一种新的演化文本流聚类算法*)

邓维维　彭　宏
(华南理工大学计算机学院　广州 510641)

摘　要　数据流的聚类作为聚类的一个分支,已经成为了数据挖掘的研究热点。虽然已经有不少数据流算法出现,但是大部分都是针对低维的数值型数据,很少有高维文本流的研究。本文在传统的数据流聚类框架基础上,提出了一种新的文本微聚类结构体,它更适合文本聚类,同时还将在线微聚类分为潜在微聚类和异常微聚类,提高了对孤立点的适应能力。实验表明该算法相对于其他文本流聚类算法更有效。
关键词　聚类,数据流,文本流

An Algorithm for Clustering Evolving Text Data Stream with Outliers

DENG Wei-Wei　PENG Hong
(Computer Science Department, South China University of China, Guangzhou 510641)

Abstract　As a branch of clustering, data stream clustering has become a hot spot in data mining. Although there are many stream clustering algorithms, they are only suitable for low dimensional numeric data type, and few of them are designed for high dimensional text streams. A novel online micro cluster structure based on the traditional stream clustering framework was proposed and it is suitable for clustering text. Dividing the online micro cluster into potential and outlier micro clusters also brings advantage when outliers appear frequently in stream. Experiments show that these methods bring advancements for processing text streams when compared to others.
Keywords　Clustering, Data stream, Text stream

1　引言

近几年,数据流开始成为计算机某些领域的一个研究热点,比如数据库系统,数据挖掘和分布式系统。数据流可以简单看成有序的数据点序列,随着时间的流逝,不断有新的数据涌来。网络上传递的数据包、Web点击流、电信的通话记录、股票数据和感应网络采集的数据都可以看作是数据流。数据流处理方法具有一些传统数据处理方法没有的特点:

一遍扫描:在满足处理要求的情况下,要尽可能少地扫描数据集,最好是一遍扫描;

有限的内存及存贮空间:由于数据流具有无限连续性,不可能存贮如此海量的数据,因而要对数据流进行概化,构造概要结构(synopsis)或有选择地舍弃;

实时性:每一个记录的处理时间要尽可能的短,要能够跟上流的速度。

数据流聚类问题可以定义为:对给定的数据元素构成序列$d_1,d_2,\cdots,d_i,\cdots,d_n$,将它们分成多个组$\{C_1,C_2,C_3,\cdots,C_k\}$,使得组内元素有尽可能地相似,组间的元素尽可能地相异。不同的数据类型元素一般采用不同的相似度度量方法,距离是比较常见的度量方法。流聚类算法分为基于划分、基于密度[10]和基于网格[11]等几类,但是以基于划分的居多。现在已经有不少基于划分的数据流聚类方法[1~6],但这些方法都只适合处理数值型数据,文[7]对二值型数据流聚类方法进行了研究,它利用了二值型数据的特点,简化了稀疏二值矩阵的计算,提高了性能和聚类质量,但它仍不适合文本这类高维数据;文[8]用投影的方式研究了高维数据流聚类的问题,但它也只是在高维的稀疏空间里面寻找某些低维空间,使得在这些低维空间能够构成有意义的聚类。文[9]提出了一种可以对文本和标称型数据流进行聚类方法,它采用和文[4]中类似的方法,没有考虑存在孤立点的情形和文本之间相似度度量的特点。本文中的方法充分考虑了流文本中存在孤立点的问题,专门设计了异常微聚类,用来处理文本流中孤立点过多时聚类质量下降的问题。设计了新的微聚类结构,在流环境下维护了文本相似计算中的IDF(Inverse Document Frequency),使得文本聚类效果更好。实验表明,该新方法在聚类效果上优于文[9]中提出的方法,特别是在文本流中出现孤立点时。

2 离线和在线问题

由于数据流中的数据不能被再次访问,数据流聚类算法一般要在有限的存储空间里维护已经流逝的数据的概要信息(synopsis)或者说浓缩信息(condensed information)。比如说,文[2]实现了一个流环境下的K-means算法,它维护k个聚类,当新的元素到来时,根据需要作聚类合并或者是修改当前某个聚类的特征。K-means的聚类结果和元素到达的次序相关,同时文[2]中提到的方法只考虑了将元素加入到某个Cluster,或者做Cluster合并,如果事后发现合并是有害于聚类效果时,也无法回头。文[4]提出了在线和离线的处理方式。该方法将聚类过程分为在线微聚类生成部分和离线宏聚类查询部分。宏聚类可以看作是我们传统意义上的聚类,而微聚类是比宏聚类更小的聚类,它的个数一般比宏聚类的个数多很多。当用户查询当前宏聚类的时候,可以用某些聚类算法,比如K- means或者HAC等传统聚类方法,从当前的微聚类中生成宏聚类。为了方便用户查询历史上某个时间段的聚类,每隔一定的时间,微聚类部分将自己的快照保存到磁盘,宏聚类部分从这些微聚类中选择出落在相应时间段的微聚类来产生结果。考虑到随着时间的流逝,磁盘上的快照会越来越多,有可能导致磁盘空间消耗完。文[4]中提到了一种时间金子塔模型来进行快照的选择,即哪些快照留下来,哪些

*)基金项目:国家自然科学基金(60574078);广东省自然科学基金(31454)。**邓维维**　博士研究生,主要研究方向:移动计算,数据流挖掘;**彭宏**　博士生导师,博士后,主要研究方向:数据仓库、数据挖掘。

从磁盘删除。考虑到历史快照重要性更低，它按照一定方式将历史快照删除，越久的数据被删除得越多。该方法能够保证磁盘消耗相对流数据来说是次线性的 $O(\log(n))$，其中 n 为已经到达的元素个数。很多数据流聚类算法都使用了类似的处理框架，比如文[10,11]。本文中的方法也采用类似的框架。

3 演化问题

很多时候，我们只关心数据流中近期数据的聚类情况，而不关心整个数据流的聚类情况。当数据流具有演化特点的时候，即聚类个数，聚类的中心随着时间变化而变化，我们需要考虑如何处理它。很容易想到的就是降低历史数据的重要性。我们可以给每个数据点一个与时间相关的重要性系数，它随时间的增加而减小。比较常用的是指数衰减函数。假设数据点的重要性的半衰期为 t_0，即 $f(t_0)=\frac{1}{2}f(0)$，定义衰减系数 $\lambda=1/t_0$，衰减函数可以写为 $f(t)=2^{-\lambda}$ 通过改变 λ，我们可以改变历史数据的重要性。λ 越大，则历史数据的重要性越小。

4 在线结构

4.1 文档的表示及相似度定义

文档一般用向量空间模型(VSM)来表示，文档 d 可以表示为向量$(w_1,w_2,\cdots,w_n)$，w_i 为文档中第 i 个词的权重，n 为文档 d 中所有词的总数。比较经典也是最常用的权重计算方式是 TF-IDF。其中 TF(Term Frequency)是指词在文本中的绝对频度，用 $tf(t,d)$ 表示。反比文档频率 IDF(Inverse Document Frequency)，定义为 $IDF(t)=\log(N/n_t)$。其中 N 为文档的总数目，n_t 为词 t 出现过的不同文档的数目。某个词在文档 d 中的权重 $w_i(d)=tf(t,d)*idf(t)$。文档表示成向量以后，文档间的语义距离或者语义相似度就可以通过空间中这两个向量的关系来度量。最常用的文档相似性的度量方式是余弦相似度。假定 $\vec{X}$ 和 $\vec{Y}$ 为两个文档向量，其余弦相似度计算如下：

$$\mathrm{Cos}(\vec{X},\vec{Y})=\frac{\vec{X}\cdot\vec{Y}}{\|\vec{X}\|\,\|\vec{Y}\|}=\frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2\cdot\sum_i y_i^2}} \tag{1}$$

4.2 在线结构及微聚类

本算法的在线结构包含以下几个部分：

1)当前所有微聚类中的词构成的词库 WL，该词库为一个 Hash 表，表中维护当前所有的词以及含有该词的文档计数$\{(t_1,tc_1),(t_2,tc_2),\cdots,(t_n,tc_n)\}$，$tc_i$ 为包含词 t_i 的文档个数。

2)当前微聚类中所有文档的个数 D_{total}，若现在总共有 k 个微聚类 $C_1,C_2,\cdots,C_k$，则 $D_{total}=\sum_{j=1}^{k}m_j$，其中 m_j 为 C_j 中的文档个数。

3)文档集合 C 在某个时刻 t_l(最后一个元素加入该微聚类的时间)构成微聚类 $P(t_l,C)$，它由向量$(CF3,\overrightarrow{CF2},\overrightarrow{CF1},m,w(t_l),t_l)$表示。

- $CF3$ 为微聚类中出现的特征词构成的序列$(t_1,t_2,t_3,\cdots,t_n)$；
- $\overrightarrow{CF2}$为微聚类中出现特征词 t_i 的文档计数$(dc_1,dc_2,dc_3,\cdots,dc_n)$，$dc_i$ 是对应的词 t_i 的文档计数，因为很多词都只出现在一个文档中，所以一个比较省空间的办法就是该向量只记录文档计数大于 1 的词。未在$\overrightarrow{CF2}$中出现而在 $CF3$ 中出现的词的文档计数都是 1；
- $\overrightarrow{CF1}$为微聚类中各个特征词 t_l 在时间的权重向量$(w_1,w_2,w_3,\cdots,w_n)$，w_i 是对应的词 t_i 的权重，这里的 w_i 只维护聚类中的词频信息，即 TF；
- m 为微聚类包含的文档的个数；
- $w(t_l)$为$\overrightarrow{CF1}$中所有权重的和，$w(t_l)=\sum_{i=1}^{n}w_i$；
- t_l 为最后一个文档加入到该微聚类的时间；

由半衰函数可知，在时间 $t(t>t_l)$，

$P(t,C)=P(t_l+(t-t_l),C)$

$=(CF3,\overrightarrow{CF2},\overrightarrow{CF1}*2^{-\lambda(t-t_l)},n,w(t_l)*2^{-\lambda(t-t_l)},t)$

所以可以通过 $P(t_l,C)$很快得到 $P(t,C)$。

微聚类分为两种。一种是潜在微聚类(potential-micro-cluster)PC，离线宏聚类部分只使用这些微聚类进行宏聚类的生成，这样减少了孤立点对聚类效果的影响；另一种是异常微聚类(outlier-micro-cluster)OC，该微聚类存放的是孤立点或者新生的更小的微聚类，当它的文档个数大于某个阈值 τ 时，则升级为潜在微聚类。

4.3 微聚类合并运算

令 $P(t_{l1},C_1)=(CF3_1,\overrightarrow{CF2}_1,\overrightarrow{CF1}_1,n_2,w(t_l)_1,t_{l1})$；$P(t_{l2},C_2)=(CF3_2,\overrightarrow{CF2}_2,\overrightarrow{CF1}_2,n_2,w(t_l)_2,t_{l2})$。其中 $t_{l1}\geqslant t_{l2}$，则合并两个微聚类 C_1,C_2 得到：

$$P(t_{l3},C_3)=P(t_{l3},C_1\cup C_2)=(CF3_3,\overrightarrow{CF2}_3,\overrightarrow{CF1}_3,n_3,w(t_l)_3,t_{l3})$$

其中 $t_{l3}=\max(t_{l1},t_{l2})=t_{l1}$。

设 $CF3_1$ 中的词构成的集合为 $tc_1=\{t_{11},t_{21},t_{31},\cdots,t_{n1}\}$，$CF3_2$ 中的词构成的集合为 $tc_2=\{t_{12},t_{22},t_{32},\cdots,t_{n2}\}$，$tc_3=tc_1\cup tc_2=\{t_{11},t_{21},t_{31},\cdots,t_{n1},t_{x_1 2},t_{x_2 2},\cdots,t_{x_r 2}\}$，其中 $t_{x_1 2},t_{x_2 2},\cdots,t_{x_r 2}$为在 C_2 中出现，而未在 C_1 中出现的词。则

$$CF3_3=(t_{11},t_{21},t_{31},\cdots,t_{n1},t_{x_1 2},t_{x_2 2},\cdots,t_{x_r 2})$$

$$\overrightarrow{CF3}_2=(dc_{11}+dc_{s_{12}(t_{11})2},dc_{12}+dc_{s_{12}(t_{12})2},\cdots,dc_{1n}+dc_{s_{12}(t_{1n})n},dc_{x_1 2},dc_{x_2 2},\cdots,dc_{x_r 2})$$

其中 $s_{12}(t)$为一个查询函数，用来查找在 C_1 中的某个词 t 在 C_2 的$\overrightarrow{CF3}_2$中的序号(下标)，若不存在，则返回 0。另外定义 $dc_{02}=0$。令 $\delta=2^{-\lambda(t_{l1}-t_{l2})}$，则

$$\overrightarrow{CF3}_1=(w_{11}+\delta\cdot w_{s_{12}(t_{11})2},w_{12}+\delta\cdot w_{s_{12}(t_{12})2},\cdots w_{1n}+\delta\cdot w_{s_{12}(t_{1n})n},\delta\cdot w_{x_1 2},\delta\cdot w_{x_2 2},\cdots,\delta\cdot w_{x_r 2})$$

C_3 中的文档计数 $m_3=m_1+m_2$。

4.4 微聚类的相似度计算

设某个微聚类 C 中第 i 个词的 TF-IDF 权重为 $x_i(C)=tf(t,C)*idf(t)$；其中 $idf(t)=\log(N/n_t)$，N 来自文档总数的计数器 D_{total}，n_t 来自当前词库；$tf(t,C)$由$\overrightarrow{CF1}$得来。得到 $x_i(C)$后，可以利用公式(1)计算两个微聚类的余弦夹角值。

5 在线结构的维护算法

```
当一个新的文档 d_new 到来时，计算与它最近的潜在微聚类 C_q 的相似度 α；
if(α>ε_p)//ε_p∈(0,1)
{
    将 d_new 加入到 C_q；
}else{
    计算 d_new 与它最近的孤立微聚类 OC_u 的相似度 β；
    if(β>ε_o){
        将 d_new 加入 OC_u；
        if(OC_u 的文档个数 m>τ)
            将 OC_u 从异常微聚类转成潜在微聚类；
}else{
    if(有剩余内存空间){
      则将 d_new 作为一个新的异常微聚类 OC_new 加入到 OC 集合中。
    }else{
            从潜在微聚类中查找最相似的两个微聚类 C_1,C_2；
            if(similarity(C_1,C_2)>ε_p)
            {
              对 C_1,C_2 进行合并操作，释放空间。
            }else{
```

C_{lazy}=findInactive();
freeCluster(C_{lazy});
}
将 OC_{new} 加入到 OC 集合中。
}
}
}
更新 WL;
$D_{total}=D_{total}+1$;

• 找到当前的非活动微聚类。

findInactive()
{扫描所有 PC 和 OC，返回 t_l 最小的那个。}

• 释放微聚类所占的空间

freeMicroCluster(C)
{ 设 $C=(CF3,\overrightarrow{CF2},\overrightarrow{CF1},m,w(t_l),t_l)$
从 WL 减去 $\overrightarrow{CF2}$ 中每个词的对应的文档计数，计数等于 0 时，释放 WL 中该词占用的空间；
清除 C 占用的空间；
$D_{total}-=m$;
}

6 实验及结果分析

6.1 试验数据

实验采用与文[9]中类似的方法。文[9]中关于文本的试验采用了 Yahoo 1996 年的数据。Yahoo 是按照概念层次结构组织它的内容的，因此，文[9]采用深度优先搜索的方式扫描该层次结构，可以依次得到具有相同概念或者说 Yahoo label 的文本。该方法在某个概念层次共产生了包含 251 个不同分类，共 15 万份文本的文本集。由于 yahoo 的数据难以再获得，实验选择 newsgroup 20 的数据，该数据总共有 20 个新闻组，每个新闻组 1000 份文档，总共 2 万份文档。

6.2 试验方法

我们将这 20000 份文档按照扫描次序设计了 4 个文本流，其中前 3 个文本流按照文[9]中方法设计。

Text(S)：文档在流中出现的次序按照文档的类别，也就是说这 20 个新闻组的文档在文档流中依次出现。

Text(R)：对这 20000 份文档进行随机抽取构成文档流。

Text(E)：该数据流由 Text(S)按照以下方法得来：相邻的 2 个新闻组的文档进行随机抽取。设 Text(S)中的 20 个文档块分别为 $S1,S2,S3,\cdots,S_{20}$，每个文档块中有 1000 份文档，则 Text(E)中的文档块 $E_1,E_2,\cdots,E_x,\cdots,E_{20}$ 满足：

$E_x=subseq(rand(E'_x+S_{x+1}),0,999)$;

其中 $E'_x=subseq(rand(E'_{x-1}+S_x),1000,1999)$；$E'_1=S_1$。$E$ 的下标 x 为文档块的编号；+表示将两个文本序列连接起来；$Rand(S)$ 表示对 S 做随机处理，类似 Text(R)的生成，$subseq(S,x,y)$ 表示取文本序列 S 中子序列，编号从 x 到 y。变换后的 Text(E)表现为一个演化的文本流。

Text(O)：该文本流包含孤立点。用 Text(E)的生成方式组构造 Text2(E)，不过只使用 18 个新闻组，然后对剩下 2 个新闻组中做如下混合和插入操作：

1)依次从新闻组 1 中选出 1 份文档，对该文档中的每个单词，随机选择新闻组 2 中的 1 份文档，把单词加在文档后面。

2)对新闻组 2 中的每 1 份文档，进行拆分操作，将 1 份文档拆分为两份。设文档有 n 个单词，随机从中抽出 $n/2$ 个构成 1 份文档，剩下的构成另 1 份文档。

3)将生成的新文档随机选择一个位置插入到 Text2(E)。

完成以上操作后，Text2(E)包含了 10%的孤立点，这时的 Text2(E)即第 4 个测试文本流 Text(O)。

6.3 实验结果

本文采用与文[9]中类似的评估函数-平均聚类纯度来衡量聚类的质量。平均聚类纯度定义如下：

$$purity=(\sum_{i=1}^{k}\frac{|C_i^d|}{|C_i|})/k\times 100\%,$$

k 表示聚类的个数，$|C_i^d|$ 表示 C_i 中优势分类的元素个数，$|C_i|$ 表示 C_i 中元素的个数。

每 100 个数据我们执行一次宏查询，对每连续 10 次宏查询的聚类纯度作一次平均，并生成一个宏聚类生成编号。总共产生了 20 个宏聚类生成编号。以下为与文[9]中算法 CMTC 进行的比较。实验中 ε_p 取 0.40，ε_o 取 0.40，流速度 v 取 100 文档/秒，λ 取 0.15，τ 取 10。Text(S)的比较结果如图 1 所示。

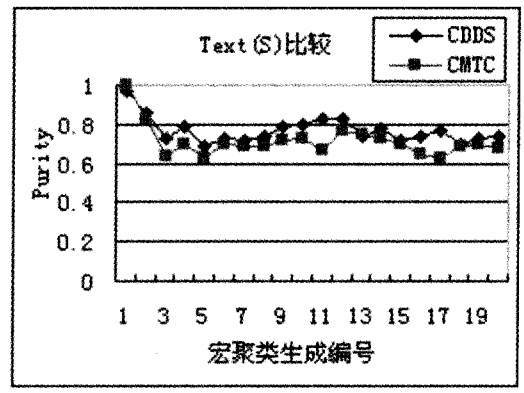

图 1 Text(S)实验数据结果比较

其中 CDDS 的平均纯度为 0.74，CMTC 的为 0.69。有 0.05 的优势。Text(E)的比较结果如图 2 所示。

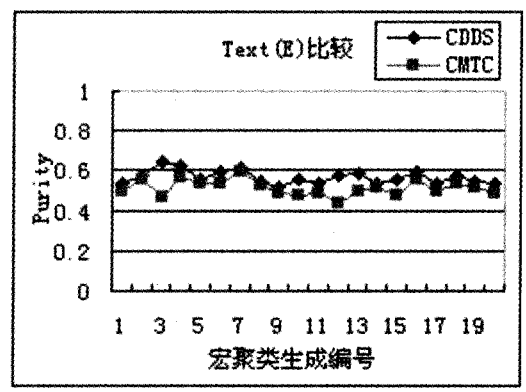

图 2 Text(E)实验数据的结果比较

其中 CDDS 的平均纯度 0.54。CMTC 的平均纯度 0.49，有 0.05 的优势。Text(R)的比较结果如图 3 所法。

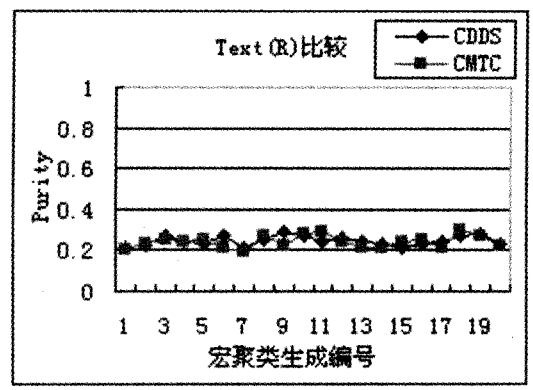

图 3 Text(R)实验数据的结果比较

（下转第 162 页）

从上面可以看出，尽管根据标准被评判为错误的这些关联，在此特定领域和 ADLG 的应用中还是有意义的，对标引员和用户选择主题词也是有帮助的。

结论和展望 本研究从地名中抽取出有检索价值的通用词，然后依据关联统计分析将其中的通用词与 FTT 主题词建立映射。实验从已有的大量语料出发，围绕定位分析问题设计了整套算法，充分利用了 FTT 词表的结构来筛选和关联通用词，同时也考虑到原词表语义上的特点，具有中西文跨语言处理的通用性。实验结果显示，该方法的有效性达到 82.7%。

注意到 ADLG 使用了多个词表标引，包括 FTT、NIMA、USGS 三个词表。基于本研究的成果，可进一步解决不同 Ontology 的互操作问题。解决办法是：应用文中的方法对 NIMA 和 USGS 进行丰富，然后以通用词的关联映射为中介，实现不同地名特征词表中正式主题词的对应和转换；进而为没有结构的词表建立概念等级关系，甚至用来反映各类用户对地理、地质分类认识的差异。

本文提出的分析方法还可以应用到其它类似的语料库上，例如词典、百科全书、分类表、产品目录等，起到发现新术语、丰富词汇辅助系统、更新词表、帮助用户更友好地使用系统等作用。也可应用在数字图书馆、网页博物馆、电子商务网站、企业知识库等各种信息资源组织系统中。

参 考 文 献

1 Velardi P, Fabriani P, Missikoff M. Using text processing techniques to automatically enrich a domain ontology. In: Proceedings of the International Conferenc on Formal Ontology in Information Systems. New York: ACM Press, 2001. 270～284
2 和延立，杨海成，何卫平，等. 信息集成与知识集成. 计算机工程与应用，2003，4：38～41
3 Kramer R, Nikolai R, Habeck C. Thesaurus Federations: Loosely Integrated Thesauri for Document Retrieval in Networks based on Internet Technologies. International Journal of Digital Libraries, 1997, 1(2): 122～131
4 Doerr M. Semantic Problems of Thesaurus Mapping. Journal of Digital Information, 2001, 1(8). http://jodi. ecs. soton. ac. uk/Articles/v01/i08/Doerr/ (accessed Sep 5, 2005)
5 Tudhope D, Alani H, Jones C. Augmenting Thesaurus Relationships: Possibilities for Retrieval. Journal of Digital Information, 2001, 1(8). http://jodi. ecs. soton. ac. uk/Articles/v01/i08/Tudhope/ (accessed Sep 5, 2005)
6 Tseng Y H. Automatic Thesaurus Generation for Chinese Documents. Journal of the American Society for Information Science and Technology, 2002, 53(13): 1130～1138
7 裴炳镇，陈晓明，胡熠，等. 一种建立中文概念分类关系的新算法. 计算机工程与应用，2004，36：18～21
8 Hill L L, Frew J, Zheng Qi. Geographic Names - The Implementation of a Gazetteer in a Georeferenced Digital Library. D-Lib Magazine, 1999, 5(1). http://www. dlib. org/dlib/january99/hill/01hill. html (accessed Sep 5, 2005)
9 Hill L L. Metadata for the ADL Feature Type Thesaurus, 2004-11-15. http://www. alexandria. ucsb. edu/gazetteer/FeatureTypes/FTT_metadata. htm (accessed Sep 5, 2005)
10 Church K W, Hanks P. Word Association Norms, Mutual Information and Lexicography. In: Proceedings of the 27th Annual Conference of the Association of Computational Linguistics. New Brunswick, New York: Association for Computational Linguistics, 1989. 76～83
11 Church K W, Gale W A, Hanks P, et al. Using statistics in lexical analysis. In: Uri Zernik. Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1991. 115～164
12 罗盛芬，孙茂松. 基于字串内部结合紧密度的汉语自动抽词实验研究. 中文信息学报，2003，17(3)：9～14
13 Church K W, Gale W A. Concordances for parallel text. In: Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research. Using Corpora. Berkeley, California: Association for Computational Linguistics, University of California, 1991. 40～62
14 孙茂松，黄昌宁，邹嘉彦，等. 利用汉字二元语法关系解决汉语自动分词中的交集型歧义. 计算机研究与发展，1997，34(5)：332～339
15 刘建舟，何婷婷，骆昌日. 基于语料库和网络的新词自动识别. 计算机应用，2004，24(7)：112～134
16 Turney P D. Coherent keyphrase extraction via Web mining. In: Proceedings Eighteenth International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 2003. 434～439
17 Yang C C, Luk J W K, Yung S K, et al. Combination and Boundary Detection Approach for Chinese Indexing. Journal of the American Society for Information Science (Special topic issue on digital libraries), 2000, 51(4): 340～351
18 张国煊，郁梅，王小华. 基于互信息的汉语短语边界划分. 杭州电子工业学院学报，1995，15(1)：1～5
19 Manning C D, Schütze H. Foundations of Statistical Natural Language Processing. London: The MIT Press, 1999. 182
20 Chan L M, Comaromi J P, Mitchell J S, et al. Dewey Decimal Classification - A Practical Guide. Second Edition. New York: Forest Press, 1996. 55
21 Chan L M. Library of Congress Subject Headings - Principles and Application. Third Edition. Colorado: Libraries Unlimited, 1995. 30

(上接第 127 页)

其中 CDDS 的平均纯度 0.233，CMTC 的平均纯度 0.228。可以认为没有优势。Text(O)的比较结果如图 4 所示。

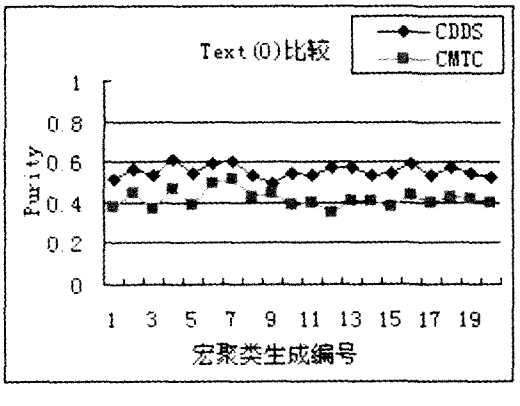

图 4 Text(O)实验数据的结果比较

其中 CDDS 的平均纯度 0.53，CMTC 的平均纯度 0.39，有 0.14 的优势。可以看到，当存在孤立点的时候，该方法具有更大的比较优势。

结束语 本文提出了一种文本流聚类算法，该方法能够在具有演化特征文本流中进行聚类，并且对孤立点不敏感。新的在线微聚类结构提高了聚类的性能；将微聚类分为潜在微聚类和异常微聚类，提高了算法对孤立点的适应能力。实验表明，文中的方法比已有的方法有更好的聚类质量。

参 考 文 献

1 Guha S, Mishra N, Motwani R, et al. Clustering Data Streams. In: IEEE FOCS Conference, 2000
2 O'Callaghan L, et al. Streaming-Data Algorithms For High-Quality Clustering. Wiley Series in Probability and Math. Sciences, 1990
3 Guha S, Rastogi R, Shim K. CURE: An Efficient Clustering Algorithm for Large Databases. In: ACM SIGMOD Conference, 1998
4 Aggarwal C C, Han J, Wang J, et al. A Framework for Clustering Evolving Data Streams. In: VLDB Conference, 2003. 81～92
5 Aggarwal C C. A Framework for Diagnosing Changes in Evolving Data Streams. In: ACM SIGMOD Conference, 2003. 575～586
6 O'Callaghan L, Mishra N, Meyerson A, et al. Streaming-Data Algorithms For High-Quality Clustering. In: ICDE Conference, 2002. 685～696
7 Ordonez C. Clustering Binary Data Streams with Kmeans. DMKD'03, San Diego, CA, USA, June, 2003
8 Aggarwal C C. A Framework for Projected Clustering of High Dimensional Data Streams. In: Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004
9 Aggarwal C C. A Framework for Clustering Massive Text and Categorical Data Streams
10 Cao Feng. Density-Based Clustering over an Evolving Data Stream with Noise. In: Proceedings of the 2006 SIAM Conference on Data Mining (SDM'2006)
11 朱蔚恒，等. 基于数据流的任意形状的聚类算法. 软件学报，2006(3)：379～388