach: Introduction to Data Mining, Published by Pearson Education, Inc., Publishing As Addison Wesley

[7]Blake CL, Merz CJ, UCI Machine Learning Repository of Machine Learning Databases.1998. http://www.ics.uci.edu/~mlearn/MLSummary. html.

[8]K.Lang. 20 Newsgroups Data Set. http://www.ai.mit.edu/people/jrennie/20newsgrups/

[9]Shi J B, Malik J, Normalized Cuts and Image Segmentation, IEEE Transaction on Pattern Analysis and Macine Intelligence, 2000, 22(8):888~905

[10]Fan R. K. Chung, Spectral Graph Theory, AMS Bookstore, ISBN 0821803158, 9780821803158:2-5

[11]H. Zha, X.He, C. H.Q. Ding, M. Gu, H. D. Simon. Spectral Relaxation for K-means Clustering. In T. G. Dietterich,S. Becker, and Z.Ghahramani, NIPS, pages 1057~1064.MIT Press,2001

[12]Shi J B, Malik J. Normalized Cuts and Image Segmentation. IEEE Trasaction on Pattern Analysis and Machine Intelligence, 2000, 22(8):888~905

An Improved Algorithm Based on Divide-and-Merge Clustering Algorithm

HUANG Zhi-wu , ZHANG Dong-zhan , DUAN Jiang-jiao

(School of Information Science and Technology, Xiamen University, Xiamen 361005)

Abstract: BNAK-Divide-and-Merge clustering algorithm is an improved algorithm which is based on the Divide-and-Merge clustering algorithm proposed by David et al. Divide-and-Merge is a methodology which combines a top-down divide method with a bottom-up merge method. Although it has been proved to be a method with high efficiency and quality of clustering by implementing lots of relevant experiment, its divide phase will consume too much time and space resources when it is applied to very huge sets; furthermore the method which can figure out the number of clustering with a threshold is also not best. Accordingly, improves the original algorithm to overcome the two major flaws mentioned above.

Keywords: Clustering Algorithm; Divide Method; Merge Method; Time and Space Resources; Clustering Number