基于 Kolmogorov 复杂性的文本聚类算法改进

王有华 陈笑蓉
(贵州大学计算机科学与技术学院 贵阳 550025)

摘 要 基于Kolmogorov复杂性的聚类算法虽然具有普适性、参数无关性的优点,但是应用到文本内容语义信息聚类时往往准确率较低。针对这一问题,提出了一种基于特征扩展的文本聚类改进算法——DEF-KC算法。该算法通过引用百度百科中特定词条的信息,对预处理过的文本中的关键词进行特征扩展,从而提高特征词的主题贡献度,增强文本的结构辨识度,并通过选取特定压缩算法近似计算Kolmogorov复杂性得到文本相似度,最后使用谱聚类算法进行聚类。实验结果表明,与传统的基于Kolmogorov复杂性的文本聚类算法相比,使用该算法时聚类准确率和召回率均得到了较大提升。

关键词 Kolmogorov复杂性,文本聚类,特征扩展,谱聚类
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.5.045

Improved Text Clustering Algorithm Based on Kolmogorov Complexity

WANG You-hua CHEN Xiao-rong
(College of Computer Science and Technology, Guizhou University, Guiyang 550025, China)

Abstract Clustering algorithm based on Kolmogorov complexity has the advantages of generality, parameter independence, but always shows low accuracy when applied to the text semantic information clustering. In order to solve this problem, this paper proposed a text clustering algorithm based on feature extension—DEF-KC. For improving keyword's theme contribution, DEF-KC applies feature extension to the keyword in the pretreated text by referencing information of specific entry in a baidu encyclopedia, and calculates the text similarity by approximate Kolmogorov complexity of the text. Finally it clusters text using spectral clustering algorithm. The experimental results show that the proposed algorithm has much better accuracy and recall rate compared to the traditional text clustering algorithm based on Kolmogorov complexity.

Keywords Kolmogorov complexity, Text clustering, Feature extension, Spectral clustering

1 引言

随着互联网信息的快速增长,如何对海量文本信息进行有效聚类一直是文本挖掘领域的研究热点。典型的文本聚类过程分为文本表示、文本间相似性计算、文本聚类和聚类结果评价4个阶段。传统的文本聚类算法通常采用向量空间模型(VSM)[1]进行文本表示,但是文本向量空间存在高维稀疏的问题,而且随着文本数据规模的增大,向量空间维度变大,导致文本相似度计算复杂,聚类准确度下降。

基于Kolmogorov复杂性的通用相似度度量方法[2]的提出,为文本聚类算法的研究提供了新的思路。通过压缩算法模型近似计算文本对象之间的Kolmogorov复杂性,进而获取文本间的相似度,省去了传统文本聚类方法中文本表示、特征提取等复杂计算过程,具有普适性、参数无关性等优点。基于Kolmogorov复杂性的文本聚类方法已成功应用于代码抄袭检测[3]、网络异常检测[4]、垃圾邮件过滤[5]等应用领域。但是上述应用领域均针对文本特定结构进行聚类(例如代码抄袭检测根据代码共现度进行聚类,网络异常检测根据网络流量结构变化进行检测等),聚类结果解释也依赖于文本具体应用场景,在针对文本内容语义信息进行抽象聚类方面(例如根据文本的表示内容,按照不同主题即军事、文化、教育等进行聚类),聚类准确率较低[6]。针对以上问题,本文提出了一种基于特征扩展的文本聚类算法(Document Feature Expansion based Kolmogorov Clustering, DFE-KC),同时设计了实验,将DEF-KC算法与几种传统文本聚类方法进行对比,结果表明DEF-KC相比于传统的基于Kolmogorov复杂性的聚类方法在准确率和召回率方面有较大提升,且更加适合大规模文本聚类。

本文第2节主要介绍Kolmogorov复杂性的基本概念;第3节具体介绍DEF-KC算法;第4节进行实验并分析实验结果;最后对全文进行总结。

到稿日期:2015-07-02 返修日期:2015-09-20 本文受国家自然科学基金(61363028)资助。
王有华(1990—),男,硕士生,主要研究方向为自然语言处理、数据挖掘,E-mail:wannianma@foxmail.com;陈笑蓉(1954—),女,教授,主要研究方向为信息检索与信息挖掘、自然语言处理等,E-mail:xrchengz@163.com。

2 Kolmogorov 复杂性

2.1 基本概念

Kolmogorov 复杂性[7] (Kolmogorov Complexity, KC, K 复杂性)由 Andrey N. Kolmogorov 于 1963 年首次提出，它的提出主要是为了更加准确地度量单个字符串或对象中包含的随机性或信息量。随着研究的深入，K 复杂性已成为信息论、计算理论和人工智能等诸多领域的重要理论基础。

定义字符串 s 的 K 复杂性为 $K(s)$，它表示在图灵机 T 上输出串 s 并停机时的最短程序 p 的长度，其数学描述如下：

$$K(s)=\min_{p:T(p)=s} L(p) \tag{1}$$

其中，L 表示程序 p 的长度函数，$T(p)$ 表示在图灵机 T 上运行程序 p 得到字符串 s。例如，$s=00\cdots0$（n 个 0），可以使用如下程序对字符串 s 进行描述：for $t=1:n$ print '0'。但是若 s 为采用抛硬币形式得到的 n 位的字符串，那么我们很难找到一个程序对 s 进行直观描述。

实际上，K 复杂性是一个建立在理论基础之上的理想值，是不可计算的，通常采用数据压缩算法来近似得到，而 KC 是任意压缩值的下界。数据压缩算法包括有损和无损压缩两种，在近似计算 KC 值时主要使用无损压缩算法。

2.2 基于 K 复杂性的距离度量准则

受到 K 复杂性思想的启发，Li 等人提出了基于 K 复杂性的对象相似度度量理论，称为正规化信息距离[8] (Normalized Information Distance, NID)，其提供了判断对象之间相似性的通用计算方法。不同于以往基于领域知识进行特征提取的对象相似度计算方法，它具有普适性、领域无关性和参数无关性的优点。

字符串 a 与 b 之间的正规化信息距离定义如下：

$$NID(a,b)=\frac{\max\{K(a|b),K(b|a)\}}{\max\{K(a),K(b)\}}=\frac{K(ab)-\min\{K(a),K(b)\}}{\max\{K(a),K(b)\}} \tag{2}$$

其中，$K(a|b)$ 表示已知 b 的情况下 a 的 KC 值，更直观地表示，已知 b 计算出 a 的最短计算机程序的长度；$K(b|a)$ 表示已知 a 的情况下 b 的 KC 值；$K(a)$、$K(b)$ 表示字符串 a、b 对应的 KC 值。

由于 K 复杂性是不可计算的，因此一般选择无损压缩算法 C 来近似计算 KC 值，则有：$K(a)\approx C(a)$，相应的字符串 a 与 b 之间的正规化压缩距离定义如下：

$$d(a,b)=\frac{C(ab)-\min\{C(a),C(b)\}}{\max\{C(a),C(b)\}}\approx NID(a,b) \tag{3}$$

其中，$C(ab)$ 表示串 a 与 b 连接之后的压缩值，$C(a)$ 表示字符串 a 的压缩值，$C(b)$ 表示字符串 b 的压缩值，$d(a,b)$ 的取值范围为 $0\leqslant d\leqslant 1+r$。

目前，基于 Kolmogorov 复杂性的距离度量准则已成功应用到不同领域，如归纳推理、机器学习、模式识别、基因序列分析等。

3 DEF-KC 算法

与传统文本聚类算法相比，DEF-KC 算法省去了文本表示、特征提取过程，增加了文本特征扩展步骤，且文本相似度采用正规化压缩距离进行计算，聚类处理采用谱聚类算法实现。DEF-KC 算法的具体执行过程如图 1 所示。

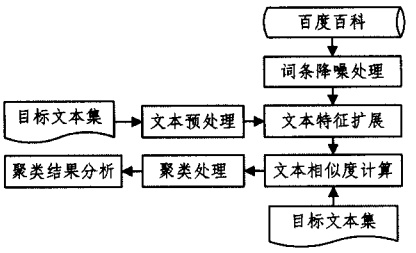

图 1 基于文本特征扩展的文本聚类过程

3.1 文本集的预处理

由于正规化压缩距离具有普适性的优点，现有基于 K 复杂性的聚类算法一般不会对文本进行预处理，但是在针对文本内容信息聚类方面，文献[9]已证明适当的文本预处理工作可以有效地减少对文本主题贡献度低的冗余信息，从而提高聚类准确度。

本文选取中文文本数据集进行聚类实验，所以文本预处理过程主要包括 3 个部分：首先，对文本进行分词，本文选用中科院的“ICTCLAS”分词系统进行分词；其次，进行停用词过滤操作，根据停用词表中列出的诸如“的”、“了”、“并且”、“但是”等在文本中频繁出现但又无法表征文本特征的词汇进行过滤；最后，在上述两步工作的基础之上，选择名词、形容词、动词 3 类更能表达原文语义的词汇来表示文本。

3.2 文本特征扩展

在文本预处理之后，需要对文本内容进行特征扩展，从而丰富文本关键词汇的表达能力。通过百度百科提供的在线 API 对关键词进行查询，并对返回的 HTML 内容进行解析，获取当前词条的名片内容。由于某些关键词条可能在文本或文本集中多次出现，因此为了提高特征扩展效率，避免每次查询百度 API 的时间开销，需要将已查询过的词条存储到数据库中的用于后续对该词条的访问。

当输入的关键词是多义词时，百度百科中可能会出现多个针对该词条的解释，比如当输入的关键词为“小米”时，百度百科中针对“小米”有 9 个义项，分别为小米公司、禾本科谷类植物、某电视剧主人公等。为了选择正确的词条解释语句，需要对词条扩展内容进行降噪处理。对于特定词条的多个解释语句，首先对解释语句进行预处理，然后将处理结果与特征词所在文本进行匹配，最后根据式(4)计算匹配度。当匹配度超过某一阈值时，则认为该解释语句与当前特征词条相似度高，如果存在多个符合条件的解释语句，则选取匹配度最高的解释语句用于特征扩展。匹配度的数学定义如下所示：

$$P(w_i,d_j)=\frac{A_{ij}}{B_i\times\beta}+\frac{A_{ij}}{(B_i\times\beta+C_j-A_{ij})} \tag{4}$$

其中，w_i 表示某文本中需要扩展的第 i 个特征词；d_j 表示该特征词条对应的第 j 个解释语句；A_{ij} 表示当前特征词对应文本与第 j 个解释语句的完全匹配的非重复词汇数目；B_i 表示文本非重复词汇数目；C_j 为第 j 个解释语句的非重复词汇数目；β 表示文本长度修正参数。

基于百度百科外部知识的文本特征扩展算法的具体过程如下：

(1)对于输入的查询关键字 Key，检查 Key 是否已存在于〈Key, Value〉键值对中，如果存在，则返回对应 Value 值，执行步骤(4)。

(2)通过百度百科在线 API 对关键字 Key 进行查询，解析返回结果并进行预处理，返回待扩展内容；如果关键字对应

多条解释语句，则以数组形式返回待扩展内容。

(3)执行数据库插入命令，将待扩展内容转为 JSON 格式，以〈Key，Value〉形式存入数据库。

(4)对关键字 Key 的多条解释语句进行降噪处理，选择匹配度最高的进行扩展。

(5)算法结束，返回扩展结果。

得到关键字对应的特征扩展结果之后，需要将特征扩展结果插入到对应文本中。本文只选取对文本语义贡献度较大且查询百度百科失败率最低的名词进行特征扩展操作。

3.3 文本相似度计算

将文本内容进行特征扩展之后，就需要通过近似计算 KC 值来得到文本间的相似度，建立文本相似度矩阵。对于给定的文本 x 和 y，其相似度 $s(x,y)$ 的数学定义如下：

$$s(x,y)=1-d(x,y)$$
$$=1-\frac{C(xy)-\min\{C(x),C(y)\}}{\max\{C(a),C(y)\}} \quad (5)$$

其中，$d(x,y)$ 为文本 x 到 y 之间的正规化压缩距离，相似度 $s(x,y)$ 的取值范围为 $0\leqslant s\leqslant 1$。

文本的 K 复杂性需要通过压缩算法来近似计算，而压缩效果的好坏直接影响文本相似度的计算结果。为了选取文本压缩方面表现最好的压缩算法，设计了以下实验，通过对随机选取的 6 类文本中的 1000 篇文本进行压缩实验，对比以下 5 种常用压缩算法的平均压缩比，即 Huffman、LZW、Gzip、Zlib 和改进的无缓冲限制的 Lemple-Ziv(LZ)压缩算法。其中本文使用的无缓冲限制的 LZ 压缩算法是对传统 LZ77 压缩算法在提高压缩比方面的一种改进。传统 LZ77 算法为了保证算法的执行效率，会对已编码字典的缓冲窗口作出一定限制，而本文为了追求最好的压缩效果，选择在算法效率上作出一定牺牲，减少对已编码字典大小的限制，增大匹配串长度，从而一定程度上提高压缩算法的压缩比。压缩比率对比结果如表 1 所列。

表 1 压缩算法的文本压缩比率对比

文本类别	Huffman	LZW	Gzip	Zlib	LZ Without buffer bound
旅游	1.237	1.978	1.492	1.489	2.270
IT	1.023	1.889	1.432	1.455	2.120
财经	1.268	2.111	1.598	1.624	2.357
体育	1.078	1.927	1.511	1.499	2.247
教育	1.189	2.086	1.594	1.602	2.363
军事	1.204	2.128	1.597	1.584	2.365

通过观察表 1 可以发现，相对于其他压缩算法，改进的无缓冲限制的 LZ 压缩算法在针对文本压缩方面具有更高的压缩比，所以本文选取无缓冲限制的 LZ 压缩算法来近似计算 K 复杂性。

3.4 文本聚类处理

聚类处理是文本聚类算法的核心。对于给定的文本数据集合 $D=\{d_1,d_2,\cdots,d_i,\cdots,d_n\}$，聚类处理将每对文本 (d_i,d_j) 之间的文本相似度 $sim(d_i,d_j)$ 作为输入，选用特定聚类算法进行分析处理，最后将聚类结果输出。

作为聚类分析中一个崭新的分支，谱聚类算法[10]将聚类问题转化为图的最优划分问题来解决，可以聚类任意形状的样本数据且能够收敛于全局最优解，所以本文选取谱聚类算法来实现文本聚类分析。经典 NJW 谱聚类算法的实现过程如下所示。

输入：文本相似度矩阵和聚类数目 K

输出：文本聚类结果

Step1 构造无向加权图的邻接矩阵 W，计算得到 Laplacian 矩阵 L；

Step2 计算矩阵 L 的前 k 个特征值和特征向量，构造特征向量空间；

Step3 使用 K-means 算法对特征向量空间聚类；

Step4 根据特征向量聚类结果，返回文本集聚类结果。

4 实验及结果分析

4.1 实验数据集

实验选取搜狗中文实验室提供的文本分类语料库 Sogou-C，该语料库来源于 Sohu 新闻网保存的大量手工整理和分类的新闻语料，数据分为 10 个大类，如表 1 所列。

表 2 实验数据集分布统计详情

类别名称	类别代码	文本平均字数	文档数量
汽车	C07	1161	8000
财经	C08	1205	8000
IT	C10	927	8000
健康	C13	1121	8000
体育	C14	720	8000
旅游	C16	956	8000
教育	C20	1523	8000
招聘	C22	1708	8000
文化	C23	3486	8000
军事	C24	1205	8000

文章篇幅的长短对文本压缩比有较大影响。为了保证实验的稳定性，本课题选择文本数据集中平均字数差别较小的 8 个类别(除文化和招聘类)，每个类别随机选择 500 篇文本，即共选择 4000 篇文本进行实验。

4.2 评价标准

由于本课题是对已知类别的文本内容进行聚类，实验采用准确率(Precision)和召回率(Recall)两个外部评价指标对实验结果进行测评，其中准确率考察聚类的精确度，而召回率考察聚类的完整性。准确率和召回率的数学定义如下：

$$Precision(i,j)=\frac{n(i,j)}{n_j} \quad (6)$$

$$Recall(i,j)=\frac{n(i,j)}{n_i} \quad (7)$$

其中，$n(i,j)$ 表示在聚类结果 j 中包含预定义类别 i 的文本个数；n_j 表示聚类结果 j 中文本的个数；n_i 表示预定义类别 i 中文本的个数。

算法整体的准确率和召回率定义为各个类别的准确率和召回率的加权平均值，数学定义如下：

$$P=\sum_{i=1}^{K}\frac{n_i}{N}\times Precision(i,j) \quad (8)$$

$$R=\sum_{i=1}^{K}\frac{n_i}{N}\times Recall(i,j) \quad (9)$$

其中，K 表示文本聚类总个数；N 表示实验数据集中文本总个数；n_i 表示预定义类别 i 中文本的个数。

4.3 实验结果分析

为了验证本文提出的 DEF-KC 算法的有效性，本文设计了两组对比实验。第一组实验中将 DFE-KC 算法同传统的基于 K 复杂性的聚类算法、基于 VSM 的文本聚类算法进行比较；第二组实验中通过改变测试文本的数量，观察 DEF-KC 算法和 VSM 聚类算法的准确率和召回率随着文本数量增大的变化情况，测试两种算法的稳定性。为了保证实验的公平性，以上文本聚类算法的聚类处理过程统一使用谱聚类算法。考虑到谱聚类算法的不稳定性，选取 6 次运行结果的平均值

作为最终的实验结果。

表3是第一组实验中3种文本聚类算法的聚类结果对比，其中每个类别的测试文本数目为500。

表3　3种算法的准确率和召回率比较(%)

文本类别	KC聚类算法		VSM聚类算法		DEF-KC聚类算法	
	准确率	召回率	准确率	召回率	准确率	召回率
汽车	56.1	39.4	84.3	72.8	82.3	67.1
财经	38.0	26.7	68.4	61.9	66.5	54.9
IT	49.7	47.5	74.5	80.3	75.7	74.1
健康	54.1	35.2	78.9	69.7	80.2	62.8
体育	40.5	40.8	70.3	74.5	66.5	68.1
旅游	33.4	32.9	66.4	68.4	62.6	62.0
教育	38.5	51.0	64.2	86.1	68.4	79.4
军事	45.9	31.4	77.1	65.8	73.2	59.6
平均值	44.5	38.1	72.9	72.4	71.8	66.0

从表3中可以看出，相对于传统KC聚类算法，DEF-KC聚类算法在准确率和召回率两个方面均有25%～30%的提升，说明DEF-KC算法相对于传统KC聚类算法在文本内容信息聚类方面改进明显；而相比于传统基于VSM的聚类算法，DEF-KC算法在准确率方面达到了与之相近的效果，而召回率方面降低了5%～7%，这可能是因为DEF-KC虽然通过特征扩展方法提升了文本的类别区分度，但是相比于基于VSM的文本表示方法，其在不同类别的文本之间的区分能力还是较弱。DEF-KC算法虽然在文本特征扩展阶段进行了降噪处理，但仍然会引入小部分对文本类别信息无贡献甚至起反作用的噪音词语，这需要在特征扩展阶段进行更加细致的处理(如文献[11]中采取的方法)。但相比于传统VSM聚类算法，DEF-KC算法不存在特征选择、高维矩阵运算等复杂过程，所以DEF-KC算法在算法性能方面优势明显。

图2、图3是第二组实验的结果，两图分别表示两种文本聚类算法的平均准确率和召回率与测试文本数目的关系，其中测试文本总数依次选取为800、1600、2400、4000、5600、7200。

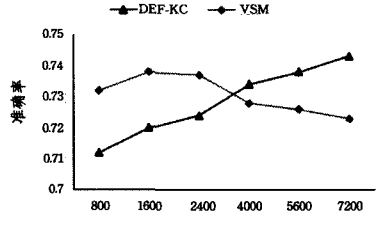

图2　两种算法的准确率与文本数目的关系

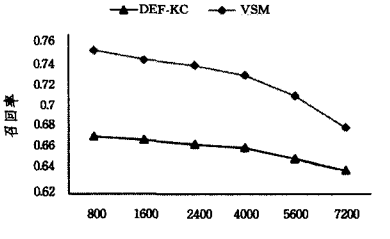

图3　两种算法的召回率与文本数目的关系

通过图2可以明显看出，传统基于VSM的文本聚类算法在测试文本数目超过2400时，准确率开始下降；而随着文本数目的增加，DEF-KC算法在准确率方面保持上升趋势。图3显示随着测试文本数目的增加，两种聚类算法的召回率均有所下降，但DEF-KC算法下降趋势缓慢，算法稳定性更好。结合图2、图3的结果可以看出，相比于传统VSM算法，DEF-KC算法更加适合大规模文本聚类。

结束语　本文提出了一种基于特征扩展的文本聚类算法，通过引入百度百科作为外部知识对预处理过的文本关键词进行特征扩展，使用正规化压缩距离计算文本间相似度，最后选取谱聚类算法对文本进行聚类处理。实验结果表明，DEF-KC算法相比于传统基于K复杂性的聚类方法在准确率和召回率方面有较大提升，而且在大规模文本聚类方面，比传统基于VSM的文本聚类算法表现更加稳定。但是，DEF-KC算法在准确率和召回率的表现上仍与现有常用的文本聚类算法有一定差距，本文后续将进一步研究提升DEF-KC算法的聚类的准确率和召回率，考虑细化特征扩展对象，引入百度百科语义结构信息；优化正规化压缩度量方法，提升文本间相似度的准确性。

参考文献

[1] Ming Jun-ren. Research on Text Clustering Model Based on Ontology Graph[J]. Information Science, 2013, 31(2): 29-33(in Chinese)
明均仁. 基于本体图的文本聚类模型研究[J]. 情报科学, 2013, 31(2): 29-33

[2] Nikvand N, Wang Z. Generic image similarity based on Kolmogorov complexity[C]// 2010 17th IEEE International Conference on Image Processing(ICIP). IEEE, 2010: 309-312

[3] Zhang L, Zhuang Y, Yuan Z. A program plagiarism detection model based on information distance and clustering[C]// The 2007 International Conference on Intelligent Pervasive Computing, 2007(IPC). IEEE, 2007: 431-436

[4] Ukil A. Application of Kolmogorov complexity in anomaly detection[C]// 2010 16th Asia-Pacific Conference on Communications(APCC). IEEE, 2010: 141-146

[5] Belabbes S, Richard G. On Using SVM and Kolmogorov Complexity for Spam Filtering[C]// FLAIRS Conference. 2008: 130-135

[6] Geweniger T, Schleif F M, Hasenfuss A, et al. Comparison of cluster algorithms for the analysis of text data using kolmogorov complexity[M]// Advances in Neuro-Information Processing. Springer Berlin Heidelberg, 2009: 61-69

[7] Vita, nyi, Paul M B. Information Distance in Multiples[J]. IEEE Transactions on Information Theory, 2011, 57(4): 2451-2456

[8] Vitanyi P M B, Balbach F J, Cilibrasi R L, et al. Normalized information distance[M]// Information Theory and Statistical Learning. Springer US, 2009: 45-82

[9] Tao Xiao-lei. Study of Kolmogorov Complexity Based Clustering Algorithms[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2013(in Chinese)
陶小雷. 基于Kolmogorov复杂性的聚类方法研究[D]. 南京：南京航空航天大学, 2013

[10] Wang Hui-qing, Chen Jun-jie. Research of spectral clustering based on graph partition[J]. Computer Engineering and Design, 2011(1): 289-292(in Chinese)
王会青, 陈俊杰. 基于图划分的谱聚类方法的研究[J]. 计算机工程与设计, 2011(1): 289-292

[11] Lv Chao-zhen, Ji Dong-hong, Wu Fei-fei. Short text classification based on expanding feature of LDA[J]. Computer Engineering and Applications, 2015(4): 123-127(in Chinese)
吕超镇, 姬东鸿, 吴飞飞. 基于LDA特征扩展的短文本分类[J]. 计算机工程与应用, 2015(4): 123-127