ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.11, November 2010, pp.2802−2813
doi: 10.3724/SP.J.1001.2010.03677

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

一种基于语料特性的聚类算法*

曾依灵[1,2+], 许洪波[1], 吴高巍[1], 白 硕[1]

[1](中国科学院 计算技术研究所 网络重点实验室,北京 100190)

[2](中国科学院 研究生院,北京 100049)

Clustering Algorithm Based on the Distributions of Intrinsic Clusters

ZENG Yi-Ling[1,2+], XU Hong-Bo[1], WU Gao-Wei[1], BAI Shuo[1]

[1](Key Laboratory of Network Science and Technology, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100190, China)

[2](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: E-mail: zengyiling@software.ict.ac.cn, http://www.ict.ac.cn

Zeng YL, Xu HB, Wu GW, Bai S. Clustering algorithm based on the distributions of intrinsic clusters. *Journal of Software*, 2010,21(11):2802−2813. http://www.jos.org.cn/1000-9825/3677.htm

Abstract: In finding a flexible approach to solve the model misfit problem, a clustering algorithm based on the distributions of intrinsic clusters (CADIC) is proposed, which implicitly integrates distribution characteristics into the clustering framework by applying rescaling operations. In the clustering process, a set of discriminative directions are chosen to construct the CADIC coordinate, under which the distribution characteristics are analyzed in order to design rescaling functions. Along every axis, rescaling functions are applied to implicitly normalize the data distribution such that more reasonable clustering decisions can be made. As a result, the reliability of clustering decisions is improved. The time complexity of CADIC remains the same as *K*-means by using a *K*-means-like iteration strategy. Experiments on well-known benchmark evaluation datasets show that the framework of CADIC is reasonable, and its performance in text clustering is comparable to that of state-of-the-art algorithms.

Key words: CADIC (clustering algorithm based on the distributions of intrinsic clusters); text clustering; model misfit; rescaling; information retrieval

摘 要: 为寻求模型不匹配问题的一种恰当的解决途径,提出了基于语料分布特性的 CADIC(clustering algorithm based on the distributions of intrinsic clusters)聚类算法.CADIC 以重标度的形式隐式地将语料特性融入算法框架,从而使算法模型具备更灵活的适应能力.在聚类过程中,CADIC 选择一组具有良好区分度的方向构建 CADIC 坐标系,在该坐标系下统计固有簇的分布特性,以构造各个坐标轴的重标度函数,并以重标度的形式对语料分布进行隐式的归一化,从而提高聚类决策的有效性.CADIC 以迭代的方式收敛到最终解,其时间复杂度与 *K*-means 保持在同一量级.在国际知名评测语料上的实验结果表明,CADIC 算法的基本框架是合理的,其聚类性能与当前领先水平的聚类

* Supported by the National Natural Science Foundation of China under Grant No.60933005 (国家自然科学基金); the National Basic Research Program of China under Grant Nos.2007CB311100, 2004CB318109 (国家重点基础研究发展计划(973)); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z441 (国家高技术研究发展计划(863))

Received 2008-10-22; Revised 2009-03-05; Accepted 2009-07-07

算法相当.

关键词: CADIC(clustering algorithm based on the distributions of intrinsic clusters);文本聚类;模型不匹配;重标度;信息检索

中图法分类号: TP18 **文献标识码**: A

随着互联网内容的指数增长,对有效管理大规模文档的需求日益急切.聚类作为一种重要的文本分析方法,已在信息检索领域得到了广泛的应用.它被用于加速信息检索过程[1]、提高信息检索系统的准确率和召回率[2]以及组织用户的查询结果[3].近些年,随着话题发现与跟踪(topic detection and tracking,简称 TDT)技术[4]的发展,文本聚类被广泛用于发现和提取大规模文档中的话题.由此,文本聚类也成为向用户提供个性化信息服务以及相关部门追踪重要情报的重要技术.

然而,聚类算法的性能常常受到模型不匹配问题的影响.大部分聚类算法都基于一些潜在的模型,当语料特性恰好满足聚类策略的潜在模型时,算法性能良好,反之则性能欠佳.例如,*K*-means 算法的潜在假定是,属于同一簇的文档分布在特征空间的一个球形区域,该区域的球心即为该簇的中心,不同簇所在的球形区域半径相同.然而,真实语料很少满足如此理想且严格的假定,因此,当 *K*-means 的基本假定遭到破坏时,*K*-means 算法的性能会在不同程度上受模型不匹配问题的影响.

通常,解决模型不匹配问题可以采取两种策略:(1) 调整算法模型以适应语料分布[5,6];(2) 变换特征空间以修正文本表示层的固有问题[1,7−10].其中,第 1 种策略往往通过训练错误进行局部修正,但容易忽略语料分布的全局特性;第 2 种策略往往通过空间变换来解决表示空间的固有问题,却很少考虑算法模型.在本文中,我们试图构造一种基于语料全局分布特性的隐式空间变换,并将这种隐式变化融入算法框架,以形成一种基于语料特性的聚类算法,从而同时达到立足全局特性和面向算法模型的需求.

本文提出的基于语料特性的聚类算法称作 CADIC 算法(a clustering algorithm based on the distributions of intrinsic clusters,简称 CADIC).与大多数聚类算法忽略聚类模型与语料特性之间的差异相反,CADIC 自动分析语料中所有固有簇的分布特性,并以重标度的方式将其隐式地融入算法框架,以使算法在面对不同分布的语料时具备更为灵活的适应能力.

为了更好地进行聚类判别,CADIC 借鉴 Fishier 判别的思想,首先选择一组具有良好区分度的方向构建一个具有良好判别能力的坐标系(CADIC 坐标系).在该坐标系下,CADIC 分析各个固有簇的分布特性,并根据这些特性构建重标度函数.通过重标度函数的隐式映射,使语料在新的标度下分布更为理想,从而聚类策略也更为有效.由此,将语料特性隐式地融入了算法框架.CADIC 以迭代的方式收敛到最终解,其算法的时间复杂度与 *K*-means 保持在同一量级.不同标准测试集上的比较实验结果表明,CADIC 的性能相当优秀,与当前处于领先水平的聚类算法性能相当.

本文第 1 节首先分析不同聚类算法的隐含假定,并介绍关于模型不匹配的已有研究.在第 2 节中,我们提出自己的 CADIC 聚类算法.在第 3 节中,我们组织实验验证 CADIC 算法的真实性能.第 4 节对全文进行总结.

1 相关工作

根据不同的聚类模型,经典聚类[1]算法可大致地分为层次式聚类、划分式聚类、基于网格的聚类以及基于密度的聚类.不同的聚类算法面对语料分布采取不同的基本假定.层次式聚类的基本假定是,任何聚类簇都是由更小的、在语意上类似的子簇或者子话题构成,在一个语料集中,处于底层的最小子簇只包含 1 个文档,处于顶层的则是一个包含所有文档的大簇.划分式聚类,以 *K*-means 为例,则认为各个簇满足方差相同的、各向同性的高斯分布.由此,在空间特性上,不同簇分布在半径相同的球形区域内,因而在作聚类决策时,只需计算当前文档到每个簇中心的距离,并将其放入最近的簇中.基于密度的聚类和基于网格的聚类则关注于语料分布的局部特性,并用这些特性进行聚类判别.其中,密度聚类的基本假定是:如果一个点的ε-邻域所包含的邻居数大于某个阈值,即认为它处于某个簇的内部,否则处于簇的边缘.基于网格的聚类则把数据空间划分为小的单元,所有的聚

类操作都是基于这些小单元进行,最终组织出聚类结果(如图 1 所示).

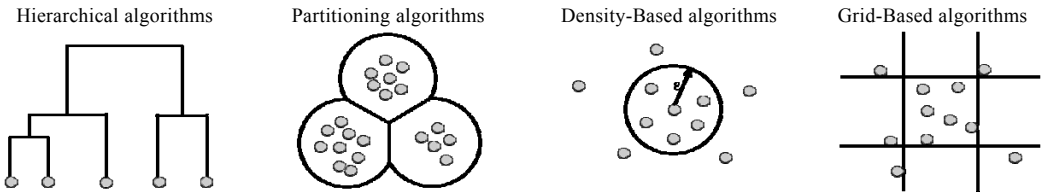


Fig.1 Illustrations of assumptions for different clustering algorithms

图 1 不同聚类算法的假设模型

而事实上,经典聚类算法的假定模型过于理想和严格,以至于真实语料或多或少和这些基本假定有所出入.因此.模型不匹配问题成为文本挖掘领域的一个常见问题.为解决这个问题,研究人员在如下两个方面寻求各种解决途径.

一方面的解决途径是通过修正算法模型来适应语料分布.这方面的研究通常需要基于训练错误进行,因此集中在分类领域.典型的研究有:Wu 等人[5]根据训练错误,用同样的学习方法对每一个类的训练样本重新训练一个子分类器,强迫子分类器根据训练语料学习需要修正的区域,从而使得分类模型更适应当前语料特性.Tan[6]则提出一种简洁有效的"拉推"策略,通过在线修正分类模型的方式提高分类结果:每一个分类错误的文档都将正确的类中心拉向自身,而将错误的类中心推向相反方向,从而使得正确的类中心离自己更近,而错误的类中心离自己更远.修正之后,被错分的文档也就更容易被重新分到正确的类中.在聚类领域,由于无法通过标签获取信息,关于模型不匹配问题的研究较为罕见.总体而言,这一类的研究通过局部修正来提高性能,但容易忽略语料的全局分布.

另一方面的解决途径是通过空间映射,使原空间的相关问题在新空间得以恰当的解决.典型的研究有:Dumais[1]认为,向量空间模型(vector space model,简称 VSM)的一大问题是用非独立的词(term)作为独立的空间维度,于是他们提出 LSI 以解决此问题导致的空间缺陷,通过 LSI 分解,文档与文档以及词与词的相关性能在 k 维表示下进行恰当的计算.核方法[7]是另一种通过空间变换来解决问题的重要方法,此类方法通过核函数 $k(x,y)=\phi(x)\cdot\phi(y)$来寻求一种隐式的空间映射$\phi$,使得在新的空间中问题更易解(比如线性可分).特别地,在聚类领域,近些年流行一种以空间映射的方式来解决聚类问题的方法,称为谱聚类.谱聚类的基本思想是,将整个语料看作一个加权图,用图划分的方式将语料划成一个个聚类,并最优化给定的划分权重(如 Normalized Cut[8],Ratio Cut[9]及 Min-Max Cut[10]).这些最优化划分目标可以通过特征分解的方式获得,特征空间也因此降到了 k 维(k 为聚类簇的个数).由于划分目标能够保证全局最优,谱聚类的性能往往好过传统聚类算法.大致而言,这一类研究专注于解决当前特征空间的固有问题,其中大部分研究并未考虑算法的固有假定,而空间变换往往意味着较高的时间复杂度和较大的存储空间.

在下一节中,我们提出自己的解决框架,通过提取语料全局特性并隐式地融入算法模型,形成一种新的聚类算法,以期结合如上两方面研究的优势来提高聚类性能.

2 CADIC 聚类算法

2.1 聚类中的模型不匹配

如上一节所述,在算法隐含的假定模型与语料的分布特性之间通常存在着差异,这种现象称作模型不匹配.以 *K*-means 算法为例,一个简单明了的关于模型不匹配的例子如图 2 所示.假定语料集中存在两个固有簇,固有簇中的点分别由"+"和"○"标识,并由虚线圈出.标为"+"的点分布在一个狭长的椭球形区域内,而标为"○"的点分布在一个标准的球形区域内.显然,语料的分布特性破坏了 *K*-means 的潜在假设.由于无法获取语料的分布信息,*K*-means 将按照自己的假定进行划分,即沿图中实线标识的球形将所有点分成两个簇.于是,部分标记为"+"

的点被错误地与标记为“○”的点划分在一起.

问题是,如何才能避免如图 2 所示的错误划分.根据 *K*-means 的决策准则,一个数据点将被放到最近的簇中.以图 2 中的 x 为例,假定 x 到 cluster 1 的簇中心的距离为 d_1,到 cluster 2 的簇中心的距离为 d_2.尽管 x 属于标记为“+”的固有簇,由于 $d_2<d_1$,x 还是被判入 cluster 2.为了避免这样的错误,我们需要根据一定的策略进行空间变换,使得在新的场景中 $d_1<d_2$.也就是说,如果我们将左边椭球形的固有簇转化成一个和右边一样的标准球形区域(如图 3 所示),那么距离的度量将更为合理,x 将更可能被放入 cluster 1 中.

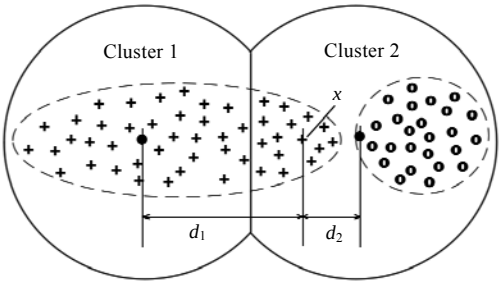


Fig.2 Model misfit of *K*-means

图 2 *K*-means 的模型不匹配问题

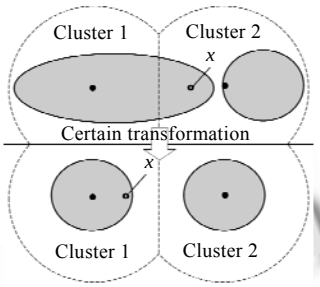


Fig.3 Scenario transformation

图 3 场景转换

2.2 CADIC基本框架

基于如上对模型不匹配问题的分析,为了使距离度量更为合理,可以将每一个簇都归一化到一个标准的球形区域.然而,文本空间的维度灾难问题和数据稀疏特性使得这样的归一化操作不切实际.对此,CADIC 提出了自己的解决框架,该框架分为如下两个步骤:

(1) 选择一系列具有区分度的方向构成一个新的坐标系;

(2) 根据各个固有簇的特性对新的坐标系进行重标度(rescaling),以使距离更为合理.

在上述的基本框架下,我们通过迭代的方式不断提高聚类划分的精度,直至算法收敛到最终解.

为了便于后文的叙述,我们先作一些约定:对于一个包含 n 篇文档共包含 k 个类/簇的语料集,我们将其中的类/簇分别标记为 $C_1,C_2,\ldots,C_i,\ldots,C_k(1\leqslant i\leqslant k)$,它们对应的文档数分别为 $n_1,n_2,\ldots,n_i,\ldots,n_k$.其中,$\boldsymbol{m}_i=\frac{1}{n_i}\sum_{\boldsymbol{x}\in C_i}\boldsymbol{x}$ 是 C_i 的中心,$\boldsymbol{m}=\frac{1}{n}\sum_{\boldsymbol{x}}\boldsymbol{x}$ 是整个语料的中心.

回想经典的 Fisher 线性判别[12],对于多类别情况,算法通过最大化类间离散度和最小化类内离散度来求解最有区分度的方向,其最优化准则函数为

$$J(\boldsymbol{W})=\frac{|\boldsymbol{W}^t\boldsymbol{S}_B\boldsymbol{W}|}{|\boldsymbol{W}^t\boldsymbol{S}_W\boldsymbol{W}|} \tag{1}$$

其中,

$$\boldsymbol{S}_B=\sum_{i=1}^{k}n_i(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^t \tag{2}$$

$$\boldsymbol{S}_W=\sum_{i=1}^{k}\sum_{\boldsymbol{x}\in C_i}(\boldsymbol{x}-\boldsymbol{m}_i)(\boldsymbol{x}-\boldsymbol{m}_i)^t \tag{3}$$

分别为语料的类间散布矩阵和类内散布矩阵,$\boldsymbol{W}$ 即为所要求的最有区分度的方向构成的矩阵,$|\boldsymbol{W}^t\boldsymbol{S}_B\boldsymbol{W}|$表示在判别方向张成空间中的类间离散度,$|\boldsymbol{W}^t\boldsymbol{S}_W\boldsymbol{W}|$表示在判别方向张成空间中的类内离散度.

在上述最优化准则中,需要同时对类间离散度和类内离散度进行最优化.而在我们提出的解决框架中,构建新的坐标系之后将进一步根据固有簇的特性执行重标度操作,重标度以隐式映射的方式对不同簇的分布进行归一化,以使距离度量更为合理.这种隐式的归一化旨在消除不同簇分布上的差异.以图 3 作为直观例子,重标度

操作之前,两个簇的分布存在着明显的差异;通过合理的重标度操作,两个簇的分布差异性得以消除,不同簇内的数据分布大致相当.在这样归一化的场景下,类内离散度这个反映类内数据点散布程度的指标得以相应的规范,由类内离散度造成的区分度影响也大为降低.因此,在求解具有区分度方向的过程中,我们将优化准则函数简化,忽略类内离散度,仅考虑类间离散度,最优化准则函数变为

$$J(\boldsymbol{W})=|\boldsymbol{W}^t\boldsymbol{S}_B\boldsymbol{W}|=\left|\boldsymbol{W}^t\sum_{i=1}^{k}n_i(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^t\boldsymbol{W}\right| \tag{4}$$

根据与 Fisher 判别类似的推导过程,最优矩阵 $\boldsymbol{W}$ 的列向量可以通过求解 $\boldsymbol{S}_B$ 的最大特征值对应的特征向量,即求解 $\boldsymbol{S}_B\boldsymbol{w}_i=\lambda_i\boldsymbol{w}_i$ 而得到.由公式(2)$\boldsymbol{S}_B$ 的定义可知,这些特征向量所张成的空间就是向量 $\boldsymbol{m}_i-\boldsymbol{m}(1\leqslant i\leqslant k)$张成的空间.因此,我们不妨直接选择 $\boldsymbol{m}_i-\boldsymbol{m}(1\leqslant i\leqslant k)$构成一个新的坐标系.

定义 1(CADIC 坐标系). 给定一个包含 k 个固有簇的语料集,各个簇的中心和语料整体中心的定义如前文所述.那么,可由方向 $\boldsymbol{m}_1-\boldsymbol{m},\ldots,\boldsymbol{m}_i-\boldsymbol{m},\ldots,\boldsymbol{m}_k-\boldsymbol{m}(1\leqslant i\leqslant k)$构成一个坐标系,该坐标系称为 CADIC 坐标系.对于原空间数据点 $\boldsymbol{x}_j(1\leqslant j\leqslant n)$,其在 CADIC 坐标系中的坐标标记为 $(\boldsymbol{x}^c_{j,1},\ldots,\boldsymbol{x}^c_{j,i},\ldots,\boldsymbol{x}^c_{j,k})$,其中,

$$\boldsymbol{x}^c_{j,i}=(\boldsymbol{x}_j-\boldsymbol{m})^t\frac{(\boldsymbol{m}_i-\boldsymbol{m})}{\|\boldsymbol{m}_i-\boldsymbol{m}\|_2} \tag{5}$$

尽管 CADIC 坐标系的坐标轴选择从数学上未必是最佳的(可以通过对这些方向正交化以获取更为理想的坐标系),但却具备有利于重标度工作的重要性质:首先,$\boldsymbol{m}_1-\boldsymbol{m},\ldots,\boldsymbol{m}_i-\boldsymbol{m},\ldots,\boldsymbol{m}_k-\boldsymbol{m}$ 由最优化公式(4)求得,因此,其张成的空间在整体上对各个簇具有较好的区分性;其次,对于其中每一个单独的方向,如 $\boldsymbol{m}_i-\boldsymbol{m}$,它也能以最大化类间离散度的方式区分簇 C_i 与其他簇,这是因为:如果将簇 C_i 看成一个类,将除簇 C_i 以外的所有簇看成另一个类(记作 $C_{i'}$,其中心记为 $\boldsymbol{m}_{i'}$),根据二类情况下[12]类间散布矩阵的定义 $\boldsymbol{S}_B=(\boldsymbol{m}_i-\boldsymbol{m}_{i'})(\boldsymbol{m}_i-\boldsymbol{m}_{i'})^t$,最优化准则函数(4)的方向(标记为 $\hat{\boldsymbol{d}}_i$)即为过两个类中心的方向 $\boldsymbol{m}_i-\boldsymbol{m}_{i'}$.可以证明,该方向与 $\boldsymbol{m}_i-\boldsymbol{m}$ 是同一个方向,详细如下:

$$\hat{\boldsymbol{d}}_i=\boldsymbol{m}_i-\boldsymbol{m}_{i'}=\boldsymbol{m}_i-\frac{1}{n-n_i}\sum_{\boldsymbol{x}\notin C_i}\boldsymbol{x}=\boldsymbol{m}_i-\frac{n\boldsymbol{m}-n_i\boldsymbol{m}_i}{n-n_i}=\frac{n}{n-n_i}(\boldsymbol{m}_i-\boldsymbol{m}).$$

因此,CADIC 坐标系的每一个方向对应着一个固有簇,该方向能够以最大化类间离散度的方式区分当前簇和其他簇.于是,在进一步的重标度操作中,对于每一个方向,可用对应簇的相关特性作为该维度上重标度操作的重要参照.

定义 2(CADIC 距离). 假定 CADIC 坐标系下各个坐标轴对应的重标度函数分别为 $R_1(\cdot),\ldots,R_i(\cdot),\ldots,R_k(\cdot)$,那么,其下的距离函数可写成

$$d_C(\boldsymbol{x}_i-\boldsymbol{x}_j)=\sqrt{\sum_{t=1}^{k}(R_t(\boldsymbol{x}^c_{i,t})-R_t(\boldsymbol{x}^c_{j,t}))^2} \tag{6}$$

公式(6)给出了 CADIC 坐标系中距离的一般形式.根据语料中固有簇的分布特性,我们可以采取不同的重标度函数,以使公式(6)给出的距离更为合理.重标度函数可以取线性函数,也可以取非线性函数.如果重标度函数为线性函数,也就是说,$R_i(\cdot)$的形式为 $R_i(x)=\xi_i x+\ell_i$,那么公式(6)可以进一步写成

$$d_C(\boldsymbol{x}_i-\boldsymbol{x}_j)=\sqrt{\sum_{t=1}^{k}((\xi_t\boldsymbol{x}^c_{i,t}+\ell_t)-(\xi_t\boldsymbol{x}^c_{j,t}+\ell_t))^2}=\sqrt{\sum_{t=1}^{k}(\xi_t(\boldsymbol{x}^c_{i,t}-\boldsymbol{x}^c_{j,t}))^2} \tag{7}$$

其中,$\xi_i(1\leqslant i\leqslant k)$可看作各个坐标轴的重标度系数,通过它对各个维度的坐标值进行扩放或者收缩,以使各个维度的标度更具可比性.

如何使各个维度在重标度之后更具可比性?如前文所述,不同的簇有着不同的分布,各个簇在对应坐标轴上的投影也有着不同的分布.我们期望各个坐标轴有可比较的标度,也就是说,各个坐标轴上点的分布有着相当的疏密程度.一个较为直观、合理的解决方式是用一个与各个坐标轴数据点分布疏密程度相关的统计量作为该坐标轴的重标度系数.在所有的统计量中,标准差是一个反映一组数据散布尺度的统计量.因此,我们可以计算固有簇投影分布的标准差,并将其倒数作为对应轴的重标度系数,即 $\xi_i=1/\sigma_i$,以使各个轴语料的散布尺度趋于一致.于是,公式(7)可以表示如下:

$$d_C(\boldsymbol{x}_i-\boldsymbol{x}_j)=\sqrt{\sum_{t=1}^{k}\left(\frac{\boldsymbol{x}_{i,t}^c-\boldsymbol{x}_{j,t}^c}{\sigma_t}\right)^2} \tag{8}$$

其中,

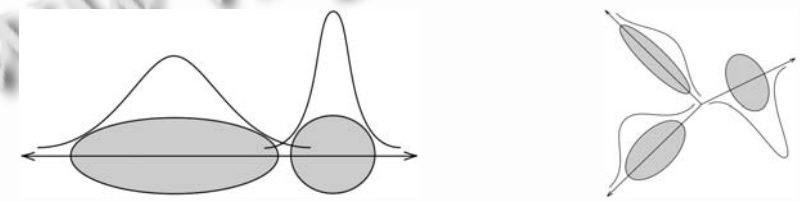
$$\sigma_t=\sqrt{\frac{1}{n_t}\sum_{\boldsymbol{x}\in C_t}(\boldsymbol{x}_{j,i}^c-\|\boldsymbol{m}_t-\boldsymbol{m}\|_2)^2}=\sqrt{\frac{1}{n_t}\sum_{\boldsymbol{x}\in C_t}\left((\boldsymbol{x}_j-\boldsymbol{m})^t\frac{(\boldsymbol{m}_t-\boldsymbol{m})}{\|\boldsymbol{m}_t-\boldsymbol{m}\|_2}\right)^2-(\boldsymbol{m}_t-\boldsymbol{m})^t(\boldsymbol{m}_t-\boldsymbol{m})} \tag{9}$$

通过以标准差为倒数的重标度函数,各个固有簇在对应坐标轴上的投影实现了等方差的归一化,各个坐标轴由此而具有更为可比的尺度,基于此尺度的距离计算也更为合理.需要指出的是,本文采用的标准差倒数只是众多重标度函数中的一个特例,我们完全可以根据语料的特定分布情况构造各种更精细、更合理的重标度函数,以使语料能够更好地转化到更为理想的形式.

显然,基于 CADIC 坐标系的重标度也可看作一种隐式的空间变换,它根据语料分布特性进行尺度变换,目标是使语料转换到更理想的分布,从而获得更合理的聚类划分.

定义 3(CADIC 基本框架). CADIC 坐标系以及定义在此上的基于重标度的 CADIC 距离,形成了一套独有的融入了语料分布特性的聚类框架,我们称其为 CADIC 基本框架.

CADIC 基本框架指出了一种通过空间映射和重标度来归一化语料分布的聚类算法解决框架.该框架隐式地将语料特性融入算法模型,从而使聚类判别具备更灵活的适应能力.关于该框架的基本原理,图 4 中给出了两个例子.图 4(a)中给出的语料包含两个分布各异的固有簇,两个簇的边界显然更靠近右边的簇.如果以重标度的方式按照标准差归一化两个簇的分布,在新的标度下,它们的边界将靠向两个簇中心的中点.因此,在新标度下,距离度量更为合理.图 4(b)给出了一个包含多个固有簇的语料例子,同理,在 CADIC 坐标系下,语料分布沿着各个坐标轴通过重标度进行归一化,聚类判别自然也更为准确.



(a) A dataset with two intrinsic clusters
(a) 具有两个固有簇的语料

(b) A dataset with multiple intrinsic clusters
(b) 具有多个固有簇的语料

Fig.4 An Illustration of how CADIC framework cope with different datasets

图 4 CADIC 基本框架对不同语料的处理

当前未解决的一个重要问题是:我们如何获取固有簇的信息.由于没有类标签的帮助,我们无法获取固有簇的精确划分.然而,如果能够获得一个大致反映固有簇分布的初始划分,我们就能统计出粗略反应各个簇分布特性的重标度系数.因此,我们的解决方案是,通过调用 *K*-means 的两三次迭代获取一个初始划分,将其作为 CADIC 算法的输入.根据该初始划分,可建立 CADIC 坐标系,并计算出各个轴的重标度系数.根据基于重标度的 CADIC 距离,可以将所有文档重新划分到最近的簇内,以得到更为合理的划分结果.在新划分结果的基础上,可继续构造出更为合理的 CADIC 坐标系并计算出更为准确的重标度系数,并再一次进行更为精细的重划分.为此,我们将迭代策略引入 CADIC 算法.通过迭代,能够促使聚类结果和 CADIC 坐标系相互修正,并最终获得一个令人满意的聚类结果.

2.3 CADIC聚类算法

对于一个包含 *n* 个点的给定语料集,CADIC 聚类算法工作的详细流程如下:

CADIC 聚类算法.

1. 迭代 *K*-means 算法 *r* 次生成语料的初始划分,将该初始划分作为 CADIC 算法的输入;

2. For i=1 to k do:

2.1. 重新计算簇 C_i 的中心;

3. 根据新的簇中心重新构造 CADIC 坐标系,根据公式(9)计算各个坐标轴的重标度系数;

4. For i=1 to n do:

4.1. 根据公式(5)计算 $\boldsymbol{x}_i$ 在新的 CADIC 坐标系下的坐标;

4.2. 初始化最近簇和最近距离:*Nearest_Cluster* =*NULL*,*Nearest_Distance*=∞;

4.3. For j=1 to k do:

4.3.1. 根据公式(8)计算点到 $\boldsymbol{x}_i$ 簇 C_j 中心的距离 $d_C(\boldsymbol{x}_i,\boldsymbol{m}_j)$;

4.3.2. 如果 $d_C(\boldsymbol{x}_i,\boldsymbol{m}_j)$<*Nearest_Distance*:

4.3.2.1. 相应修改最近簇和最近距离:*Nearest_Distance*=$d_C(\boldsymbol{x}_i,\boldsymbol{m}_j)$; *Nearest_Cluster*=$C_j$;

4.4. 将 $\boldsymbol{x}_i$ 划入 *Nearest_Cluster* 代表的簇中;

5. 重复步骤 2~步骤 4,直到簇与簇之间不再存在数据点的迁移,或者已达到最大迭代次数为止.

在上述算法中,参数 r 控制 *K*-means 迭代次数,为保证初始划分的迅速产生,r 的取值通常很小(r=2 或 3).显然,第 1 步中,通过 *K*-means 迭代产生初始划分的时间复杂度为 $O(krn)$,其中,k 为簇的个数,r 为迭代次数,n 为数据点的个数.在 CADIC 的每一次迭代中,存在 3 种主要运算:重新构建 CADIC 坐标系、计算每个点在新 CADIC 坐标系下的坐标以及根据 CADIC 距离公式计算点到所有簇中心的距离.每次新的迭代都将重新构建 CADIC 坐标系,对于每个坐标轴,都需要计算语料中心到簇中心的归一化向量,所需时间为 $O(k)$;接着,需要计算所有点在新 CADIC 坐标系下的坐标,对于每个点,将根据公式(5)计算 k 个点乘,因此,计算所有点的 CADIC 坐标的时间复杂度为 $O(kn)$.接着,将在 CADIC 坐标系下根据公式(8)计算所有点到簇中心的 CADIC 距离,总的计算量显然为 $k\times n$,但由于每个距离计算只涉及 k 维,与原始空间相比,维度大为降低,因此,与前面的高维计算相比,这部分的计算量可以省略.将 3 部分运算汇总,一次迭代的总运算量为 $O(k)+O(kn)=O(kn)$.假定 CADIC 总的迭代次数为 T(包括用 *K*-means 产生初始划分的 r 次迭代),CADIC 聚类算法的时间复杂度为 $O(knT)$.可见,CADIC 聚类算法的时间复杂度保持在与 *K*-means 算法同一量级.

3 实验结果

本节内容如下:第 3.1 节介绍实验中将用到的语料集;第 3.2 节介绍实验中将用到的聚类结果评价指标;在第 3.3 节中,我们分析语料分布的差异性,以验证本文提出的 CADIC 框架的有效性;在第 3.4 节中,我们设计实验来验证 CADIC 算法的真实性能.

3.1 语料集

在本节的实验中,我们将用到在分类聚类研究里常用的两个国际知名文本语料库:RCV1-v2[13]和 20Newsgroup(http://www.ai.mit.edu/people/jrennie/20Newsgroups/).

RCV1-v2 语料集包含人工采集的来自路透社的 800 000 余篇新闻专线语料,存储在 103 个类别中.我们从 RCV1-v2 中随机地挑选类别及文档,构造出了一系列的测试语料集:R1,R2,R3,R4,R5 和 R6,它们的类别数从 5 类到 15 类不等,文档数从 1 305 到 3 330 不等.

20Newsgroup 语料集包含了来自 20 个 Usenet 新闻组中的 20 000 篇文章,每个类别 1 000 篇.我们随机挑选 20Newsgroup 中的类别及文档,构造出了一系列测试语料集:N1,N2,N3,N4,N5 和 N6,它们的类别数从 4 类到 15 类不等,文档数从 1 264 到 3 406 不等.

我们所构造的两个系列的测试语料集的概况见表 1.在这两个系列的语料上,我们都保持了类别数和文档数的跨度,其原因是,我们想观察 CADIC 在不同尺度的语料集上的性能;同时,在同一个语料内,我们也保持了不同类簇中文档数的跨度(参见表 1 中 Minimum class size 和 Maximum class size 栏),以使语料保持一定的不均衡度,并检验 CADIC 在其上的性能.对这 12 个测试语料集进行分词、去停用词之后,转换成归一化的 VSM 向量矩阵,以供进一步的实验之用.

Table 1 Overview of the datasets (corpus type: R-RCV1, N-20Newsgroup)

表 1 实验语料概况(语料类型:R-RCV1,N-20Newsgroup)

Datasets	R1	R2	R3	R4	R5	R6	N1	N2	N3	N4	N5	N6
#classes	5	7	9	11	13	15	4	6	9	11	13	15
#documents	1 305	1 932	2 482	2 932	3 220	3 330	1 264	2 112	3 168	3 248	3 398	3 406
Average class size	261	276	257.8	266.5	247.7	222	316	352	352	464	377.6	227.1
Minimum class size	97	97	97	97	97	97	208	208	160	160	160	160
Maximum class size	471	499	499	539	499	433	448	448	544	352	332	266

3.2 聚类评价指标

聚类算法的评价指标多种多样,包括准确率、召回率、*F-Measure*、熵等.其中最为常用的是 *F-Measure* 和熵[14].

F-Measure 将准确率和召回率结合在一起.对于语料集中的任何一个类,*F-Measure* 寻找聚类结果中与其最相似的一个簇,根据这个簇计算该类的准确率、召回率及 *F-Measure*.而整个语料集的 *F-Measure* 由所有类的 *F-Measure* 加权累加的方式产生,其中,权值是该类在语料集中的比重.假定 C_i' 是聚类结果中的第 i 个簇,C_i 是答案集合中的第 i 个类,那么,聚类结果的整体 *F-Measure* 为

$$F\text{-}Measure=\frac{\sum_{i=1}^{k}(|C_i|\times F(C_i))}{\sum_{i=1}^{k}|C_i|}=\frac{\sum_{i=1}^{k}\left(|C_i|\times\max_{j=1}^{k}\left(\frac{2\times Precision(C_j',C_i)\times Recall(C_j',C_i)}{Precision(C_j',C_i)+Recall(C_j',C_i)}\right)\right)}{\sum_{i=1}^{k}|C_i|} \tag{10}$$

其中,

$$Precision(C_j',C_i)=\frac{|C_j'\cap C_i|}{|C_j'|},\ Recall(C_j',C_i)=\frac{|C_j'\cap C_i|}{|C_i|}.$$

熵是一种衡量聚类簇纯度的指标,值越小表明聚类结果纯度越大.当所有簇的纯度都最高时,熵值最佳.需要注意的是,当所有簇都仅包含 1 个文档时,熵值亦达到最佳值.整个语料的熵值由聚类结果中所有簇的熵值加权累加产生,权值是该簇在聚类结果中的比重.具体而言,聚类结果的整体熵值定义为

$$Entropy=\frac{\sum_{i=1}^{k}(|C_i'|\times Entropy(C_i'))}{\sum_{i=1}^{k}|C_i'|}=\frac{\sum_{i=1}^{k}\left(|C_i'|\times\left(-\frac{1}{\log k}\sum_{j=1}^{k}\frac{|C_i'\cap C_j|}{|C_i'|}\log\frac{|C_i'\cap C_j|}{|C_i'|}\right)\right)}{\sum_{i=1}^{k}|C_i'|} \tag{11}$$

在后面的实验中,我们用 *F-Measure* 和熵值共同评价聚类结果.

3.3 语料分布差异性验证

我们设计的CADIC聚类算法蕴含着一个潜在前提,即语料中的各个固有簇分布各异(具体表现为各簇在对应 CADIC 坐标轴上的投影分布有较大差异).在本节中,我们设计实验对语料中各个固有簇分布的差异性进行简单验证,以证实 CADIC 中间层关于重标度和语料归一化思想的合理性.

对于一个具有 n 个文档 k 个类别的语料集,我们根据如下步骤简单验证其包含的各个簇的分布差异性:

(1) 计算语料整体的中心 $\boldsymbol{m}=\frac{1}{n}\sum_{\boldsymbol{x}}\boldsymbol{x}$,根据所有文档的类别标签计算各个类的中心 $\boldsymbol{m}_i=\frac{1}{n_i}\sum_{\boldsymbol{x}\in C_i}\boldsymbol{x}$,计算方向 $\boldsymbol{m}_1-\boldsymbol{m},\ldots,\boldsymbol{m}_i-\boldsymbol{m},\ldots,\boldsymbol{m}_k-\boldsymbol{m}(1\leqslant i\leqslant k)$,以构建 CADIC 坐标系.

(2) 对于任意 $C_i(1\leqslant i\leqslant k)$,根据公式(5)计算其包含的所有数据点 $\boldsymbol{x}_j\in C_i(1\leqslant j\leqslant n)$在 $\boldsymbol{m}_i-\boldsymbol{m}$ 坐标轴上的投影.

(3) 将每个类在对应坐标轴方向的投影看作一个分布,对第 2 步计算所得的 k 个分布进行方差齐性检验,以验证它们在散布程度上是否存在明显的不同.

(4) 进一步计算各个分布的标准差,以柱状图的形式进行更为直观的比较.

对于 RCV1 系列的语料集和 20Newsgroup 系列的语料集,我们分别选择类别数最多的 R6 和 N6 进行分布

差异性验证.我们首先调用统计分析软件 SPSS(http://www.spss.com)对语料中各个类在对应坐标轴上的投影进行方差齐性检验,该检验的 H_0 假设为方差齐性,即各个分布的方差不存在显著的不同.在 R6 和 N6 上的实验,均以显著性水平小于 0.001 的检验结果否定了 H_0 假设,因此可以认为实验中各组数据的方差存在显著差异.

更进一步地,我们计算出了各个类在对应坐标轴上投影的标准差,并以柱状图进行比较,结果如图 5 所示.它较为清晰、直观地显示了语料中各个类别分布的差异性,其中,横坐标为各个类别的 ID,纵坐标表示不同类别的数据点在对应 CADIC 坐标轴上投影的标准差.由图 5 可知,R6 中各个类在对应坐标轴投影的标准差从 0.045(Class ID=1)到 0.092(Class ID=11)不等,而 N6 中各个类别的标准差也从 0.038(Class ID=8)到 0.077(Class ID=3)不等.在两个语料集中,最大标准差皆达到了最小标准差的 2 倍.这样的差异表明,语料中各个类簇有着不同的分布,并差异显著,无视这种差异,用统一策略进行聚类是不合理的.因此,CADIC 算法中根据不同类的分布差异设计重标度函数对语料分布进行归一化,以期获得更为合理的聚类判别的思想是合理的.

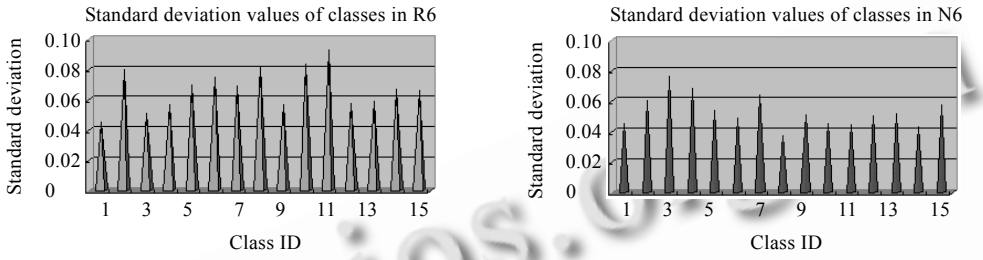


Fig.5 Standard deviation values of classes in R6 and N6

图 5 语料集 R6 和 N6 中各个类别的标准差比较

3.4 CADIC算法性能

在本节中,我们首先设计实验,以验证 CADIC 算法的实际性能.为了较为客观地反映 CADIC 的真实性能,我们选择以下两个算法与之进行性能比较:*K*-means 算法和谱聚类系列中的 Ncut 算法[8].选择 *K*-means 算法作为基准,我们可以比较 CADIC 利用重标度进行语料归一化带来的性能提高;选择 Ncut,我们可以与当前领先水平的聚类算法进行性能比较.

实验中的 *K*-means 和 CADIC 用 C++实现,实验中用到的 Ncut 源码来自美国华盛顿大学(University of Washington)的谱聚类工具箱(http://www.cs.washington.edu/homes/sagarwal/code.html).实验中用到的所有算法都涉及到初始点的选择.为避免初始点的选择对算法性能的影响,对于同一个语料集,每种算法运行 10 次,并且,对其中的每一次运行,*K*-means,Ncut 和 CADIC 都采用同样一组随机生成的初始点.最终,对于每一种算法,我们将 10 次实验的 *F-Measure* 和熵进行平均,以供性能比较.对于 *K*-means,Ncut 和 CADIC 这 3 种算法,也同样涉及到聚类结果中簇数的设定,在实验中,我们统一将聚类结果的簇数设定为语料集中的类别数.在 *K*-means 算法中,我们采用欧氏距离,以同 CADIC 算法中的 CADIC 距离进行比较.在我们的实验中,*K*-means 的收敛条件是,平方和准则函数(所有点到最近簇中心距离的平方和)在两次迭代中的差值不超过 0.001;Ncut 的收敛条件与 *K*-means 一致;CADIC 的收敛条件是,簇与簇之间不存在文档的迁移,或者达到了最大迭代次数(实验中设为 20).

我们在如前所述的两个系列的语料集上比较 *K*-means,Ncut 和 CADIC 的性能,实验结果的 *F-Measure* 和熵值分别见表 2 和表 3.为了便于直观地比较,实验结果同样以折线图的方式在图 6 和图 7 中呈现.图 6 以折线图的方式比较了 *K*-means,Ncut 和 CADIC 算法在 RCV1 系列语料集上的 *F-Measure* 和熵值;图 7 比较了 3 种算法在 20Newsgoup 系列语料集上的 *F-Measure* 和熵值.

Table 2 *F-Measures* of the clustering results for all datasets

表 2 所有语料集上的聚类结果 *F-Measure* 值

Dataset	R1	R2	R3	R4	R5	R6	N1	N2	N3	N4	N5	N6
K-Means	0.597	0.568	0.534	0.529	0.483	0.468	0.717	0.702	0.672	0.647	0.605	0.632
Ncut	**0.714**	**0.698**	0.567	**0.578**	**0.550**	0.521	**0.886**	**0.811**	0.691	0.662	0.652	0.701
CADIC	0.639	0.630	**0.577**	0.562	0.528	**0.535**	0.772	0.792	**0.737**	**0.699**	**0.673**	**0.713**

Table 3 Average entropies of the clustering results for all datasets

表 3 所有语料集上的聚类结果熵值

Dataset	R1	R2	R3	R4	R5	R6	N1	N2	N3	N4	N5	N6
K-Means	0.540	0.545	0.548	0.513	0.538	0.514	0.470	0.424	0.399	0.388	0.408	0.362
Ncut	**0.410**	**0.403**	0.472	**0.446**	0.474	0.488	**0.254**	0.324	0.407	0.393	0.382	0.317
CADIC	0.484	0.468	**0.463**	0.474	**0.470**	**0.476**	0.317	**0.313**	**0.319**	**0.318**	**0.311**	**0.269**

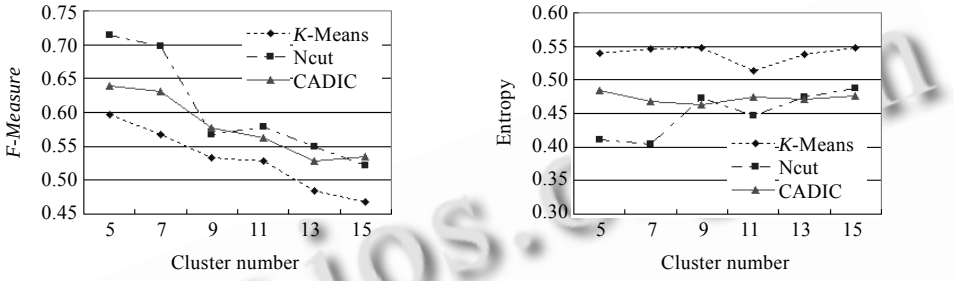


Fig.6 *F-Measure* and entropy scores of different algorithms on datasets of RCV1 series

图 6 不同算法在 RCV1 系列语料集上的 *F-Measure* 和熵值

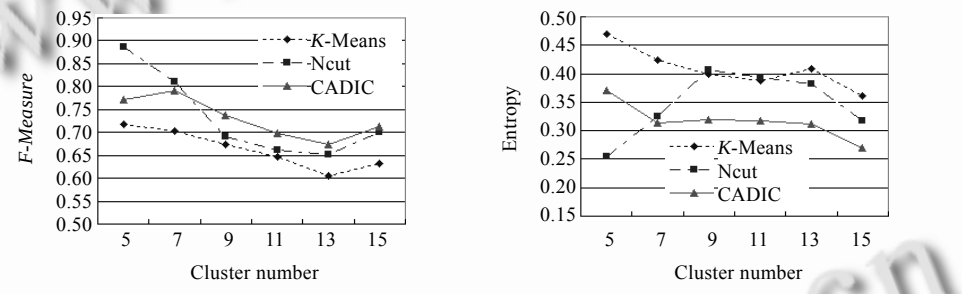


Fig.7 *F-Measure* and entropy scores of different algorithms on datasets of 20Newsgroup series

图 7 不同算法在 20Newsgroup 系列语料集上的 *F-Measure* 和熵值

从实验结果可以看出:

(1) 与 *K*-means 相比,CADIC 在所有语料集上的性能取得了全面而稳定的提升,这证实了 CADIC 算法基于语料特性进行重标度策略的有效性.

(2) 与 Ncut 相比,CADIC 也取得了与之可比的性能.具体分析图 6 和图 7 可知,当语料类别数较小时,Ncut 的性能优于 CADIC(如 R1,R2,N1 等);当语料类别数较大时,CADIC 能够取得与 Ncut 可比甚至略好的性能——在 RCV1 系列的实验中,CADIC 在类别数较大时(R3~R6)取得了与 Ncut 相当的性能;而在 20Newsgroup 系上,CADIC 在除 N1 外的语料集上取得了比 Ncut 略好的性能.

一个值得思考的问题是,为什么语料集的类别个数能够在一定程度上影响实验的比较结果?为什么我们的 CADIC 算法在类别数较小时比 Ncut 略差,而类别数较大时与 Ncut 性能相当甚至略好?考虑 Ncut 聚类过程中构造新特征空间的方法——Ncut 通过特征分解的方式来最小化聚类划分的权重,对应的 k 个最优特征向量也就构成了新的特征空间.因此,当 k 值较小时,特征空间由最优的特征向量构成,于是也会带来最优的聚类结果;当 k

值增大时,随着新加入的特征向量的质量下滑,特征空间的特性也受其影响,聚类结果质量比之 k 值较小时也有所不及.然而,我们的 CADIC 聚类算法由于基于重标度的空间隐式归一化思想的有效性,在不同类别数的所有语料集上都取得了稳定的性能提升(如图 6、图 7 所示).因此,随着 k 值的增加,我们能够取得与 Ncut 相当甚至更好的结果.

在第 2.3 节中,我们证明了 CADIC 算法的时间复杂度保持在与 *K*-means 同一量级.为了进一步验证这一结论,我们实验比较了 CADIC 和 *K*-means 的执行时间.在实验中我们未纳入 Ncut 算法,一方面因为 Ncut 源码来自华盛顿大学的 matlab 谱聚类工具箱,与我们用 C++实现的 CADIC 和 *K*-means 在执行时间上不具可比性;另一方面,Ncut 与 CADIC 和 *K*-means 相比,有较高的时间复杂度量级,我们更关注同一量级的 CADIC 和 *K*-means 在执行时间上的差异.实验在一台联想开天 M4600 计算机上执行,处理器为 Pentium(R) 4 2.93GHz,内存为 1.25GB.在我们用 C++实现的代码中,CADIC 和 *K*-means 采取同样的数据结构,并共用尽可能多的函数以保证实验比较的公平性.CADIC 和 *K*-means 的初始点选择策略、收敛条件如前文所述.对于每一个语料集,同样采取运行 10 次求平均的方式以计算算法执行时间.实验结果见表 4.表中第 2 行和第 3 行分别列出了 *K*-means 和 CADIC 的执行时间.第 4 行结果为 CADIC 执行时间与 *K*-means 执行时间之比,用以验证二者时间复杂度上的常数级差异.

Table 4 Execution time comparison between CADIC and *K*-means

表 4 CADIC 与 *K*-means 执行时间比较

Dataset	R1	R2	R3	R4	R5	R6	N1	N2	N3	N4	N5	N6
K-Means (s)	8.49	25.30	67.78	86.13	121.4	141.6	11.37	52.07	160.8	485.8	289.6	288.2
CADIC (s)	15.68	41.08	92.89	131.7	185.5	206.8	17.63	62.36	179.7	569.6	392.0	433.8
CADIC/*K*-Means	1.85	1.62	1.37	1.53	1.53	1.46	1.55	1.19	1.12	1.17	1.35	1.46

由表 4 可以看出,随着语料中文档数和类别数的增多,*K*-means 和 CADIC 的执行时间都相应地增长,但它们的执行时间之比却一直保持一个比较稳定的数值.在 RCV1 系列语料集上,CADIC 的执行时间大约为 *K*-means 的 1.46~1.85 倍;而在 Newsgroup 系列语料集上,CADIC 的执行时间大约为 *K*-means 的 1.12~1.55 倍.在已有语料集上的实验表明,CADIC 的执行时间通常不会超过 *K*-means 的 2 倍,这证实了第 2.3 节中关于 CADIC 算法的时间复杂度保持在与 *K*-means 同一量级的结论.我们还同时实验比较了 CADIC 和 *K*-means 在迭代次数和每代执行时间上的差异.CADIC 与 *K*-means 迭代次数之比在 RCV1 系列语料集和 Newsgroup 系列语料集上分别在 0.93~1.12 之间和 0.94~1.03 之间.而在每代执行时间的比较上,CADIC 在 RCV1 系列语料集上大约是 *K*-means 的 1.45~1.64 倍,在 Newsgroup 系列语料集上则为 *K*-means 的 1.14~1.57 倍.这两组实验进一步证实了 CADIC 算法的时间复杂度保持在与 *K*-means 同一量级的结论.限于篇幅,这两组实验结果不再以表格的形式给出.

本节分别在算法性能和执行时间两方面对 CADIC 进行了实验验证.总的来说,在时间效率上,CADIC 具有与 *K*-means 同一量级的速度;而在聚类结果的质量上,CADIC 能够取得与当前领先水平的聚类算法 Ncut 相当的实际性能.

4 结论及下一步的研究

大部分聚类算法都固守于自身的潜在假定,而无视潜在假定与语料的真实分布情况之间有多大差异.由此,本文提出了一种基于语料特性的聚类算法 CADIC.CADIC 算法选择一组具有区分度的方向构造 CADIC 坐标系,并在该坐标系下自动分析待聚类语料的分布特性.利用这些分布特性构造重标度函数,以重标度的方式隐式地归一化语料的分布,以使距离度量在新的标度下更为合理.由于重标度的过程实际上是隐式地将分布特性融入聚类判别框架,算法得到的聚类划分自然更为合理.通过恰当的设计,算法的时间复杂度保持在与 *K*-means 同一量级.在 RCV1 语料集及 20Newsgroup 语料集上的实验结果表明,CADIC 根据语料特性进行重标度的思想取得了聚类结果质量的全面提升,并具有与当前领先聚类算法可比的实际性能.

CADIC 聚类算法最重要的贡献是提出了一种基于重标度思想的空间映射,该映射能够将语料归一化到更为理想的形态.显然,这种根据语料特性进行重标度映射的思想同样可以应用到分类领域.因此,在下一步的工

作中,我们一方面进一步提高 CADIC 聚类算法的性能;另一方面将 CADIC 中间层推广到分类领域,以期获得分类性能的提升.

References:

[1] Dumais ST. LSI meets TREC: A status report. In: Harman D, ed. Proc. of the 1st Text Retrieval Conf. (TREC1). National Institute of Standards and Technology Special Publication 500-207, 1993. 137−152.

[2] Kowalski G. Information Retrieval Systems—Theory and Implementation. Boston: Kluwer Academic Publishers, 1997.

[3] Zamir O, Etzioni O, Madani O, Karp RM. Fast and intuitive clustering of Web documents. In: Proc. of the KDD'97. 1997. 287−290.

[4] Allan J, ed. Topic Detection and Tracking: Event-Based Information Organization. Dordrecht: Kluwer Academic Publishers, 2002.

[5] Wu H, Phang TH, Liu B, Li X. A refinement approach to handling model misfit in text categorization. In: Proc. of the SIGKDD 2002. 2002. 207−216.

[6] Tan SB, Cheng XQ, Ghanem MM, Wang B, Xu HB. A novel refinement approach for text categorization. In: Proc. of the 14th ACM CIKM 2005. Bremen: ACM Press, 2005. 469−476.

[7] Shawe-Taylor J, Cristianini N. Kernel Methods for Pattern Analysis. Cambridge: Cambridge University Press, 2004.

[8] Ng A, Jordan M, Weiss Y. On spectral clustering: Analysis and an algorithm. In: Dietterich T, Becker S, Ghahramani Z, eds. Advances in Neural Information Processing Systems 14. Cambridge: MIT Press, 2002.

[9] Chan PK, Schlag DF, Zien JY. Spectral *K*-way ratio-cut partitioning and clustering. IEEE Trans. Computer-Aided Design, 1994, 13(9):1088−1096. [doi: 10.1109/43.310898]

[10] Ding C, He X, Zha H, Gu M, Simon HD. A min-max cut algorithm for graph partitioning and data clustering. In: Proc. of the 1st Int'l Conf. on Data Mining (ICDM). 2001. 107−114.

[11] Han J, Kamber M. Data Mining: Concepts and Techniques. 2nd ed., San Francisco: Morgan Kaufmann Publishers, 2006.

[12] Duda RO, Hart PE, Stork DG. Pattern Classification. 2nd ed., New York: Wiley-Interscience Publishers, 2000.

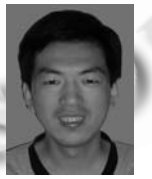
[13] Lewis DD, Yang Y, Rose T, Li F. RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research, 2004,5:361−397.

[14] Zhou ZT. Quality evaluation of text clustering results and investigation on text representation [MS. Thesis]. Beijing: Institute of Computing Technology, the Chinese Academy of Sciences, 2005 (in Chinese with English abstract).

附中文参考文献:

[14] 周昭涛.文本聚类分析效果评价及文本表示研究[硕士学位论文].北京:中国科学院计算技术研究所,2005.

曾依灵(1980－),男,重庆人,博士生,主要研究领域为大规模文本处理,文本表示,文本聚类.



吴高巍(1975－),男,博士,助理研究员,主要研究领域为机器学习,数据挖掘.

许洪波(1975－),男,博士,副研究员,主要研究领域为大规模文本处理,互联网搜索,文本过滤.

白硕(1956－),男,博士,研究员,博士生导师,主要研究领域为计算语言学,数据挖掘,网络安全.