计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2018.20160662
55(1):102-112, 2018

一种基于频繁词集表示的新文本聚类方法

张雪松　　贾彩燕
(交通数据分析与数据挖掘北京市重点实验室(北京交通大学)　北京　100044)
(北京交通大学计算机与信息技术学院　北京　100044)
(15120467@bjtu.edu.cn)

A New Documents Clustering Method Based on Frequent Itemsets

Zhang Xuesong and Jia Caiyan
(*Beijing Key Lab of Traffic Data Analysis and Mining* (*Beijing Jiaotong University*), *Beijing* 100044)
(*School of Computer and Information Technology*, *Beijing Jiaotong University*, *Beijing* 100044)

Abstract　Traditional document clustering methods use vector space model (VSM) of words to represent documents. This VSM representation only measures the importance of a single words, while ignores the semantic relationship between words, and has high dimensionality. In this study, we propose a new document clustering method: FIC (frequent itemsets based document clustering method). In the method, we use frequent itemsets (where a frequent itemset is a set of frequently co-occurred words) mined by FP-Growth algorithm in documents to represent each document. We then construct the document-document relationship network based on the similarity between pairs of documents at this new representation. At last, we divide the network into communities using a given community detection method to complete document clustering. Thereby, FIC can not only overcome the high dimensionality of VSM, but also fully make use of topological relationship among documents. The experimental results on two English corpora (Reters-21578 and 20Newsgroup) and one Chinese corpus (Sougou-News) demonstrate that the proposed method FIC is superior to the other existing frequent itemsets based methods and other classical state-of-the-art document clustering methods, and the top K words for characterizing each topic of documents identified by FIC are more meaningful than the classical topic model LDA (latent Dirichlet allocation).

Key words　document clustering; frequent itemsets; complex network; community division; text representation model

摘　要　传统的文本聚类方法大部分采用基于词的文本表示模型,这种模型只考虑单个词的重要度而忽略了词与词之间的语义关系.同时,传统文本表示模型存在高维的问题.为解决以上问题,提出一种基于频繁词集的文本聚类方法(frequent itemsets based document clustering method, FIC).该方法从文档集中运用FP-Growth算法挖掘出频繁词集,运用频繁词集来表示每个文本从而大大降低了文本维度,根据文本间相似度建立文本网络,运用社区划分的算法对网络进行划分,从而达到文本聚类的目的.FIC算法不仅能降低文本表示的维度,还可以构建文本集中文本间的关联关系,使文本与文本间不再是

收稿日期:2016-08-30;**修回日期**:2017-07-04
基金项目:国家自然科学基金面上项目(61473030);数字出版国家重点实验室专项课题
This work was supported by the General Program of the National Natural Science Foundation of China (61473030) and the Opening Project of State Key Laboratory of Digital Publishing Technology.

独立的两两关系.实验中运用2个英文语料库Reuters-21578,20NewsGroup和1个中文语料库——搜狗新闻数据集来测试算法精度.实验表明:较传统的利用文本空间向量模型的聚类方法,该方法能够有效地降低文本表示的维度,并且,相比于常见的基于频繁词集的聚类方法能获得更好的聚类效果.

关键词 文本聚类;频繁词集;复杂网络;社区划分;文本表示模型

中图法分类号 TP301

文本聚类是文本挖掘中的一个重要课题,是指在没有人工干预的情况下,将大量无标注的数据集根据文本间相似性划分成若干个类,使得同类内的文本相似度较大而类间的文本相似度较低.在文本聚类中要解决三大问题:1)数据的高维问题,解决此问题需要处理稀疏的数据空间或引用数据降维的方法;2)聚类算法在大规模文本集上聚类效果的问题,即聚类算法要有很好的精度和可扩展性;3)对类簇的主题描述,一个好的主题描述方法能够让人们直观地了解每个类簇所代表的主题.在文本聚类中,向量空间模型(vector space model, VSM)[1]是常用的模型,在VSM模型中,每个文本集中的词作为空间坐标系中的一个维度,每个文本是特征空间中的1个向量.尽管基于VSM的K-means聚类方法在文本聚类中被广泛应用,但这种文本表示的方法存在高维问题,向量空间中的特征维度过高导致文本向量过于稀疏,影响聚类的效果,并且VSM模型使用单个词汇作为特征,忽略了词之间的语义关系.

基于频繁词集的文本聚类方法,可以解决传统文本表示的高维和语义缺失问题.频繁词集是指同时出现在一定比例的文本中的词的集合,这些集合在某种程度上能够反映出一些关于主题的语义.基于频繁词集的文本聚类首先挖掘出文本集中的频繁词集,将文本指派到与其具有相同主题的频繁词集中[2].Beil等人[3]提出FTC(frequent term-based clustering)算法和HFTC(hierarchical frequent term-based clustering)算法,该算法将挖掘出的频繁词集当作聚类初始簇,采用贪婪策略优先选择簇重叠度最小的频繁词集进行聚类,该算法采用一种局部最优策略,对项集的选择顺序过于依赖,算法效率较低.Fung等人[4]提出一种层次聚类方法,该算法同样将挖掘出的频繁词集当作初始簇,为解决类重叠问题,将每个文本通过打分函数划分到最适合的簇中,然后构建树,运用剪枝策略进行聚类.Zhang等人[5]提出的MC(maximum capturing)算法将挖掘出的频繁词集当作特征来表示文本,通过计算文本间共有的词集个数来度量文本间相似性并直接对文本进行聚类.为了充分挖掘语义关系,Hernández-Reyes等人[6]考虑了文本中词集的语序,提出了基于最大频繁序列(maximal frequent sequence, MFS)的文本聚类方法.

除了基于频繁词集的文本聚类方法,基于LDA(latent Dirichlet allocation)的文本聚类算法[7]、SPK-means(spherical-K-means)算法[8]和GNMF(graph regularized nonnegative matrix factorization)算法[9]也是较流行的文本聚类方法.其中,基于LDA的文本聚类方法基于这样一个假设:一个文本中会包含多个主题,每个主题生成各种词语的概率服从多项分布,通过该聚类方法会得到一个文本在不同主题下的分布情况,即文本在各主题下的概率,最后将文本指派到隶属度最高的主题中.并且,通过统计出各个主题下词的概率分布,那些在该主题上概率高的词能够非常好地描述该主题的语义.GNMF是一种基于非负矩阵分解(nonnegative matrix factorization, NMF)[10]的方法,该方法基于这样的假设:如果2个数据点在高维空间具有相近的几何分布,则在降维后的空间中2个数据点应该也相近.而NMF方法是近年来兴起的无监督聚类方法,其本身也是一种很有效的数据降维的方法,因此GNMF在文本聚类中可以解决文本维度过高的问题.SPK-means算法运用余弦相似度作为衡量文本间相似度的指标,将文本与初始中心间的相似度总和作为待优化的代价函数,在迭代过程中最大化全局代价函数并更新聚类中心.SPK-means运行速度快、内存消耗较小,因为采用余弦相似度,能够有效地处理大文本数据集[11].

本文引用频繁词集理论,提出了一种基于频繁词集的文本聚类方法(frequent itemsets based document clustering method, FIC).该方法首先挖掘出频繁词集,运用频繁词集来构建文本表示模型.将每个文本看作网络中的节点,通过度量2个文本间的相似性来构建边,从而构建成一个文本网络,

运用社区划分的方法来实现文本的聚类.该方法不仅有效地降低了文本表示的维度,而且文本社区的引入建立了文本之间的关联关系,使得聚类效果得到了有效提升.

1 相关算法介绍

基于频繁词集的文本聚类方法是解决数据高维问题的一个重要方法,对于聚类精度也比传统模型有了进一步提高,同时能够更加具体地描述数据集中的每个主题.

1.1 FTC 算法

FTC 算法首先从文本集中挖掘出所有满足最小支持度的频繁词集,将每个频繁词集当作聚类的候选簇,然后通过一种贪心策略,从中选择与其他候选簇重叠度最小的作为结果簇,直到文本全部被指派到相应簇中为止.

由于一个文本通常会包含多个频繁词集,因此候选簇之间会出现重叠的现象,即一个文本可能会属于不同的候选簇,为了消除类间重叠,定义熵重叠度(entropy overlap)$EO(C_i)$来衡量候选簇 C_i 与其他候选簇的重叠情况:

$$EO(C_i)=\sum_{D_j\in C_i}-\frac{1}{f_j}\ln\left(\frac{1}{f_j}\right), \tag{1}$$

其中,D_j 表示文本,f_j 表示文本 D_j 所支持的频繁词集的个数.式(1)反映了 C_i 所支持的频繁词集在其他候选簇中的分布情况,值越大,C_i 与其他簇的重叠度越高.

算法 1. FTC 算法.

1) 从文本集中挖掘满足最小支持度($minsup$)的频繁词集 $F=\{F_1,F_2,\cdots,F_n\}$,每个频繁词集 F_i 对应的文本集合构成候选簇;

2) 计算每个候选簇 C_i 的熵重叠度;

3) 把熵重叠度最小的 C_i 加入结果簇集 C 中;

4) 若文本 D_j 包含在 C_i 中,则将其在其他候选簇中删除;

5) 删除候选簇 C_i;

6)判断文本是否全部包含在结果簇中,如果没有则返回 3)续执行,否则结束.

FTC 算法虽然在降低文本维度和效率上较传统基于 VSM 模型的聚类方法提高了效率,但使用贪心策略选择结果簇无法达到全局最优,且对候选簇的选择顺序有依赖,在选择结果簇中会消耗大量时间.

1.2 FIHC 算法

FIHC 算法由 Fung 等人[4]提出.步骤分为频繁词集挖掘、构建初始簇、建树、剪枝,最后得到树形结构的结果簇.此方法基于层次聚类,将文本集中挖掘出的频繁词集当作初始簇,在解决类重叠问题中,采用一种打分函数将文本归到最适合的类簇中,将初始簇构建成树结构,运用剪枝策略来实现聚类.其中的文本归档打分函数为

$$Score(C_i\leftarrow D_j)=\left[\sum_x n(x)\times cluster\_support(x)\right]-\left[\sum_{x'} n(x')\times global\_support(x')\right], \tag{2}$$

其中,$Score(C_i\leftarrow D_j)$表示文本 D_j 属于类簇 C_i 的分数;$global\_support$ 代表全局频繁词集集合,指的是在所有文本集中满足最小支持度的所有频繁词集;$cluster\_support$ 表示簇频繁词集集合,它指的是将文本指派到其含有的初始簇之后,根据每个簇内的文本中频繁词集的出现频率筛选出大于某一最小支持度的频繁词集,并将其作为簇频繁词集,簇频繁词集是全局频繁词集的一个子集;x 表示 D_j 中出现的既属于全局频繁词集集合又属于簇频繁词集集合的词集;x'表示只属于全局频繁词集集合而不属于簇频繁词集集合的词集;$n(\cdot)$是用来统计项集个数的函数.通过式(2)即可将文本对归到其最适合的初始簇中.

在树形结构中,每个节点代表 1 个类簇,为了得到指定数量的类簇或避免类簇描述过于具体,需要对树进行剪枝.其策略是:通过计算簇之间的相似度,将相似度大于相应阈值的 2 簇合并.公式如下:

$$Inter\_Sim(C_a\leftrightarrow C_b)=\left[Sim(C_a\leftarrow C_b)\times Sim(C_b\leftarrow C_a)\right]^{\frac{1}{2}}, \tag{3}$$

$$Sim(C_i\leftarrow C_j)=\frac{Score(C_i\leftarrow D(C_j))}{\sum_x n(x)+\sum_{x'} n(x')}+1, \tag{4}$$

式(3)中通过计算 C_b 到 C_a 和 C_a 到 C_b 的相似度并取几何平均来得到 2 簇间的相似度.式(4)为计算单向相似度的公式.簇间相似度阈值为 1,即将相似度大于 1 的 2 簇合并.通过这种方法实现对父子节点和兄弟节点的剪枝.FIHC 算法在聚类效果和时间效率方面均优于 FTC 算法.

1.3 MC 算法

MC 算法是一种直接对文本进行聚类的方法,利用文本中包含的频繁词集来度量文本之间的相似性.此方法运用频繁词集来表示每个文本,构建成文

本相似矩阵 $\boldsymbol{A}$,其中 $A[i][j]=MC\_Sim(D_i,D_j)$. MC 聚类的基本思想是:如果 2 个文本有最高的相似度,则这对文本应归入同一类簇中. 例如:如果文本 D_i 和 D_j 在同一簇中,若文本 D_k 和 D_j 有最高的相似度,则 D_k 将与 D_i 和 D_j 归入同一簇中.

算法 2. MC 算法.

输入:待聚类的文本集合;

输出:不同簇下的文本集合.

1) 根据相似度构建相似矩阵 $\boldsymbol{A}$;

2) 寻找 $\boldsymbol{A}$ 中除 0 之外的最小相似度值;

3) 寻找具有最大相似度的未被归类的文本对;

4) 如果步骤 3 中寻找的最大相似度值与步骤 2 中的最小相似度值相等,则相应的节点对组成一个新的类簇,否则执行下列步骤:①如果文本对中的 2 个文本中任何一个都未被归类,则将 2 个文本对构成一个新的类簇;②若文本对中 1 个文本已被归类,则另 1 个文本也同样归于此类;③将矩阵 $\boldsymbol{A}$ 中文本对所对应的值置 0,返回到步骤 3;

5) 如果存在还未被归类的文本,则将其组成一个新的类簇.

由于在文本聚类中文本的类数大部分是预先设定好的,因此需要对 MC 算法聚成的类簇进行合并,将不同簇中文本间的相似度值的总和作为簇间的相似度然后进行合并:

$$MC\_Sim(C_i,C_j)=\sum_{D_m\in C_i}\sum_{D_n\in C_j}MC\_Sim(D_m,D_n). \tag{5}$$

2 FIC 算法

FIC 算法框架图如图 1 所示. 该算法运用频繁词集来表示文本和衡量文本间的相似度,这些频繁词集同时可以反映出一定的语义关系,使得相同主题下的文本间相似度较之不同主题下的文本相似度更高,运用频繁词集表示的文本相较于传统的 VSM 模型能够有效地降低维度,减小数据稀疏的问题. 在构建文本网络后,可以引用社区划分或谱聚类的方法进行文本聚类,使得聚类方法更加灵活.

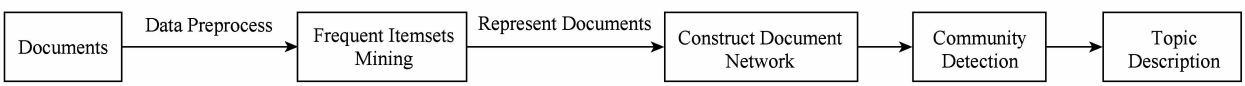

Fig. 1 Procedure of FIC

图 1 FIC 算法流程图

图 1 中的主要步骤为:数据预处理、频繁词集挖掘、文本表示、构建文本网络、社区划分和主题描述. 数据预处理对原始数据进行分词和去掉停用词;然后对处理完的数据进行频繁词集挖掘,将用频繁词集表示的文本构建成文本网络之后进行社区划分,每个社区即为文本聚类中的一个类簇,簇中的节点即为文本;再对划分好的簇进行主题描述. 下面将具体介绍算法中的关键步骤.

2.1 频繁词集挖掘

本文采用 FP-Growth[12] 算法来挖掘频繁词集,在进行频繁词集挖掘时,根据文本的数量来设置最小支持度,对进行特征选择之后的文本直接进行频繁词集挖掘. 下面给出定义:

定义 1. 文本数据库. 将对文本进行分词、去停用词之后的文本组成文本数据库 $D=\{D_1,D_2,\cdots,D_n\}$.

定义 2. 频繁词集. 通过频繁词集挖掘,得到频率大于最小支持度的频繁词集集合 $F=\{F_1,F_2,\cdots,F_m\}$,其中 F_i 表示 1 个频繁词集,1 个频繁词集中包含 1 个或多个词语 $F_i=\{w_1,w_2,\cdots,w_t\}$.

2.2 文本表示模型

较之传统的 VSM 模型,本文采用基于频繁词集的文本表示模型. 即用挖掘出的频繁词集来表示每个文本,构建文本-频繁词矩阵 $\boldsymbol{M}$. 其中矩阵 $\boldsymbol{M}$ 为 0-1 矩阵,通过衡量文本中是否含有频繁词集来赋值:

$$M[i][j]=\begin{cases}1, & F_j\in D_i,\\ 0, & F_j\notin D_i,\end{cases} \tag{6}$$

其中,$M[i][j]$表示矩阵 $\boldsymbol{M}$ 中文本 D_i 对于频繁词集 F_j 的值,若文本 D_i 中含有频繁词集 F_j,则 $M[i][j]=1$,否则 $M[i][j]=0$. 对于一个具体的文本可以直接将其中的特征单词用频繁词集来代替. 通过这种方法可以有效降低文本特征维度,解决了文本稀疏的问题. 频繁词集本身可以在某些程度上反映出文本的主题,它基于这样的假设:若某些词汇经常同时出现在同一文本中,则它们能够呈现出一定的语义关系,例如:{apple}表示水果类,{window}表示装饰类,若两词经常出现在同一文本,则作为频繁词集{apple,window}可以表示出计算机类. 因此以频繁

词集作为特征在度量文本间相似度时可以避免不同类别间的文本产生过大的相似度而影响聚类效果.

2.3 构建文本网络

FIC 算法将文本集中的每个文本当作文本网络中的节点,根据 2 文本之间的关联程度来建立边. 文本间的关联程度常常用它们之间的相似度来度量,已有的相似度量方法有很多,例如余弦相似度和欧氏距离. 本文采用 Jaccard 相似度来度量:

$$Jaccard\_Sim(D_i,D_j)=\frac{count(D_i\bigcap D_j)}{count(D_i\bigcup D_j)},\quad(7)$$

其中,$Jaccard_Sim(D_i,D_j)$表示文档之间的相似度,分子表示 2 个文档之间共有的频繁词集的个数,分母表示 2 个文档一共含有的频繁词集的个数. 通过此方法可以得到文本集中两两文本间的相似度权值,设置阈值 θ,若 2 边相似度大于 θ 则在 2 个文本间建立 1 条边,构建成 1 个文本网络 $G=\{V,E\}$,其中 V 为节点的集合,E 为网络中边的集合. 文本网络的构建,不仅使相似的文本间有边相连,而且也建立了文本与其他文本间的关联关系.

2.4 社区划分

对于上述构建的文本网络,我们需要对其进行社区划分,得到网络中不同的社团,通过文本与节点的对应关系可以将文本直接匹配到对应的社区,即类簇. 我们使用硬划分的社区划分方法,每个文本只能被指派到唯一的社区中,解决了类间重叠的问题. 本文中选用 K-rank-D[13] 与谱聚类[14] (spectral clustering)对网络进行划分.

K-rank-D 算法是一种网络社区划分算法,此算法在基于 K-means 的基础上,通过计算节点的 pagerank 中心性来进行初始中心的选择,解决了传统 K-means 算法中随机选取初始中心对聚类结果的影响. 此算法具有精度高的优点.

谱聚类是一种经典的图切割聚类方法,从图论的角度来说,聚类问题就相当于一个图的分割问题. 即给定一个图 $G=(V,E)$,顶点 V 表示各样本,带权的边表示各样本之间的相似度,谱聚类的目的是要找到一种合理的分割图的方法,使得分割后形成若干子图,连接不同子图的边的权重(相似度)尽可能低,同子图内的边的权重(相似度)尽可能高.

算法 3. 谱聚类算法.

输入:待聚类的文本集合;

输出:不同簇下的文本集合.

1) 根据数据生成图的邻接矩阵;

2) 根据邻接矩阵生成拉普拉斯矩阵并进行归一化;

3) 求最小的前 K 个特征值和对应的特征向量,其中 K 为聚类的类簇个数;

4) 将特征向量用 K-means 进行聚类.

因此,FIC 算法采用以上 2 种聚类方法,组成 FIC-S 和 FIC-K 这 2 种算法进行文本聚类.

2.5 主题描述

为了能够更加直观地了解每个类簇下的主题,我们需要对主题进行描述. 对于每一个类簇内的文本,统计其中频繁词集的出现频率,将出现频率较大的频繁词集作为主题的描述词.

3 实验结果与分析

为了验证本文提出的文本聚类方法,我们选用 3 个文本聚类中标准的数据集来进行实验,对照实验选用常用的基于 K-means 文本聚类方法[15]、SPK-means、MC、LDA、FIHC、GNMF 算法. 其中在 MC 算法中,根据不同的文本相似度量方案分成 MC-1,MC-2,MC-3. 这里我们选择精度较高的 MC-1 作为对照,在 MC-1 中 2 个文本间相似度的值为 2 个文本含有的共同频繁词集的个数.

3.1 数据集

本次实验采用文本聚类常用的标准数据集 20-Newsgroup① 和 Reuters-21578②,对于中文数据,选择文本分类语料库搜狗新闻数据③. 其中,20-newsgroup 数据包括近 20 000 篇新闻报道,分为 20 个不同的新闻组,除了小部分文档,每个文档都只属于一个新闻组;Reuters-21578 是文本分类的测试集,其中包含的文档来自于路透社 1987 年的新闻,搜狗新闻数据包括 9 个新闻类,共有 17 910 个文本.

由于 FIC 算法与 MC 同样采用频繁词集表示的策略,我们选择抽取和 MC 算法同样量级的数据,从以上数据集中随机抽取文档组成了 5 组测试集,测试集的详细信息如表 1～6 所示,其中 R12,R5,R8 来自数据集 Reuters-21578,N4 来自 20Newsgroup,C6 来自搜狗新闻数据.

① http://kdd.ics.uci.edu/databases/20newsgroups/

② http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

③ http://www.sougou.com/labs/resources/list_yuliao.php

Table 1 Dataset in the Experiment

表 1 实验中的数据集

Dataset	Number of Documents	Number of Clusters	Maximum Number of Documents in the Cluster	Minimum Number of Documents in the Cluster	Number of Different Words
N4	1 200	4	300	300	14 608
R5	2 274	5	991	130	11 082
R8	946	8	216	63	6 572
R12	1 712	12	326	44	9 072
C6	2 689	6	599	236	72 805

Table 2 Description of N4

表 2 N4 数据描述

Name of the Class	Number of Documents
Atheism	300
Baseball	300
Space	300
Mac. hardware	300

Table 3 Description of R5

表 3 R5 数据描述

Name of the Class	Number of Documents
Earn	726
Money-fx	130
Trade	216
Acq	991
Crude	211

Table 4 Description of R12

表 4 R12 数据描述

Name of the Class	Number of Documents
Cpi	44
Earn	228
Gold	68
Money-fx	155
Gnp	49
Coffee	88
Trade	225
Acq	326
Interest	114
Grude	224
Ship	106
Sugar	85

Table 5 Description of R8

表 5 R8 数据描述

Name of the Class	Number of Documents
Money-fx	130
Trade	216
Ship	91
Interest	85
Crude	211
Sugar	73
Coffee	77
Gold	63

Table 6 Description of C6

表 6 C6 数据描述

Name of the Class	Number of Documents
IT	549
财经	472
体育	599
健康	236
教育	458
军事	375

3.2 聚类评价指标

为了测试 FIC 算法的性能,我们采用文本聚类中常用的外部评价标准 F-measure. 它经常被用作衡量聚类方法的精度,是一种平面和层次聚类结构都适用的评价标准. F-measure 综合了召回率和准确率 2 种评价指标:

$$Recall(K_i,C_j)=\frac{n_{ij}}{|K_i|}, \tag{8}$$

$$Precision(K_i,C_j)=\frac{n_{ij}}{|C_j|}, \tag{9}$$

其中,n_{ij} 表示簇 C_j 中属于类 K_i 的文本数. 由召回率和准确率可得到表示簇 C_j 描述类 K_i 的能力计算:

$$F(K_i,C_j)=\frac{2\times Recall(K_i,C_j)\times Precision(K_i,C_j)}{Recall(K_i,C_j)+Precision(K_i,C_j)}, \tag{10}$$

聚类的总体 F-measure 值则可用每个类簇的最大 F-measure 值并采用该类簇的大小加权之后的总和:

$$F(C)=\sum_{K_i\in K}\left|\frac{K_i}{D_j}\right|\max_{C_j\in C}\{F(K_i,C_j)\}, \tag{11}$$

F-measure 值的取值范围在[0,1],值越大表示聚类效果越好.

3.3 实验设计方案

在进行数据的预处理时,我们运用特征选择[16]的方法. 文本经过分词和去停用词之后,依然会保留大量与主题不相关或无意义的单词,例如 Reuters 数据中的 Reuter,write 等和一些未过滤的作者姓名单词. 为了只保留那些对划分文本更有利的特征单词,本文采用文档-反文档频率(term frequency-inverse document frequency, TF-IDF)方法对文本中的单词进行进一步过滤. TF-IDF 用以评估 1 个单词对于 1 个文件或 1 个语料库的其中 1 份文件的重要程度. 词的重要度随着它在 1 个文本中出现的次数成正比增加,但同时会随着它在文本集中出现的频率成反比下降:

$$tfidf_{i,j}=tf_{i,j}\times idf_i, \tag{12}$$

$$tf_{i,j}=\frac{n_{i,j}}{\sum_k n_{k,j}}, \tag{13}$$

其中,$tf_{i,j}$表示单词 i 在文本 d_j 中的词频,分子是该词在文本 d_j 中出现的次数,分母则是在文本 d_j 中所有词的出现次数之和.

$$idf_i=\log\frac{|D|}{|\{j:t_i\in d_j\}|}, \tag{14}$$

其中,idf_i 表示该词的反文档频率,分子表示语料库中的文本总数,分母表示含词语 t_i 的文本数目. 实验中我们根据每个词的重要度分数进行排序,只保留前 3/4 的词. 通过此方法可以有效地过滤文本中大量无意义的词.

3.4 实验分析

我们在不同的算法上对 5 组数据进行聚类,其中对于 K-means,SPK-means,LDA,GNMF,FIC-S 这 5 种不确定型算法分别运行 10 次取平均值作为最后聚类的精度结果,每个算法的聚类结果如表 7 所示.

从表 7 中可以看出,FIC 算法较之其他方法在精度上有明显的优势. 究其原因主要有 3 个方面:

1) 在数据处理上,我们并不只是分词和去掉停用词,为了保留那些更有代表性的词汇,采用了通过 TF-IDF 值来筛选特征词.

2) 运用频繁词集来构建文本表示模型,文本集中挖掘出的频繁词集都在千量集以内,较之传统的 VSM 模型可将维数从上万维度降到只有几千维,另外,从文本集中挖掘的频繁词集本身带有一定的语义,在衡量文本间相似度时,相较单个特征词的衡量方法,能够考虑到频繁词中的语义关系,使得相同主题下的文本相似度更高且降低了不同主题下文本间的相似度. 相比于直接用 2 文本间共有的频繁词集的数量来衡量文本间的相似度,式(7)可以有效地避免类不平衡问题带来的影响. 当测试数据集中不同类间文本数量差距较大时,挖掘出的频繁词集集合中关于大类主题的频繁词集会比较多,而关于小类主题的频繁词集较少,因此在用频繁词集表示每个文本时会出现小类文本中所具有的频繁词集较少的情况,若直接按照共有频繁词集的数量来衡量相似度则会出现小类间文本连接不紧密的情况,在社区划分中会出现小类被大类吞并的问题. 为了更加准确地衡量文本间相似性,式(7)考虑到文本内频繁词集的数量,将相似度看做文本间共有的频繁词集个数在 2 文本中所有频繁词集里所占的权重比例. 通过这种方法使得小类间的文本连接更加紧密,减小被大类吞并的风险.

3) 引用网络和图论的思想,将文本当作网络中的节点,建立文本建的关联关系,使文本与文本之间不再是独立的两两关系,并且使用文本网络的分析方法可以更加直观地分析文本集中的类结构. 同时可以引入更多基于网络社区划分和图切割的聚类算法,使得 FIC 算法更加灵活.

Table 7 Result of Algorithms

表 7 算法精度

Algorithms	F-measure				
	N4	R5	R8	R12	C6
K-means	0.572 1	0.777 4	0.409 9	0.484 8	0.415 2
SPK-means	**0.968 8**	0.693 6	0.757 7	0.6942	**0.581 8**
GNMF	0.916 0	0.412 3	0.583 4	0.479 4	0.486 5
LDA	0.953 2	0.719 5	0.514 5	0.619 0	0.508 0
FIHC	0.80	0.704 0	0.659 9	0.613 7	0.453 6
MC-1	0.593 6	0.460 2	0.642 1	0.484 2	0.426 6
FIC-K	0.950 7	**0.857 6**	**0.814 6**	**0.736 4**	0.567 7
FIC-S	0.587 6	0.763 4	0.716 2	0.653 0	0.509 4

3.5 实验中的阈值调整

本文算法中主要涉及到3个参数,包括在筛选特征词时的阈值、挖掘频繁词集中的最小支持度和计算文本间相似性的相似度阈值.本文通过采用手动调整、多次实验的方式,获得了聚类的最佳效果.其中对于最小支持度的选择要尽量保证每个文本能够至少有5个频繁词集来表示,相似度阈值则通过文本网络中的边数来衡量,避免文本网络过于密集或稀疏,在最佳效果下各个阈值的取值如表8所示.下面我们将以对聚类精度影响最高的相似性阈值为例,介绍本次实验过程.在英文数据集中,我们选取相似度阈值的步长为0.005,由于中文数据中的停用词较多以及每个文档中词数量较少,因此采用与英文数据不同的阈值选择方式,采用0.01的步长进行实验.其中FIC-K算法在处理无权网络时精度较之带权网络高,因此我们将阈值筛选后的文本网络构成一个无权网络作为输入数据,即将权值大于θ的权值置1,小于θ的权值置0.对于FIC-S算法使用带权网络,权重为文本间的相似度.图2和图3为在固定最佳最小支持度阈值和特征筛选阈值后FIC-K算法在不同相似度阈值下的精度情况.

Table 8 Thresholds in the Data

表8 数据中的阈值

Dataset	Minsupport	Similarity Threshold	Feature Selection Threshold
N4	0.020	0.005	0.75
R5	0.020	0.015	0.75
R8	0.020	0.015	0.75
R12	0.025	0.010	0.75
C6	0.030	0.060	0.50

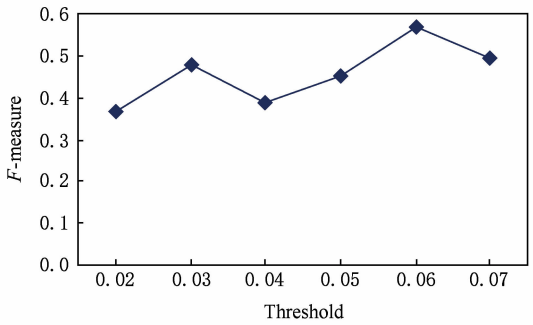

Fig. 2 Accuracy of FIC-K for Chinese data on C6

图2 FIC-K算法在C6数据集的中文数据下的精度

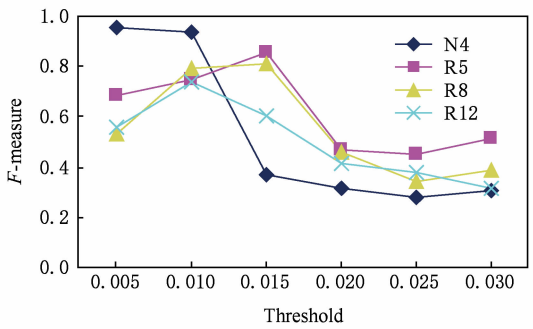

Fig. 3 Accuracy of FIC-K for English data

图3 FIC-K算法在英文数据集上的精度

3.6 实验中对文本主题的描述

对于本次实验选取的5组数据,我们采用相同的主题描述方案.对每个类簇内的文本,统计所有文本内的频繁词集的出现频率,并选择按频率排名前10的频繁词集来描述每个主题.这里主要展示由FIC-K算法所聚成类簇的主题描述情况,同时与LDA算法得到的主题描述词进行对比,结果如表9所示.从表9中可看出,FIC算法得到的主题描述词大多围绕主题,用频繁词集合的形式来描述,在一定程度上比起LDA的单个词描述可以将类簇表述的更加详细具体.

Table 9 Topic Description

表9 主题描述

Dataset	Class	FIC-K	LDA
N4	Atheism	{god}, {peopl}, {islam}, {moral}, {atheism}, {atheist}, {keith}, {jon}, {livesei}, {argument}	God, people, islam, moral, atheism, jon, evid, keith, atheist, exist
	Baseball	{game}, {team}, {hit}, {baseball}, {player}, {win}, {season}, {brave}, {pitch}, {score}	Game, team, hit, player, baseball, win, season, plai, brave, pitch
	Space	{space}, {nasa}, {orbit}, {lunar}, {nasa, sapce}, {moon}, {launch}, {satellit}, {shuttl}, {henri}	Space, nasa, orbit, moon, lunar, gov, satellite, earth, henri, scienc
	Mac. hard-ware	{appl}, {mac}, {drive}, {centri}, {quadra}, {simm}, {video}, {monitor}, {chip}, {ram}	Appl, mac, drive, video, centri, monitor, quadra, simm, speed, machin

Continued (Table 9)

Dataset	Class	FIC-K	LDA
R5	Earn	{mln},{net,mln},{profit},{dlr},{net},{ct},{loss},{earn},{oper},{billion}	Mln, profit, net, dlr, ct, loss, quarter, earn, oper, billion
	Money-fx	{money,fx},{currenc},{bank},{dollar},{rate},{moneyfx,currenc},{treasuri},{exchange},{central,bank},{currenc,dollar}	Share, offer, stock, bid, tender, board, merger, takeov, file, outstand
	Trade	{trade},{japan,trade},{tariff},{tariff,trade},{japans,trade},{reagan},{import},{export},{offici},{export,trade}	Trade, japan, bank, market, japanes, dollar, foreign, currenc, rate, import
	Acq	{share},{offer},{common},{stock},{stake},{outstand},{merger},{common,share},{board},{commiss}	Stake, subsidiari, sell, complet, purchas, asset, share, bui, common, acquir
	Crude	{oil},{barrel},{crude},{barrel,oil},{crude,barrel},{energi},{openc,oil},{crude,barrel,oil},{bpd},{product}	Oil, barrel, crude, price, petroleum, energy, ga, opec, product, bpd
R8	Interest	{bank},{rate},{pct},{rate,bank},{pct,rate},{pct,bank},{billion},{lend},{lend,bank},{billion}	Bank, rate, pct, billion, money, fund, interest, market, mln, liquid
	Trade	{trade},{japan},{japan,trade},{retali},{semiconductor},{japanes,trade},{retail,trade},{import},{tariff},{agreement}	Trade, japan, japanes, import, industry, reagan, offici, unit, retail, semiconductor
	Ship	{ship},{port},{strike},{union},{strike,union},{worker},{vessel},{pai},{cargo},{protest}	Ship, strike, union, port, worker, vessel, cargo, pai, labour, spokesman
	Money-fx	{dollar},{currrenc},{rate},{exchange},{pari},{currenc,dollar},{bank},{axchange,rate},{dollar,rate},{stabil}	Dollar, polici, econom, pari, rate, nation, exchange, ecnomi, currenc, stabil
	Crude	{oil},{barrel},{crude},{barrel,oil},{crude,oil},{price},{opec},{bpd},{energi},{opec,oil}	Oil, opec, barrel, bpd, crude, mln, price, energi, Saudi, arabia
	Sugar	{sugar},{tonn},{sugar,tonn},{export},{mln},{billion},{dlr},{import},{januari},{figur}	Mln, billion, dlr, tonn, export, figur, januari, import, sugar, februari
	Coffee	{coffe},{quota},{coffe,quota},{ico},{ico,coffee},{export},{ico,quota},{coffe,ecport},{produc},{brazil}	Coffe, export, produc, quota, price, London, trader, brazil, ico, talk
	Gold	{gold},{ounc},{ounc,gold},{mine},{mine,gold},{ton},{ton,glod},{ounc,mine},{ton,gold,ounc},{ton,ounc}	Gold, compani, mine, explor, ounc, ton, drill, resource, corp, reserv
R12	Cpi	{monei},{market},{money,market},{stg},{england},{shortag},{bill},{mln},{stg,mln},{bank}	Pct, billion, year, mln, stg, quarter, export, bank, januari, rise
	Earn	{mln},{net},{dlr},{ct},{net,mln},{loss},{shr},{ct,net},{net,dlr},{profit}	Mln, dlr, loss, ct, net, year, oper, profit, company, shr
	Gold	{gold},{mine},{mine,gold},{ounc,gold},{ounc},{ton},{resourc},{grade},{drill},{reserv}	Gold, mine, ounc, ton, company, product, Canada, per, pct, feet
	Money-fx	{currenc},{dollar},{rate},{pari},{exchange},{currenc,dollar},{treasuri},{stabil},{secretari},{stabil}	Bank, dollar, market, exchange, rate, currenc, foreign, polici, economy, debt
	Gnp	{bank},{fund},{billion},{dollar},{govern},{rate},{fund,bank},{dollar,bank},{feder},{monetari}	Minist, countri, meet, talk, gatt, agreement, year, world, gener, state
	Coffee	{coffe},{coffe,quota},{quota},{export},{coffe,export},{ico,coffe,quota},{produc},{quota,export},{brazil},{brazil,coffe}	Coffe, price, quota, export, opec, produc, brazil, market, mln, meet
	Trade	{trade},{japan},{japan,trade},{tariff},{tariff,trade},{offici},{reagan},{washington},{surplu},{offici,trade}	trade, japan, japanes, import, office, state, market, industry, Reagan, unit
	Acq	{share},{acquir},{inc},{acquisit},{compani},{sharehold},{offer},{stock},{common},{complet}	Share, compani, offer, dlr, stock, pct, corp, inc, group, sharehold
	Interest	{rate},{pct},{rate,bank},{bank},{rate,pct},{rate,bank,pct},{point},{point,rate},{bank,pct},{lend,rate}	Pct, rate, bank, price, februari, point, year, march, base, interest
	Crude	{oil},{barrel},{barrel,oil},{crude},{price},{crude,oil},{oil,price},{energi},{opec},{crude,barrel}	Oil, crude, barrel, dlr, price, energi, bpd, company, petroleum, product
	Ship	{ship},{union},{port},{strike},{worker},{pai},{disput},{spokesman},{brazil},{transport}	Ship, union, port, china, strike, spokesman, worker, vessel, seamen, offici
	Sugar	{pct},{sugar},{tonn},{year},{rose},{januari},{sugar,tonn},{compar},{rose,pct},{export}	Sugar, ton, mln, year, export, price, trader, import, produc, intervent

Continued (Table 9)

Dataset	Class	FIC-K	LDA
C6	体育	{重奖,世青赛},{世青赛,进门},{赛季,球员},{比赛},{冠军,酷暑},{快捷,忘杯},{冠军},{运动场},{严正声明,假球},{联赛}	比赛,球场,冠军,新华,队员,球队,联赛,主场,球员,赛季
	健康	{迟钝,高血压},{患者,高血压},{胃癌,倍受},{乳腺,作痛},{女性,性观念}{上身},{背痛},{医院,药物},{治疗,疾病},{医院}	治疗,患者,疾病,女性,医院,药物,身体,感染,人体,分泌
	教育	{业务培训},{拷问},{教育,大学},{全国,考生},{考试},{学校},{市场},{招聘,社会},{大学生},{报考}	大学,专业,教育,管理,考试,学生,考生,工作,学校,社会
	军事	{军事行动},{中央军事委员会},{空军},{陆军,装备},{导弹},{空军,日本},{海军},{核计划,武器},{作战,战斗},{军事,系统}	美国,系统,作战,部队,飞机,安全,战斗,武器,国防,装备
	财经	{资本运作},{资本运作,保荐人},{争辩},{特别法},{产品,企业},{市场},{投资,市场},{资金},{中国,销售},{利润率}	公司,市场,机构,本人,利害,文章,投资,产品,企业,增长
	IT	{下载安装},{操作系统,操作},{手机,市场},{传感,自动},{移动},{美国},{公司},{微软,品牌},{数码},{信息,互联网}	安全,技术,股市,手机,中国,服务,移动,数码,测试,战略

4 结束语

本文提出一种新的文本聚类方法 FIC,该方法运用基于频繁词集的文本表示模型,解决了传统的 VSM 模型的高维和数据稀疏的问题,采用基于网络的社区划分聚类方法和谱聚类算法,由于考虑了多个文本间的关系,聚类性能相比于之前的方法有了一定程度的提升. 至今文档聚类一直是一个值得深入研究的课题,其中面临着两大问题:1)数据集规模海量增长;2)文本越来越趋向于短文本的形式. 如何在这类数据集上进行有效的文本聚类是我们未来工作的挑战.

参 考 文 献

[1] Lu Yuchang, Lu Mingyu, Li Fan, et al. Analysis and construction of word weighting function in VSM [J]. Journal of Computer Research and Development, 2002, 39(10): 1205-1210 (in Chinese)
(陆玉昌, 鲁明羽, 李凡, 等. 向量空间法中单词权重函数的分析与构造[J]. 计算机研究与发展, 2002, 39(10): 1205-1210)

[2] Peng Min, Huang Jiajia, Zhu Jiahui, et al. Mass of short texts clustering and topic extraction based on frequent itemsets [J]. Journal of Computer Research and Development, 2015, 52(9): 1941-1953 (in Chinese)
(彭敏, 黄佳佳, 朱佳晖, 等. 基于频繁词集的海量短文本聚类与主题抽取[J]. 计算机研究与发展, 2015, 52(9): 1941-1953)

[3] Beil F, Ester M, Xu Xiaowei. Frequent term-based text clustering [C] //Proc of the 8th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2002: 436-442

[4] Fung M, Wang Ke, Ester M. Hierarchical document clustering using frequent itemsets [C] //Proc of the 3rd SIAM Int Conf on Data Mining. San Francisco, CA: SIAM, 2003: 59-70

[5] Zhang Wen, Yoshida T, Tang Xijin, et al. Text clustering using frequent itemsets [J]. Konwledge-Based Systems, 2010, 23(5): 379-388

[6] Hernández-Reyes E, García-Hernández R A, Carrasco-Ochoa J A, et al. Document clustering based on maximal frequent sequences [G] //Advances in Natural Language Processing. Berlin: Springer, 2006: 257-267

[7] Masada T, Kiyasu S, Miyahara S. Comparing LDA with pLSI as a dimensionality reduction method in document clustering [G] //Large-Scale Knowledge Resource. Berlin: Springer, 2008: 13-16

[8] Hornik K, Feinerer I, Kober M, et al. Spherical k-means clustering [J]. Journal of Statistical Software, 2012, 50(10): 1-22

[9] Cai Deng, He Xiaofei, Han Jiawei, et al. Graph regularized bonnegative matrix factorization for data representation [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2011, 33(8): 1548-1560

[10] Hoyer P O. Non-negative matrix factorization with sparseness constraints [J]. Journal of Machine Learning Research, 2004, 5(1): 1457-1469

[11] Xiu Yu, Wang Shitong, Zhu Lin, et al. Maximum-entropy sphere K-means document clusterinf analysis [J]. Frontiers of Computer Science and Technology, 2007, 1(3): 331-339 (in Chinese)
(修宇, 王士同, 朱林, 等. 极大熵球面 K 均值文本聚类分析[J]. 计算机科学与探索, 2007, 1(3): 331-339)

[12] Han Jiawei, Pei Jian, Yin Yiwen. Mining frequent pattern without candidate generation: A frequent-pattern tree approach [J]. Data Mining and Knowledge Discovery, 2004, 8(53): 53-87

[13] Li Yafang, Jia Caiyan, Yu Jian. A parameter-free community detection method based on centrality and dispersion of nodes in complex networks [J]. Physica A Statistical Mechanics & Its Applications, 2015, 438(43): 321-334

[14] Luxburg U V. A tutorial on spectral clustering [J]. Statistics & Computing, 2007, 17(17): 395-416

[15] Steinbach M, Karypis G, Kumar V. A comparison of document clustering techniques [C] //Proc of the 2nd KDD Workshop on Text Mining. New York: ACM, 2000: 525-526

[16] Huang Yuyan. Research and application of text clustering method based on maximal frequent itemsets *K*-means [D]. Harbin: Harbin Institute of Technology, 2011 (in Chinese)
(黄玉燕. 基于最大频繁词集 *K*-means 的文本聚类算法研究及应用[D]. 哈尔滨: 哈尔滨工业大学, 2011)

Zhang Xuesong, born in 1993. Master candidate in Beijing Jiaotong University. His main research interests include text clustering, community detection in networks.

Jia Caiyan, born in 1976. PhD and professor in Beijing Jiaotong University. Her main research interests include social computing, text clustering, community detection in networks, etc.

2018 年《计算机研究与发展》专题(正刊)征文通知

——新型存储系统前沿技术

近年来,随着国家和社会信息化发展的不断加速,对信息存储的提出了越来越高的要求.一方面,大数据时代,数据存储的规模和处理需求越来越高,亟需新型存储系统和技术以提供更高的性能和更好的可扩展性.另一方面,由于各种人工智能系统及相关技术的出现,现有的存储技术和系统难以满足上层系统和技术的需求.因此,存储系统结构技术研究面临诸多新的机遇和挑战.

《计算机研究与发展》开辟了"计算机体系结构前沿技术"系列,并计划于 2018 年出版"新型存储系统结构前沿技术"专题.本专题将集中讨论面向非易失存储器件的存储体系结构技术、海量信息存储系统研究与技术、新型工艺支撑的计算机存储体系结构技术、存储前沿技术等,相关研究可以涉及不同应用领域、环境、场景的存储系统,也可以涉及存储系统的不同层次和部件,还可以包括基于新型工艺和器件的存储结构设计等.

本次专题受到中国计算机学会信息存储专委会的支持.专题录用的文章作者将被邀请参加 2018 年 9 月全国信息存储学术会议,邀请作者到会演讲并参与讨论.欢迎相关领域的专家学者和科研人员踊跃投稿.现将专题论文征集的有关事项通知如下.

征文范围(但不限于)

1) 面向非易失存储器件的存储体系结构技术,如:面向非易失存储器件的持久性内存(memory)和存储(storage)结构与技术;面向持久性内存的编程模型、空间管理、文件系统和数据库等;固态 SSD 存储与技术;新型存储系统及其应用技术.

2) 海量信息存储系统技术,如:高性能计算机存储系统、分布式文件系统;网络存储、数据存储网格及相关技术;海量信息存储技术;网络存储系统可用性、安全性、可靠性及容灾研究;重复数据删冗、归档、分级等存储技.

3) 新型工艺支撑的计算机存储体系结构技术,如:非易失存储器结构技术;PIM (Processing in Memory)技术;3-维堆叠结构技术.

4) 存储前沿技术,如:软件定义存储与存储超融合;类脑存储;量子存储.

稿件内容也可以为上述所列内容的交叉.本刊也鼓励作者讨论与上述领域相关但此处未予提及的重要问题、创造性的研究思路和探索视角.

投稿要求

1) 论文应属于作者的科研成果,未在国内外公开发行的刊物或会议上发表,不存在一稿多投问题.作者在投稿时,需向编辑部提交版权转让协议.

2) 论文一律用 Word 格式排版,格式体例参考近期出版的《计算机研究与发展》的要求(http://crad.ict.ac.cn/下载区域).

3) 论文通过期刊网站(http://crad.ict.ac.cn)投稿,投稿时需提供作者的联系方式.作者投稿时请务必在留言中注明"存储系统前沿 2018 专题"(否则按自由来稿处理).

重要日期

征文截止日期:2018 年 4 月 5 日　　录用通知日期: 2018 年 5 月 31 日

修改稿提交日期:2018 年 6 月 15 日　　出版日期:2018 年 9 月

特邀编委

刘志勇　研究员　中国科学院计算技术研究所　zyliu@ict.ac.cn

舒继武　教　授　清华大学　shujw@tsinghua.edu.cn

联系方式

编　辑　部:crad@ict.ac.cn,010-62620696,010-62600350

通信地址:北京 2704 信箱《计算机研究与发展》编辑部　　邮政编码:100190