

# 一种基于语义内积空间模型的文本聚类算法

彭 京<sup>1),2)</sup> 杨冬青<sup>1)</sup> 唐世渭<sup>1)</sup> 付 艳<sup>1)</sup> 蒋汉奎<sup>2)</sup>

<sup>1)</sup>(北京大学信息科学技术学院 北京 100871)

<sup>2)</sup>(成都市公安局信息通信处 成都 610017)

**摘 要** 现有数据聚类方法在处理文本数据,尤其是短文本数据时,由于没有考虑词之间潜在存在的相似情况,因此导致聚类效果不理想.文中针对文本数据高维度和稀疏空间的特点,提出了一种基于语义内积空间模型的文本聚类算法.算法首先利用内积空间的定义建立了针对中文概念、词和文本的相似度度量方法,然后从理论上进行了分析.最后通过一个两阶段处理过程,即向下分裂和向上聚合,完成文本数据的聚类.该方法成功用于中文短文本数据的聚类.实验表明相对于传统方法,文中提供的方法聚类质量更好.

**关键词** 内积空间;文本聚类;概念相似度;相似计算;数据挖掘

中图法分类号 TP181

## A Novel Text Clustering Algorithm Based on Inner Product Space Model of Semantic

PENG Jing<sup>1),2)</sup> YANG Dong-Qing<sup>1)</sup> TANG Shi-Wei<sup>1)</sup> FU Yan<sup>1)</sup> JIANG Han-Kui<sup>2)</sup>

<sup>1)</sup>(School of Electronics Engineering and Computer Science, Peking University, Beijing 100871)

<sup>2)</sup>(Information and Communication Department, Chengdu Public Security Bureau, Chengdu 610017)

**Abstract** Due to lack considering the latent similarity information among words, the clustering result using exist clustering algorithms in processing text data, especially in processing short text data, is not ideal. Considering the text characteristic of high dimensions and sparse space, this paper proposes a novel text clustering algorithm based on semantic inner space model. The paper creates similarity method among Chinese concepts, words and text based on the definition of inner space at first, and then analyzes systematically the algorithm in theory. Through a two phrase processes, i. e. top-down "divide" phase and a bottom-up "merge" phase, it finishes the clustering of text data. The method has been applied into the data clustering of Chinese short documents. Extensive experiments show that the method is better than traditional algorithms.

**Keywords** inner product space; text clustering; concept similarity; similarity computing; data mining

## 1 引 言

近年来随着互联网的发展,人们可以访问的信

息也呈几何级数增长,如何从海量文本信息中获取知识是当前计算机信息理论研究的重点.聚类作为数据挖掘的重要分支,近年来逐步引起广泛的重视.相对于其他数据挖掘方法,聚类具有无需先验知识

收稿日期:2007-03-06;修改稿收到日期:2007-06-04. 本课题得到国家自然科学基金(6473051,60503037)、中国博士后科学基金(20060400002)、四川省青年科技基金(2007Q14-055)、国家“八六三”高技术研究发展计划项目基金(2006AA01Z230)和北京市自然科学基金(4062018)资助. 彭 京,男,1973年生,博士,主要研究方向为数据库与自然语言处理. E-mail: pjxxlpsj@hotmail.com; pj@pku.edu.cn. 杨冬青,女,1945年生,教授,博士生导师,主要研究领域为数据库与信息理论. 唐世渭,男,1939年生,教授,博士生导师,主要研究领域为数据模型、数据库系统、数据仓库、数据挖掘. 付 艳,女,1980年生,博士研究生,主要研究方向为数据库与数据挖掘. 蒋汉奎,男,1966年生,硕士,高级工程师,主要研究方向为软件工程与信息系统.

的优势,可以根据数据自然分布而获取知识.聚类方法在计算机界的研究有较长历史,这方面已有大量的算法,如 X-means<sup>[1]</sup>,G-means<sup>[2]</sup>,CLARANS,CURE,CLIQUE,BIRCH,DBSCAN<sup>[3]</sup>等等.

当前在网络环境下大量存在着短文本信息,诸如新闻标题、摘要、短消息等等.这类信息具有非常高的维度和稀疏空间的特点.随着互联网发展,针对短文本信息的数据聚类,逐步引起越来越多的关注.

在采用传统数据挖掘方法处理文本数据之前,必须首先将文本转换为向量空间模型或者后缀树模型等等.这些模型从不同角度使用不同的方法处理特征加权、类别学习等问题,而其中向量空间模型是最有效的模型之一.

Gerard Salton 在 20 世纪 60 年代提出采用向量空间模型进行文本特征表达,用 TFIDF(Term-Frequency Inverse-Document-Frequency)将文档转化为向量形式,然后在向量空间中计算文本相似度.在这方面近期的研究成果参见文献[4-9],国内相关研究工作参见文献[10-12].

在基于 TFIDF 的向量空间模型中,由于没有考虑词之间存在的概念相似情况,因此影响了数据聚类的准确性,尤其在文档较短的条件下,比如新闻的标题和摘要,得到的相关度与实际情况偏离严重.例如:“我爱吃苹果”和“她更喜欢香蕉”这两句话都描述了对水果的爱好,应该说具有很强的相关性.但在向量空间模型中,因为没有相同的词汇,计算出的相关度为 0.这样最终的聚类结果也就与人们的直观感受相去甚远.文献[10]基于知网模型,提出了一种相似度计算方法,但该方法只能用于词和概念的相似度计算,没有提供文本相似计算分析,同时缺少对计算公式的理论分析,因此难以用于聚类分析.

针对中文短文本聚类问题,本文基于语义内积空间模型提出了一种新的文本聚类算法——TCIS(Text Clustering algorithm based on Inner product space model of Semantic).该方法首先基于董振东、董强提出的知网概念模型<sup>①</sup>,提出了一种语义内积空间模型;然后根据语义内积空间模型推导出概念、词和文本相似度的计算方法;最后根据文本相似度的计算方法,提出了一种两阶段算法<sup>[13]</sup>完成短文本数据的聚类.与传统基于向量空间模型方法相比,计算公式更加合理,得到的结果更符合语义的判断.通过实验测试,本文的方法比基于向量空间模型的算法聚类质量更好.

本文第 2 节介绍了相关背景知识及文本向量空间模型;第 3 节给出了语义内积空间模型的计算公式及分析;第 4 节根据内积空间模型的概念,提出了两阶段 TCIS 聚类算法;第 5 节是本文的试验部分;第 6 节总结了全文,并提出了下一步的研究方向.

## 2 背景知识

### 2.1 向量空间

在向量空间模型中,每个文档或者字符串序列使用词向量来建立一个向量空间.对于一个文档,进行分词以及停用词等语言处理以后,可以得到一个词语(或短语)序列,每个词语(或短语)都在文档中有相关的权重信息,一个  $m$  维空间向量就可以建立起来,其中  $m$  表示文档中不同的词语(或短语)的数量,这个  $m$  维的空间向量就是这些不同的词语(或短语)的空间序列表示.例如文档  $D$  为“她”,“喜欢”,“吃”,“水果”四个词语组成,取每个词语的权重为 1,则文档的向量空间可以用  $V_d = \langle 1, 1, 1, 1 \rangle$  表示.

在文档向量空间模型中,每个词的权重信息可以用下面的两个参数衡量:词频(term frequency)  $f_{i,j}$ ,表示词  $w_j$  在文档  $d_i$  中出现的频率次数.逆文本词频(inverse document frequency)  $\log \frac{N}{n_j}$ ,其中  $N$  表示文本集中所有的文档数量, $n_j$  表示文本集中所有含有词语  $w_j$  的文档数量.文档  $d_i$  中词  $w_j$  的权重通常使用  $t_{i,j} = f_{i,j} \cdot \log \frac{N}{n_j}$  表示.这个权重被称为 TFIDF 权重.

建立文档向量空间模型以后,可以利用向量相似性函数计算文档之间的相似性程度.刻画文档之间的相似性主要有以下两类函数:距离函数和相似系数.其中距离函数是通过使用文档向量空间模型,把每个文档看作  $n$  维向量空间中的一个点,进而使用某种距离来表示文档之间的相似性.距离较近的文档之间较相似,距离较远的文档之间差异较大.常用的距离度量方法包括欧几里得距离、曼哈坦距离和明考斯基距离.而相似系数不同,当两个文档之间越相似,则相似系数值越接近 1;文档之间越不相似,则相似系数值愈接近 0.这样就可以使用相似系数值来刻画文档之间的相似程度.常用两个文档向量之间的余弦系数表示文档间的相似系数.余弦系数通常使用下面的公式进行计算

① 董振东,董强.知网. <http://www.keenage.com>

$$Sim(d_i, d_j) = Cos(d_i, d_j) = \frac{\sum_{k=1}^{|T|} t_{i,k} \times t_{j,k}}{\sqrt{\sum_{k=1}^{|T|} t_{i,k}^2 \times \sum_{k=1}^{|T|} t_{j,k}^2}}$$

(1)

其中  $t$  分别代表两个文档向量空间的 TFIDF 权重,  $T$  表示词集合.

2.2 知网结构<sup>[10]</sup>

《知网》是一个以汉语和英语的词语所代表的概念为描述对象,以揭示概念与概念之间以及概念所具有的属性之间的关系为基本内容的常识知识库.

《知网》中有两个主要的概念:“概念”与“义原”. “概念”是对词汇语义的一种描述. 每一个词可以表达为几个概念. “概念”是用一种“知识表示语言”来描述的,这种“知识表示语言”所用的“词汇”叫做“义原”. “义原”是用于描述一个“概念”的最小意义单位.

与一般的语义词典[如《同义词词林》或 Wordnet]不同,《知网》并不是简单地将所有的“概念”归结到一个树状的概念层次体系中,而是试图用一系列的“义原”来对每一个“概念”进行描述.

义原一方面作为描述概念的最基本单位,另一方面,义原之间又存在复杂的关系,如:上下位关系、同义关系、反义关系、对义关系、属性-宿主关系、部件-整体关系、材料-成品关系、事件-角色关系等等. 可以看出,义原之间组成的是一个复杂的网状结构,而不是一个单纯的树状结构. 不过,义原关系中最重要 的还是上下位关系. 根据义原的上下位关系,所有的“基本义原”组成了一个义原层次体系. 例如义原“人”在义原层次树上的分类依次为:entity|实体→thing|万物→physical|物质→animate|生物→AnimalHuman|动物→human|人.

3 语义内积空间模型

本节以知网为基础,逐步地推导中文语义的内积空间模型. 首先利用“概念”的相互关系构建得到“概念”的内积空间,通过概念内积空间的性质得到任意概念间的相似关系度量;然后,根据词由概念组成的特点,推导词的相似关系度量;最后,以词作为文本的组成,来构建文本的内积空间,通过文本内积空间的性质得到任意文本间的相似关系度量. 具体内容如下.

3.1 概念内积空间

根据知网的模型,每个词由 1 个和多个“概念”组成,而每个“概念”由 1 个或多个“义原”组成. 我们首先根据知网定义的义原关系模型,定义义原之间的相关度计算公式.

3.1.1 义原相关度计算

在《知网》中,一共描述了义原之间的多种关系,根据这些关系,义原之间构成了一个概念网状结构. 但义原之间最基本的是上下位关系,根据上下位关系可以得到一棵义原概念树.

首先,我们根据《知网》定义的义原之间各种关系确定距离值,然后根据这些距离值推导任意义原之间的最短距离;最后,由距离定义义原的相关度.

设义原集合为  $M$ ,义原数量表示为  $|M|$ ,义原用  $p_i$  表示,  $i=1,2,\cdots,|M|$ .

对于义原上下位关系,我们定义义原  $p_i$  与其上位义原,即父节点距离为

$$d(p_i, parent(p_i)) = y - L_i \cdot x$$

(2)

其中  $L_i$  为义原  $p_i$  在概念树中的分类深度,  $y$  为距离初始阈值,  $x$  为一正实数,满足  $y > \max(L) \cdot x$ . 而义原间的其他关系定义为

$$d_k(p_i, p_j) = \omega_k \cdot (y - \max(L_i, L_j) \cdot x)$$

(3)

其中  $i, j$  表示两个义原,  $\omega_k$  表示第  $k$  种关系对应的权重,通常取  $\omega_k \geq 1$ .

这样,根据《知网》中的定义,我们可以得到义原上下位关系集合  $P = \{ \langle n_i, p(n_i), d_i \rangle \mid p_i \in M \}$  以及义原其他关系集合

$$Q = \{ \langle p_i, p_j, d_k(p_i, p_j) \rangle \mid p_i, p_j \in M, k \in Z \}.$$

从式(2)、(3)可以看到,义原在义原层次树的分类深度越深,则距离越小,也就是越相似,这与人的直观印象一致. 根据  $P, Q$  两个集合可以推导出任意义原的距离. 算法处理过程如下.

**算法 1.** 求取任意义原间最短距离算法.

输入:  $P, Q$

输出: 最短距离矩阵 Matrix

1. 建立  $\forall p_i, p_j \in M, (i, j = 1, 2, \cdots, |M|)$  间初始矢量距离矩阵  $Matrix(|M| \times |M|)$ ;
2. 依次读取  $P, Q$  集合中的元素  $t$ , 设  $t = \langle p_s, p_d, d_{s,d} \rangle$ ;
3. 判断加入  $t$  这条路径后,其他义原与义原  $t.p_s$  或  $t.p_d$  间是否存在更小的距离,有则替换,同时将其加入到一个变更属性集合  $R$ ;
4. 然后,根据刚才产生的变更属性集合  $R$ ,求彼此间在加入  $t$  这条路径后,是否有更小的距离,有则将距离矩阵中对应值替换为  $t.d_{s,d}$ ;

5. 直到  $P, Q$  集合所有的元素处理完成;

6. 返回最短距离矩阵  $Matrix$ .

在得到任意义原的最短距离后,将距离转换为义原的相似度,转换公式为

$$Sim(p_i, p_j) = \frac{\alpha}{d(p_i, p_j) + \alpha} \quad (4)$$

其中  $d(p_i, p_j)$  表示  $p_i, p_j$  两个义原间的最短距离,  $\alpha$  为一正常数,实验中  $\alpha$  取 1. 可以看到  $Sim(p_i, p_i) =$

$\frac{\alpha}{d(p_i, p_i) + \alpha} = \frac{\alpha}{\alpha} = 1$ , 即义原与自己的相似度为 1.

**例 1.** 义原“人”与“走兽”,经计算得到的最短距离为 4.4, 相似度为 0.185, 具体算法采用的参数参见实验介绍.

本文还具体比较不同的转换公式的效果,如 Leacock 和 Chodorow 于 1998 年提出的相似计算公式

$$Sim(p_i, p_j) = -\log \frac{d(p_i, p_j)}{2 \times D} \quad (5)$$

其中  $D$  表示概念树的最大深度,  $d(p_i, p_j)$  表示概念间最短路径(通过节点计数的结果),由于效果相差不明显,因此实验部分仍采用式(4).

### 3.1.2 概念内积空间和词相似度

词是由概念组成,概念相似度的计算基于义原间相似度.我们首先定义概念的内积空间.

**定义 1**(概念内积). 设义原  $p_i$  为线性空间上的点,概念  $c = \langle p_{c1}, p_{c2}, \dots, p_{cu} \rangle$ ,  $d = \langle p_{d1}, p_{d2}, \dots, p_{dv} \rangle$  则概念  $c, d$  的内积为

$$(c, d) = \sum_{j=1}^v \sum_{i=1}^u Sim(p_{ci}, p_{dj}) \quad (6)$$

根据概念内积,可导出概念范数定义.

**定义 2**(概念范数及概念相似度). 定义概念  $c$  的范数为

$$\|c\| = \sqrt{(c, c)} \quad (7)$$

其中  $\beta$  为正常数.

令概念  $c, d$  的相似度为

$$Sim(c, d) = \frac{(c, d)}{\|c\| \cdot \|d\|} = \frac{(c, d)}{\sqrt{(c, c) \cdot (d, d)}} \quad (8)$$

现根据定义得到了概念内积和范数,现在证明概念构成的空间满足内积空间的定义.

**定理 1**(概念内积空间). 设概念空间  $U$  是任意概念依据概念内积构成的集合,则概念空间是内积空间.

证明. 需对任意  $x, y \in U$ , 证明概念空间满足:

$$(1) (a \cdot x, y) = a \cdot (x, y);$$

$$(2) (x + z, y) = (x, y) + (z, y);$$

$$(3) (x, y) = \overline{(y, x)};$$

$$(4) (x, x) \geq 0 \text{ 且 } (x, x) = 0 \Leftrightarrow x = 0.$$

现依次证明概念空间  $U$  满足这 4 个条件. 首先设  $x = \langle p_{x1}, p_{x2}, \dots, p_{xu} \rangle$ ,  $y = \langle p_{y1}, p_{y2}, \dots, p_{yv} \rangle$ , 根据定义 1 有

$$\begin{aligned} (a \cdot x, y) &= \sum_{j=1}^v \sum_{i=1}^u Sim(a \cdot p_{xi}, p_{yj}) \\ &= a \cdot \sum_{j=1}^v \sum_{i=1}^u Sim(p_{xi}, p_{yj}) \\ &= a \cdot (x, y), \end{aligned}$$

于是条件(1)成立.

现证明条件(2). 令  $z = \langle p_{z1}, p_{z2}, \dots, p_{zv} \rangle$ , 则有

$$\begin{aligned} (x + z, y) &= \sum_{j=1}^v \left( \sum_{i=1}^u Sim(p_{xi}, p_{yj}) + \sum_{i=1}^w Sim(p_{zi}, p_{yj}) \right) \\ &= \sum_{j=1}^v \sum_{i=1}^u Sim(p_{xi}, p_{yj}) + \sum_{j=1}^v \sum_{i=1}^w Sim(p_{zi}, p_{yj}) \\ &= (x, y) + (z, y), \end{aligned}$$

故条件(2)成立.

根据概念内积定义,条件(3)明显成立.

根据式(4)有  $Sim(p_i, p_j) \geq 0$ , 于是就有  $(x, x) \geq 0$ , 当  $(x, x) = 0$  时,有

$$\begin{aligned} (x, x) = 0 &\Leftrightarrow \sum_{j=1}^v \sum_{i=1}^u Sim(p_{xi}, p_{xj}) = 0 \\ &\Leftrightarrow Sim(p_{xi}, p_{xj}) = 0, \forall i, j \in u. \end{aligned}$$

因为我们知道对于任意义原有:  $Sim(p_i, p_i) = 1$ , 即义原与自己的相似度为 1. 则上式成立的唯一条件为  $u = 0$ , 即有  $x = 0$ .

综上所述命题成立.

证毕.

因为每个词可由 1 个或多个“概念”组成,于是根据概念相似度定义词的相似度如下.

**定义 3**(词相似度). 设词  $w_i = \langle c_{i1}, c_{i2}, \dots, c_{iu} \rangle$ ,  $w_j = \langle c_{j1}, c_{j2}, \dots, c_{jv} \rangle$ , 其中  $c_{j1}, c_{j2}, \dots, c_{jv}, c_{i1}, c_{i2}, \dots, c_{iu} \in U$ , 则词  $w_i, w_j$  的相似度为

$$Sim(w_i, w_j) = \max_{x=1, \dots, u, y=1, \dots, v} Sim(c_{ix}, c_{jy}) \quad (9)$$

在实际加载知网进行计算过程中,增加了一个调节参数  $\beta$ , 在概念范数计算中,用  $\beta$  作为范数平方的初始值,即有  $\|c\| = \sqrt{\beta + (c, c)}$ .  $\beta$  的含义在于对

《知网》用概念表示词义的补充,例如:“黄牛”只有一个概念“livestock|牲畜”,而很多家畜如猪、羊等都有“livestock|牲畜”这个概念,因此纯粹用《知网》定义的概念代表词的向量空间似乎有些不合理.因此,本文增加了一个量 $\beta$ ,表示在概念相同条件下,而词不同的最小距离值.

**例 2.** 根据以上定义,以词“癞皮狗”与“黄牛”为例子,分析相似度计算过程.首先根据《知网》定义,“癞皮狗”有两个概念,即“human|人,undesired|莠”和“livestock|牲畜,undesired|莠”,而“黄牛”有一个概念“livestock|牲畜”.然后,我们依次分析组成概念的义原之间的相似度,得到:人 $\rightarrow$ 牲畜=0.185;牲畜 $\rightarrow$ 牲畜=1;莠 $\rightarrow$ 牲畜=0;人 $\rightarrow$ 莠=0.

于是可以计算概念的内积和范数如下,其中 $\beta$ 取为 1:

$$(\text{“人,莠”, “牲畜”}) = 0.185;$$

$$(\text{“牲畜,莠”, “牲畜”}) = 1;$$

$$\|\text{“人,莠”}\| = \sqrt{3};$$

$$\|\text{“牲畜,莠”}\| = \sqrt{3};$$

$$\|\text{“牲畜”}\| = \sqrt{2}.$$

然后得到概念间的相似度:

$$\text{Sim}(\text{“牲畜,莠”, “牲畜”}) = 1/\sqrt{2 \times 3} \approx 0.408;$$

$$\text{Sim}(\text{“人,莠”, “牲畜”}) = 0.185/\sqrt{2 \times 3} \approx 0.076.$$

最后根据词相似度定义得到“癞皮狗”和“黄牛”的相似度为 0.408.

### 3.1.3 概念内积空间分析

由以上分析可以看到,在概念空间中的内积定义与欧氏空间有很大的不同,这二者有什么联系?另外,定义的相似度公式有什么实际物理意义?这就引出了对概念内积空间的具体分析.

首先,对组成概念的义原做一个假定,即设任意义原 $p$ 为 $n$ 维欧氏空间中的标准向量, ( $n$ 是未知的),于是有 $p = \langle w_{p1}\epsilon_1, w_{p2}\epsilon_2, \dots, w_{pn}\epsilon_n \rangle$ , 且 $\|p\| = 1$ , 其中 $\{\epsilon_1, \epsilon_2, \dots, \epsilon_n\}$ 为 $n$ 维空间规范正交系,  $w_{p1}, w_{p2}, \dots, w_{pn}$ 为 $p$ 分别在 $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ 上的投影值. 义原 $p, q$ 的内积可以表示为

$$(p, q)_E = \sum_{i=1}^n w_{p1} \cdot w_{q1} \quad (10)$$

为区别起见,欧氏空间的内积计算用 $(\cdot)_E$ 表示.

义原 $p, q$ 的相似度公式为

$$\text{Sim}(p, q) = \frac{(p, q)_E}{\|p\| \cdot \|q\|} = (p, q)_E = \sum_{i=1}^n w_{p1} \cdot w_{q1} \quad (11)$$

即与内积公式一致.

同时设概念 $c$ 是由其组成的义原 $\langle p_{c1}, p_{c2}, \dots, p_{cu} \rangle$ 构成的合向量. 于是就有

$$\begin{aligned} c &= \langle p_{c1}, p_{c2}, \dots, p_{cu} \rangle \\ &= \left\langle \sum_{i=1}^u w_{c1} \cdot \epsilon_1, \sum_{i=1}^u w_{c2} \cdot \epsilon_2, \dots, \sum_{i=1}^u w_{cn} \cdot \epsilon_n \right\rangle \end{aligned} \quad (12)$$

**定理 2**(概念内积空间与欧氏空间关系). 概念空间的内积与 $n$ 维欧氏空间对应. 即概念空间的内积计算可以转换为 $n$ 维欧氏空间的内积.

证明. 根据概念内积定义(定义 1), 我们可以将义原的向量表示带入可得

$$\begin{aligned} (c, d) &= \sum_{j=1}^v \sum_{i=1}^u \text{Sim}(p_{ci}, p_{dj}) \\ &= \sum_{j=1}^v \sum_{i=1}^u (p_{ci}, p_{dj}) \\ &= \sum_{j=1}^v \sum_{i=1}^u \sum_{k=1}^n (w_{ci,k} \cdot w_{dj,k}) = \\ &= \sum_{k=1}^n \sum_{j=1}^v \sum_{i=1}^u (w_{ci,k} \cdot w_{dj,k}) = \\ &= \sum_{k=1}^n \left( \sum_{i=1}^u (w_{ci,k}) \cdot \sum_{j=1}^v (w_{dj,k}) \right) \end{aligned} \quad (13)$$

根据式(12), 于是就有

$$(c, d)_E = \sum_{k=1}^n \left( \sum_{i=1}^u (w_{ci,k}) \cdot \sum_{j=1}^v (w_{dj,k}) \right) = (c, d).$$

故命题得证.

证毕.

**定理 3.** 概念相似度公式对应欧氏空间余弦公式, 即有

$$\text{Sim}(c, d) = \frac{(c, d)_E}{\sqrt{(c, c)_E \cdot (d, d)_E}} \quad (14)$$

其中 $c, d \in U$ .

根据定理 2 和概念相似度公式定义可以直接推导出命题, 故略去证明.

由定理 2, 3 可以看到概念内积计算实际上可以转换为一个标准的 $n$ 维欧氏空间的内积计算, 而这里 $n$ 是一个未知量. 同理, 得到的概念相似度的计算公式实际就是概念向量在欧氏空间中夹角的余弦. 这样就很容易理解概念相似度的物理含义了.

### 3.2 文本内积空间和文本相似度计算

首先将所有概念向量做规范化处理, 即令 $c' =$

$\frac{c}{\|c\|}$ , 其中  $c \in U$ , 于是就有  $\|c'\| = 1$ , 同理规范化处理词向量  $w$ , 有  $\|w\| = 1$ .

设所有词张成的空间为  $W$ , 称为文档空间, 设  $|W| = m$ , 则对任意  $s \in W$ , 均可表示为  $\langle s_1 w_1, s_2 w_2, \dots, s_m w_m \rangle$ . 设  $d_i = (t_{i1} w_{i1}, t_{i2} w_{i2}, \dots, t_{iu} w_{iu})$  和  $d_j = (t_{j1} w_{j1}, t_{j2} w_{j2}, \dots, t_{jv} w_{jv})$  为两个文档的向量空间表示, 其中  $(w_{i1}, w_{i2}, \dots, w_{iu}, w_{j1}, w_{j2}, \dots, w_{jv})$  表示组成文档的词向量, 而  $(t_{i1}, t_{i2}, \dots, t_{iu}, t_{j1}, t_{j2}, \dots, t_{jv})$  表示相应词向量的权重, 则明显有文档  $d_i, d_j$  为词空间向量. 与概念内积类似, 定义文档空间内积如下.

**定义 4**(文档空间内积). 设词  $w_i \in W$ , 文档空间向量  $d_i = (t_{i1} w_{i1}, t_{i2} w_{i2}, \dots, t_{iu} w_{iu})$ ,  $d_j = (t_{j1} w_{j1}, t_{j2} w_{j2}, \dots, t_{jv} w_{jv})$  则文档空间向量  $d_i, d_j$  的内积为

$$(d_i, d_j) = \sum_{n=1}^v \sum_{m=1}^u t_{im} \cdot t_{jn} \cdot \text{Sim}(w_{im}, w_{jn}) \quad (15)$$

根据文档空间内积, 可导出文档范数及相似度定义.

**定义 5**(文档范数及文本相似度). 定义文档  $d_i = (t_{i1} w_{i1}, t_{i2} w_{i2}, \dots, t_{iu} w_{iu})$  的范数为

$$\|d_i\| = \sqrt{(d_i, d_i)} = \sqrt{\sum_{n=1}^u \sum_{m=1}^u t_{im} \cdot t_{in} \cdot \text{Sim}(w_{im}, w_{in})} \quad (16)$$

令文档空间向量  $d_i = (t_{i1} w_{i1}, t_{i2} w_{i2}, \dots, t_{iu} w_{iu})$ ,  $d_j = (t_{j1} w_{j1}, t_{j2} w_{j2}, \dots, t_{jv} w_{jv})$  的文本相似度为

$$\text{Sim}(d_i, d_j) = \frac{(d_i, d_j)}{\|d_i\| \cdot \|d_j\|} = \frac{(d_i, d_j)}{\sqrt{(d_i, d_i)} \cdot \sqrt{(d_j, d_j)}} \quad (17)$$

**定理 4**(文档内积空间). 设文档空间为  $W$ , 文档内积满足式(15), 则文档空间是内积空间.

对任意  $x, y \in W$ , 证明文档空间满足:

- (1)  $(a \cdot x, y) = a \cdot (x, y)$ ;
- (2)  $(x + z, y) = (x, y) + (z, y)$ ;
- (3)  $(x, y) = \overline{(y, x)}$ ;
- (4)  $(x, x) \geq 0$  且  $(x, x) = 0 \Leftrightarrow x = 0$ .

证明过程与定理 1 类似. 具体证明略.

**定理 5**(文档内积空间与欧氏空间关系). 文档内积空间的内积与  $n$  维欧氏空间对应. 即文档内积空间的内积计算可以转换为标准  $n$  维欧氏空间的内积.

证明过程同定理 2, 具体证明略.

**定理 6.** 文本相似度公式对应欧氏空间余弦

公式, 即有

$$\text{Sim}(d_i, d_j) = \frac{(d_i, d_j)_E}{\sqrt{(d_i, d_i)_E} \cdot \sqrt{(d_j, d_j)_E}},$$

其中  $d_i, d_j \in W$ .

具体证明略.

由以上推导得到文本相似度公式. 具体实现在算法中, 还需要考虑文本预处理工作, 如切词、得到 TFIDF 权重等等, 文本相似度算法形式化表述如下.

**算法 2.** 文本相似度算法.

输入: 文本  $S_1, S_2$

输出:  $rc$  相似度值

Begin

1. 判断是否已经加载词网模型, 没有则加载到内存;
2. 将  $S_1, S_2$  进行拆词和计算 TFIDF, 得到  $d_i, d_j$ , 即文档向量空间表示;
3. 通过式(15)计算  $d_i, d_j$  内积空间;
4. 用式(16)分别计算  $d_i, d_j$  的范数;
5. 用式(17)计算  $rc = \text{Sim}(d_i, d_j)$ ;
6. 返回  $rc$ ;

End.

其中在计算词内积和范数时, 需要调用词相关度公式(式(9))和概念内积公式(式(6))进行计算.

根据定理 6, 同概念相似度一样, 得到的文本相似度的计算公式同样可以理解为文档向量在一个特定欧氏空间中夹角的余弦.

本节建立了一个基于语义的内积空间模型, 通过该模型成功地引入了文本语义的相似信息. 同时利用内积空间的基本性质推导出文本、词和概念的相似度计算方法. 不同于普通的相似度计算方法, 该模型由于通过严格证明满足内积空间的定义, 因此可以利用内积空间的各种性质进一步推导文本在此空间中的特性. 另外, 由于内积空间是欧式空间的概化, 因此在内积空间中也可以引入很多欧式空间的性质.

## 4 TCIS 算法

基于提出的语义内积空间模型, 本节建议一种新的文本聚类算法——TCIS. 传统的自顶向下或者自底向上的聚类算法导致一个层次的聚类结果, 而局部搜索方法, 如  $k$ -means 算法等, 往往是平的聚类结果. 这些方法要么缺少局部特征, 要么聚类形状趋于规则球形, 往往在聚类效果上不佳. TCIS 提出采

用一种两阶段聚类算法<sup>[13]</sup>,从上至下的分裂算法和从下至上合并算法,完成聚类的过程,既满足了局部搜索特征,又可以适用于任意的聚类形状.同时,因为内积空间的计算方法的特点,不能直接采用欧式空间的坐标计算,因此在每阶段的分裂时采用划分的方法,完成阶段聚类.

另外,与传统方法不同,TCIS 主要利用了语义内积空间度量任意文本之间的相似度.算法的主要内容包括:首先,给出了一种计算文本代表点的方法;其次,给出一种分裂算法,将待聚类的文本集合分裂为多个子集;然后给出一种算法合并文本集合;最后,TCIS 算法结合刚才提到的从上至下的分裂算法和从下至上合并算法完成聚类的过程,具体内容如下.

4.1 代表点计算

代表矢量计算目的是在给定文档集合条件下,确定文档集合的中心点.利用向量空间模型,采用欧氏空间的质点的计算方法可以方便求出文档的中心点作为文档集合的代表矢量,但这种方法显然没有考虑词之间的相似度.在语义内积空间模型下,由于对应的正交欧氏空间无法直接求取,本文提出了一种近似逼近中心点的代表矢量计算方法,算法形式化如下.

算法 3. 代表点计算.

输入:文档集合的向量空间表示  $S$

输出: $S$  的代表点

Begin

- 1. 初始化向量  $P$ ;
- 2. 对文档集合  $S$  中的每个文档  $s$  做以下两步操作:
- 3. 利用词内积空间中的词相似度计算方法,投影  $s$  中每个词向量  $w$  到  $P$  中的每个词向量  $p$ ,即
$$|p| = |p| + \sum_{w \in s} (\text{Cos}(p, w) \cdot |w|) = \sum_{w \in s} (\text{Sim}(p, w) \cdot |w|);$$
- 4. 如果  $P$  中不包括  $w$ ,就将  $w$  加入到  $P$ ,并计算  $s$  中其他词向量与它的投影;
- 5. 将  $P$  中向量按照降序排列;
- 6. 返回  $P$  中头  $n$  个向量,作为  $S$  的代表矢量,其中  $n$  由用户指定;

End.

从算法 3 可以看到,本文建议了一种简化的方法来计算文档集合的代表点.其中  $n$  表示代表向量的最大维度.

4.2 文档分裂算法

算法的目的是将整个文档集合分裂为多个小的集合,每个集合内部相似,而集合间相异.同样由于

不能在语义内积空间模型下,直接求取正交欧氏空间,那么如何在语义内积空间中实现文档的分裂?本文的思想是每一步将文档分裂为两个小的文档,这两个文档集应该是具有最大的相异度.具体分裂的思路是:首先,算法随机选择聚类的代表点,第 2 步将文档集合中的所有元素依次分配到相应的类别中,分类的依据是它们与代表矢量的相似度.第 3 步是重新计算每个聚类的中心点(根据算法 2);重复执行第 2,3 步,直到达到最大的迭代次数或者聚类不再变化为止.算法形式化如下.

算法 4. 文档分裂算法.

输入:文档集合的向量空间表示  $S$

输出:输出文档集合  $S_1$  和  $S_2$

Begin

- 1. 从  $S$  中随机选择两个文档向量  $s_1$  和  $s_2$  做为两个聚类  $S_1$  和  $S_2$  的代表点;
  - 2. 重复以下步骤直到  $changeNum=0$  或者  $IterNum=maximum$ ;
  - 3.  $changeNum \leftarrow 0$ ;  $IterNum \leftarrow IterNum+1$ ;
  - 4. 对  $S$  中每个文档  $s$  做如下操作:
  - 5. 根据  $s$  与  $s_1$  和  $s_2$  的相似度,将  $s$  分类到  $S_1$  或者  $S_2$  中;
  - 6. 如果  $s$  的类别发生改变,则  $changeNum \leftarrow changeNum+1$ ;
  - 7. 重新计算  $S_1$  和  $S_2$  的代表点,并替换到  $s_1$  和  $s_2$ ;
  - 8. 返回文档集合  $S_1$ ,  $S_2$  及对应的代表矢量;
- End.

本文采用了式(17)作为文档分裂算法中的第 5 行的相似度计算.算法第 7 行的代表点计算采用了以上提到的代表点计算方法.通过算法 4 本文实现了任意基于向量空间模型的文本集合分裂为两个子集的目的.

4.3 聚类合并算法

本节实现了一种组合相似聚类的方法.主要思想比较简单,就是利用聚类的代表点,计算代表点之间的相似度,如果相似度大于指定的阈值,则合并两个聚类.重复处理文档聚类,直到无法再合并为止.算法的形式化描述如下.

算法 5. 聚类合并算法.

输入:所有文档聚类集合  $S_1, S_2, \dots, S_x$  和对应的代表点  $s_1, s_2, \dots, s_x$  合并阈值  $h$ .

输出:合并后的文档聚类集合  $S_1, S_2, \dots, S_y$

Begin

- 1. 依次对每个文档集合  $S_i$  的代表点  $s$  做如下操作:
- 2. 计算  $s$  与其它聚类集合  $S'$  的代表点  $s'$  的相似度;
- 3. 如果  $\text{Sim}(s, s') > h$ ,则合并聚类  $(S, S')$ ;

4. 返回合并后的聚类结果；  
End.

其中  $Sim(s,s')$  表示计算  $s$  和  $s'$  的相似度,采用算法 2 计算.

#### 4.4 TCIS 算法

在前面小节描述的基础上,本节讨论 TCIS 算法的实现. TCIS 算法的总的流程包括两个阶段:即从上至下的“分裂”阶段和从下至上的“合并”阶段. 在“分裂”阶段,TCIS 重复调用 4.2 节描述的分裂算法,每次迭代 TCIS 从聚类中选择质量最差的聚类,通过调用算法 4 将其分裂为 2 个较小的聚类,直到每个聚类的质量达到一定阈值为止. 其中聚类质量用  $Density(S)$  表示,通过以下公式定义.

$$Density(S) = \sqrt{(\sum_{s \in S} Sim^2(s,s'))/|S|} \quad (18)$$

其中  $s$  表示聚类  $S$  的代表点.

在“合并”阶段,TCIS 调用 4.3 节的合并算法,输入为第一阶段通过分裂得到的聚类文档集合. 在合并阶段结束后,TCIS 可以输出最终的聚类结果和聚类对应的代表点集合. 聚类代表点的意义非常重要,通过代表点,用户可以发现该聚类的基本特点并且知道聚类结果的含义.

算法的形式化描述如下.

**算法 6.** 聚类主框架.  
输入: 待聚类文档集  $t$ , 合并阈值  $h$ , 分裂阈值  $k$ .  
输出: 聚类结果  
Begin  
1. 先将文档  $t$  转换为向量空间表示  $S$ , 包括分词及  $TF/IDF$  参数计算等等;  
2. 初始化设置聚类  $Se=\{S\}$ ;  
3. 依次计算聚类  $Se$  中每个子类的质量(式(18));  
4. 寻找质量最差的聚类  $S$ ;  
5. 如果  $Density(S)<k$ , 则  $split(S)$ , 同时将分裂结果在  $Se$  中替换  $S$ ;  
6. 重复步 3~5, 直到每个聚类的密度均大于等于  $k$  为止;  
7. 根据参数  $h$ , 调用聚类合并算法, 即算法 5;  
8. 返回最后聚类结果  $Se$ ;  
End.

其中算法第 5 行,  $split(S)$  表示调用算法 4, 执行分裂过程, 将聚类分裂为两个集合. 第 7 行表示通过算法 5 的调用合并聚类, 然后将结果返回给  $Se$ . 通过两阶段处理, TCIS 就完成了全部的聚类过程, 下一节将通过实验验证模型的合理性和算法聚类质量.

## 5 实 验

根据以上讨论, 本文实现了 TCIS 算法. 具体实验的硬件环境为 PIV 1.7GHz, 1024MB 内存, 80GB 硬盘, 开发工具为 DELPHI7.0, 数据库使用的是微软的 MS SQL Server 2000 数据库. 系统具体采用的缺省参数如表 2 所示.

表 1 系统缺省参数

	名称	值	说明
1	$y$	2.0	距离初始阈值
2	$x$	0.1	概念每层递减参数
3	$\alpha$	1	
4	$\beta$	1	最小距离权重

**实验 1.** 聚类效果分析. 实验中抽取了 420 条来自中文网站 [www.sina.com.cn](http://www.sina.com.cn) 和 [www.tom.com](http://www.tom.com) 在 6 月份的新闻. 通过聚类处理, 结果如表 2 所示, 其中分类阈值和合并阈值分别设置为 0.45 和 0.40, 总处理时间为 3.9s.

表 2 文本聚类结果

	数量	质量	代表词
1	35	0.52	失业率 香港 长官 花费 特区
2	46	0.64	航空界 航校 基业 新生代 户外
3	7	0.48	给钱 敲诈 觊觎 西班牙人 颠覆
4	45	0.5	义利观 多哥队 商务部 瑞士队 内地
5	29	0.85	卖场 招财进宝 创意 趣 求雨
6	27	0.83	失踪 大兵 被控 美军 动用
7	24	0.83	西门子 诺基亚 合并 电信 获益
8	43	0.73	镶嵌 周末财 宝石 珠宝 框架
9	11	0.61	力作 笔记本 寻址 展会 逻辑
10	31	0.81	榜 产 人权 动态 欧盟
11	46	0.8	演习 美军 关岛 提前 航母
12	76	0.48	美国 球迷 扁 警告 高才生 公明党
	420	0.659	

从表 2 可以看到, 聚类结果分为 12 个类, 每个类用 5 个代表矢量表示, 平均聚类的质量 0.659, 其中平均聚类质量采用加权平均的方法计算, 公式如下

$$Density_{avg}(C) = (\sum_{c \in C} density(c) \cdot quantity(c)) / |C|,$$

其中  $c$  表示  $C$  中的任意元素. 由聚类结果可以发现聚类平均质量较高, 同时可以发现聚类代表点具有明显的意义, 如第 1 个聚类的代表点为“失业率 香港 长官 花费 特区”, 由此, 人们可以估计该聚类主要包括了香港特区的就业等相关问题的新闻.

**实验 2.** 聚类质量比较. 本实验比较不同算法间的聚类效果, 数据集与上一个实验相同. 实验主要



比较 TCIS 算法和  $k$ -Means 算法( $k$  取值为 12)和 TCIS 算法在不考虑语义相似度条件下( $n$ -TCIS)聚类质量的差别,实验结果如表 3 所示. 其中每个聚类质量公式采用式(18),总体聚类质量表示为平均质量和聚类数量的比值,具体定义如下: $Quality(C) = 100 \cdot Density_{avg} / |C|$ . 图 1 表示三个算法在不同文档数量条件下耗费时间的比较.

表 3 不同算法聚类质量比较				
	聚类数	平均质量/%	耗时	总体质量
TCIS	12	65.9	3.9	5.49
$n$ -TCIS	89	69.4	1.1	0.78
$k$ -Means	12	22.3	2.3	1.86

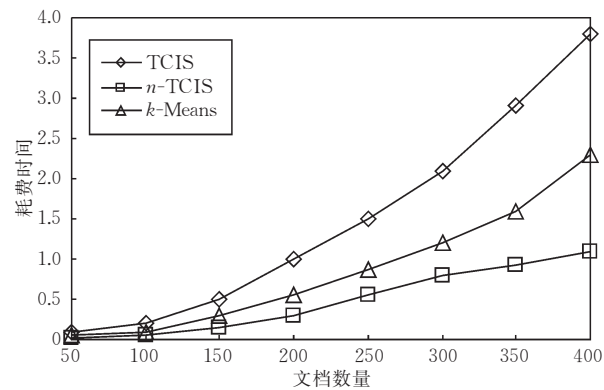


图 1 算法耗费时间

从以上实验结果可以看到,TCIS 算法的总体聚类质量明显好于其他算法,分别是  $k$ -Means 和  $n$ -TCIS 算法的 2.95 和 7 倍;但是从耗费时间上看,TCIS 算法明显高于其他两种算法,其主要原因在于 TCIS 算法基于语义内积空间模型,增加了文本相似度计算的工作. 同时根据计算性能,在短文本数据条件下,TCIS 算法的处理性能为 105~500 条/s,每小时处理数据量为 37.8 万~180 万,同时通过提高硬件性能和采用并行处理方法,可以很容易将算法的处理能力提高到 G 级甚至更高.

6 结 论

本文提出了一种新的文本聚类算法-TCIS. 算法基于知网的概念结构,提出了中文语义的内积空间模型,利用该模型得到了任意概念、词或者文本之间的相似度计算公式. 在语义内积空间模型基础上,本文提出了采用两阶段工作方式,即向下分裂和向上聚合,完成文本聚类. 算法成功用于了面向中文环境的短文本聚类问题,实验表明相对于传统方法,该方法的聚类质量更高.

下一步的工作包括:  
(1)在保证算法精度的条件下,进一步提高算法的执行效率.  
(2)结合启发式计算方法,研究文档关联规则挖掘等问题.  
(3)将算法应用到具体 Web 数据挖掘任务中,如热点发现、舆情分析等等.

致 谢 作者要向《知网》的发明人董振东先生和董强先生表示感谢,他们的工作是本文工作的基础. 另外本文在文本预处理中采用了海量公司智能分词研究版的产品,这里一并表示感谢!

参 考 文 献

[1] Pelleg D, Moore A. X-means: Extending K-means with efficient estimation of the number of clusters//Proceedings of the 17th International Conference on Machine Learning (ICML). Palo Alto, 2000: 727-734

[2] Hamerly G, Elkan C. Learning the  $k$  in  $k$ -means//Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS). 2003: 281-289

[3] Han Jia-Wei, Kamber M. Data Mining: Concepts and Techniques (2nd Edition). San Francisco: Morgan Kaufmann Publishers, 2006

[4] Corley C, Mihalcea R. Measuring the semantic similarity of texts//Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment. Ann Arbor, 2005: 13-18

[5] Possas B, Ziviani N, Meira W, Ribeiro-Neto B. Set-based vector model: An efficient approach for correlation-based ranking. ACM Transactions on Information Systems, 2005, 23(4): 397-429

[6] Zhang Z, Otterbacher J, Radev D. Learning cross-document structural relationships using boosting//Proceedings of the 12th International Conference on Information and Knowledge Management. New Orleans, 2003: 124-130

[7] Hammouda K M, Kamel M S. Efficient phrase-based document indexing for Web document clustering. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(10): 1279-1296

[8] Dolan W B, Quirk C, Brockett C. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources//Proceedings of the 20th International Conference on Computational Linguistics. Geneva, Switzerland, 2004: 350-356

[9] Beil F, Ester M, Xu Xiao-Wei. Frequent term-based text clustering//Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Edmonton, Alberta, Canada, 2002: 436-442

[10] Liu Qun, Li Su-Jian. Word similarity computing based on How-Net. Computational Linguistics and Chinese Language Processing, 2002, 7(2): 59-76

[11] Bu Dong-Bo, Bai Shuo, Li Guo-Jie. The Duplex strategy of term weighting in text clustering. Journal of Software, 2002, 13(11): 2083-2090(in Chinese)  
(卜东波,白硕,李国杰.文本聚类中权重计算的对偶性策略.软件学报,2002,13(11):2083-2090)

[12] Zhao Jun, Jin Qian-Li, Xu Bo. Semantic computation for

text retrieval. Chinese Journal of Computers, 2005, 28(12): 2068-2078(in Chinese)  
(赵军,金千里,徐波.面向文本检索的语义计算.计算机学报,2005,28(12):2068-2078)

[13] Cheng David, Vempala Santosh, Kannan Ravi, Wang Grant. A divide-and-merge methodology for clustering//Proceedings of the 24th ACM Symposium on Principles of Database Systems. New York: ACM Press, 2005: 196-205



**PENG Jing**, born in 1973, Ph. D. . His research interests include data mining and natural language processing.

**YANG Dong-Qing**, born in 1945, professor, Ph. D. supervisor. Her main research interests include database and

information theory.

**TANG Shi-Wei**, born in 1939, professor, Ph. D. supervisor. His main research interests include database and information theory.

**FU Yan**, born in 1980, Ph. D. candidate. Her main research interests include database and data mining.

**JIANG Han-Kui**, born in 1966, master, senior engineer. His research interests include software engineering and information system.

Background

This research was supported by the National Natural Science Foundation of China under grant Nos. 60473051, 60503037, the China Postdoctoral Science Foundation under grant No. 20060400002, the Sichuan Youth Science and Technology Foundation of China under grant No. 2007Q14-055, the National High-tech Research and Development of China under grant No. 2006AA01Z230 and the Natural Science Foundation of Beijing under grant No. 4062018.

In Web pages, there have many very short documents such as news title, abstract and annotation, etc. Recently, there has been increasing interest in data clustering of short document go with the development of Web technical and applications. Differences from traditional dataset, short document have very high dimensions and sparse data spaces.

Before using traditional method to cluster the documents, we must convert document to Vector Space Model-VSM or suffix-tree model at first. Because of the attributes of short document (high dimensions and spare data spaces),

the relationship such as similarity between documents is very low in a great many conditions. However, neither Vector Space Model nor suffix-tree model does not consider the relationship between words, so the distance of similarity which computed by the traditional method doesn't match the practical conditions.

This paper proposes a novel clustering algorithm of short document based on concept similarity in Chinese text processing. The paper creates similarity method among Chinese concepts, words and text based on the definition of inner space at first, and then analyzes systematically the algorithm in theory. Through a two phrase processes, i. e. top-down "divide" phase and a bottom-up "merge" phase, it finishes the clustering of text data. The method has been applied into the data clustering of Chinese short documents. Extensive experiments show that the method is better than traditional algorithms.