多分辨剪枝局部聚类算法挖掘空间 co-location 模式

吕 诚
(江西理工大学 南昌 330013)

摘 要 传统的 co-location 模式挖掘算法采取对各个特征实例进行逐一连接的挖掘方式,其结果是,常常消耗大量的时间和空间资源,甚至由于内存资源被过度消耗而无法挖掘出最终结果,特别是在数据量大的情况下更是如此。因此,提出了一种高效的多分辨剪枝局部聚类算法(MP_LC)。MP_LC 算法首先对数据区域划分网格,再对各个网格中每一特征的实例进行聚类,求出每一类所包含实例的质心,用质心代替相应的实例集,并进行后续的挖掘。大量实验结果表明,MP_LC 算法具有较高的效率、较高的准确率以及较好的实际应用价值。

关键词 co-location 模式,多分辨剪枝,聚类,质心,实例收缩率
中图法分类号 TP311 **文献标识码** A

Mining Spatial Co-location Pattern with Multiresolution Pruning and Local Clustering Algorithm

LV Cheng
(Jiangxi University of Science and Technology, Nanchang 330013, China)

Abstract The traditional co-location pattern mining algorithms take the mining method that connects each furture instance one by one. As a result, they often consume a large amount of time and space resources, even they are unable to dig out the final results because memory resources are over consumed, especially in the face of a large quantity of data case. Therefore, an efficient multiresolution pruning and local clustering algorithm (MP_LC) was proposed. The MP_LC algorithm firstly divides the data region into grids, then clusteres the instances of each feature in each grid, and calculates the centroid of the instances contained by each cluster, replaces the instance set by the centroid, and finally continues to subsequent mining work. A large number of experimental results indicate that the MP_LC algorithm has high efficiency, high accuracy, and good practical application value.

Keywords Co-location pattern, Multiresolution pruning, Cluster, Centroid, Instance shrinkage rate

空间数据具有独有的复杂性以及广泛的应用前景,与此相应的空间数据挖掘已成为热门的研究领域之一[1,2]。空间 co-location 模式挖掘是空间数据挖掘领域的一个重要任务,其在众多领域得到广泛的应用。

空间 co-location 模式挖掘首先在 2001 年被提出[3],经过十多年的研究,取得了阶段性的成果,如基于最大团的 co-location 模式挖掘[4]、带动态参数的区域 co-location 模式挖掘[5]、无支持度阈值的 co-location 规则挖掘[6]等。提出了基于网格结构的多分辨剪枝 co-location 模式挖掘算法[7]、将 co-location 挖掘问题转化为从事务中挖掘关联规则的方法[8]、部分连接方法[9]、无连接方法[10,11]、高效的 iCPI-tree 算法[12]、基于有序团的最大 co-location 模式挖掘算法[13]、降低实例连接次数的基于密度的方法[14]、用最大参与率度量稀有特征的 co-location 模式挖掘[15]、通过加权参与率进行的伪频繁 co-location 模式挖掘[16]。将映射方式的频繁模式挖掘应用到空间 co-location 挖掘中[17],以减少扫描数据次数。在不确定空间数据方面的 co-location 模式挖掘研究,有可能世界模型的 co-location 模式挖掘[18]、GIS 数据模型的 co-location 模式挖掘[19]、模糊集的 co-location 模式挖掘[20]、在基于区间数表示的不确定数据上进行 co-location 模式挖掘[21]等。

上述工作为空间 co-location 模式挖掘的顺利发展作出了重要的贡献。但在面对海量的特征实例时,算法的效率仍然较低,且常常因为内存资源的过度消耗而无法挖掘出最终结果。对此,通过深入探讨传统挖掘方式中过度消耗内存资源的问题本质,提出了多分辨剪枝局部聚类算法挖掘 co-location 模式。大量实验证明,新算法具有较高的时间效率、较低的内存消耗、较高的准确率和良好的稳定性。

1 相关定义及问题

首先介绍空间 co-location 模式挖掘的相关定义,然后就传统挖掘方式进行深入探讨,找出影响挖掘效率的问题本质,为新算法的提出奠定基础。

1.1 相关定义

空间中每一种事物均为一个空间特征,简称特征,每一种事物的一个具体个体叫做这个特征的一个实例。如某个城市中的医院、学校、银行等各种公共机构均为一个特征,而每种机构的具体个体的集合,如共有 200 所医院,为医院这一特征的实例集合。采取数据结构〈特征 ID,实例 ID,实例的空间坐标〉描述各个实例,$F=\{f_1,f_2,\cdots,f_n\}$ 表示特征集合,其中 f_i 为任一特征。实例集 S_i 表示特征 f_i 的所有实例的集合。如

吕 诚(1987-),男,硕士生,助教,主要研究方向为数据挖掘,E-mail:502619575@qq.com。

在图 1 中，存在特征集合 $F=\{A,B,C,D\}$，特征 A 的实例集合 $S_1=\{A.1,A.2,A.3,A.4\}$等。

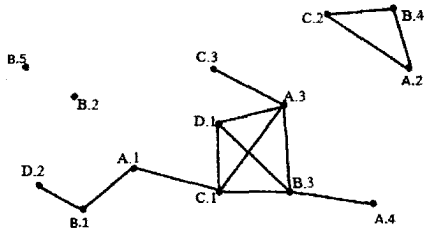

图 1　空间特征及其实例

欧几里德距离不大于距离阈值 d 的两实例 i_1 和 i_2 满足空间邻近关系 R，记 $R(i_1,i_2)\Leftrightarrow(dis(i_1,i_2)\leqslant d)$，其中 dis 表示距离。如在图 1 中，用线段把满足空间邻近关系 R 的实例对连接起来。

空间实例集 $I=\{i_1,i_2,\cdots,i_m\}$，$\forall i_k,i_l\in I$，都有 $R(i_k,i_l)$，其中 $1\leqslant k\leqslant m,1\leqslant l\leqslant m(k\neq l)$，则称 I 为一个团。如图 1 中，$\{A.3,B.3,C.1,D.1\}$是一个团。

空间 co-location 模式是一个特征集合 F 的任一非空子集，记作 c，c 中属于不同特征的实例彼此频繁关联。co-location 模式 c 所含的特征数是 c 的阶，记作 $size(c)$，如 $size(\{A,B,C\})=3$。

对于 co-location 模式 c，如果存在一个与模式 c 中各个特征一一对应的实例集 I 构成一个团，则称 I 为 co-location 模式 c 的一个行实例，记作 $row\text{-}ins(c)$。全部行实例的集合为表实例，记作 $table\text{-}ins(c)$。如在图 1 中，模式 $\{A,B,C\}$的行实例有$\{A.3,B.3,C.1\}$和$\{A.2,B.4,C.2\}$，表实例为$\{\{A.3,B.3,C.1\},\{A.2,B.4,C.2\}\}$。

对于 k 阶 co-location 模式 $c=\{f_1,f_2,\cdots,f_k\}$，f_i 在模式 c 中的参与率 $PR(c,f_i)=\dfrac{|\pi_{f_i}(table\text{-}ins(c))|}{|table\text{-}ins(\{f_i\})|}$，其中 $|\pi_{f_i}(table\text{-}ins(c))|$ 为 f_i 在 c 中不重复出现的实例数，$|table\text{-}ins(\{f_i\})|$为 f_i 的总实例数。参与度 $PI(c)=\min\limits_{i=1}^{k}\{PR(c,f)\}$。如图 1 中，3 阶 co-location 模式 $c=\{A,B,C\}$，$table\text{-}ins(c)=\{\{A.3,B.3,C.1\},\{A.2,B.4,C.2\}\}$，$PR(c,A)=2/4=0.5$，$PR(c,B)=2/5=0.4$，$PR(c,C)=2/3=0.667$。从而 $PI(c)=\min\{PR(c,A),PR(c,B),PR(c,C)\}=0.4$。

对于 k 阶 co-location 模式 $c=\{f_1,f_2,\cdots,f_k\}$，给定一个参与度阈值 min_$prev$，若 $PI(c)\geqslant$min_$prev$，则称 c 是 k 阶频繁 co-location 模式。

1.2　相关问题

在现实世界中，特征实例的分布普遍呈不均匀性，如在一个地区中，处于闹市中的各种特征的实例分布比较密集，郊区的分布则较为稀疏；又如在自然界中，植物的分布也具有类似的特点，同一特征在不同的地区，分布疏密不一，有时甚至极少数的局部区域，分布着该特征的绝大多数实例。面对这种分布特点，传统算法将各个特征的每一个实例作同等对付，逐一处理。其实，针对特征实例分布稀疏的区域，采取这种处理方式当然是无可厚非，但当特征实例分布密集时，这种处理方式务必对特征实例进行过于频繁的连接，耗费了大量的时间和空间资源，而且这种耗费是不必要的。这恰恰是传统算法低效，和常常出现无法挖掘出最终结果的问题本质。

因此，若要从根本上提高算法的挖掘效率，降低算法的内存消耗，保证算法的运行成功率，那么，精简密集区域中特征实例的信息，减少对特征实例的无谓考查与连接，是一个可取的途径。下面将对多分辨剪枝局部聚类算法进行描述。

2　多分辨剪枝局部聚类算法

首先阐述多分辨剪枝局部聚类算法的主要思想及其实现，然后通过一个例子进一步描述算法挖掘的全过程。

2.1　算法的主要思想及实现

算法的主要思想：首先对源特征实例所在区域划分网格，每格为 $d\times d$ 大小的正方形，再对任一网格 G_k 中任一特征 f_i 的实例集依次进行聚类，从而得到各个特征 f_i 的实例簇，求出各个实例簇中所有实例的质心，并用该质心代替相应的实例簇，由此形成多个质心，用这些质心代替原来的实例集进行如下的挖掘：生成一阶粗表实例及一阶细表实例，在一阶粗表实例基础上生成二阶粗表实例并实施剪枝，若无法剪枝，则在此基础上生成二阶细表实例。这里约定，一旦两质心满足邻近关系，两质心所代表的实例彼此间就具有邻近关系。如此类推，直到生成最高阶细表实例为止。

算法的实现：

多分辨剪枝局部聚类算法：

输入：特征集$\{f_i\}$，实例集$\{S_i\}$，距离阈值 d，聚类距离阈值 d_2($d_2<d$)，参与度阈值 PI_th

输出：频繁模式细表实例 rTb

变量：r 是阶数；cTb[r]是 r 阶粗表实例，cTb 是 crudeTable 的缩写；rTb[r]是 r 阶细表实例，cTb 是 refinedTable 的缩写；D 是特征实例分布区域，D 是 domain 的缩写；G 是网格集合；g 是一个网格；cen 是某特征的任一实例(子)集的质心；cenSet[i]是 f_i 的质心集合；$subS_i$ 是 f_i 的任一实例子集；PI(t)是粗(细)表实例 t 的参与度

过程：

```
1. divide D into G;
  //∀g∈G,g 是 d×d 的正方形
2. foreach (g∈G)do
     foreach(fi exist in g)do
       calculate {subSi};//执行全连接聚类算法 d2
       calculate cen from subSi;
       give cen NO;
  cenSet[i]∪=cen;
3. regard cen∈cenSet as instance NO;
4. r=1;
   calculate cTb[r];
   calculate rTb[r];
   do
   {
     calculate cTb[r+1] from cTb[r];
     calculate PI(cTb[r+1]);
     if(PI(cTb[r+1])>PI_th)
     {
       calculate rTb[r+1] from cTb[r+1];
       calculate PI(rTb[r+1]);
       if(PI(rTb[r+1])>PI_th)
       {
         根据 rTb[r+1]更新 cTb[r+1];
```

```
        output rTb[r+1];//输出结果
      }
    }
      r++;
  } while(PI(rTb[r])>PI_th)
```

算法的举例：

例 1　某区域存在 A、B、C 3 个特征的实例集，其中 A 的实例有 9 个，分别为 A. 1－A. 9，B 的实例有 10 个，分别为 B. 1－B10，C 的实例有 5 个，分别为 C. 1－C. 5。以距离阈值 $d=1$ 为网格的边长划分网格，如图 2 所示。

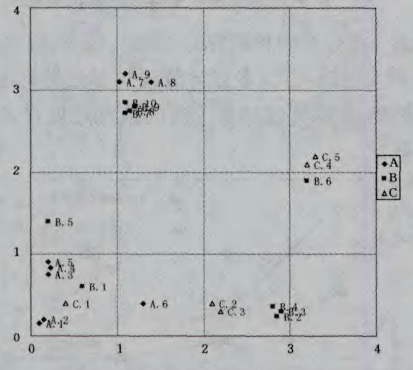

图 2　例 1 的特征实例分布图

以距离阈值为 $d/3=1/3$ 分别对各个网格中的各个特征实例子集进行聚类，并求出各个类的实例子集质心，从而得到如图 3 所示的特征质心分布图。

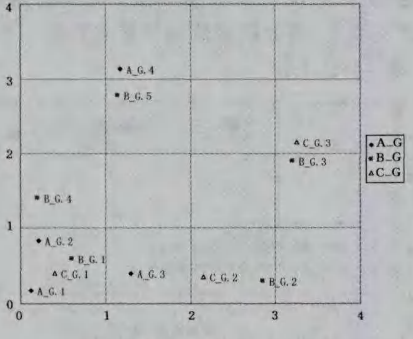

图 3　例 1 的特征质心分布图

根据图 3，可求出如下一阶粗表质心及一阶细表质心。表中各列的最后一个值代表某特征在该表实例中的实例参与率，用带阴影的数值表示。

一阶粗表质心			一阶细表质心		
A	B	C	A	B	C
(0,0)	(0,0)	(0,0)	1	1	1
(1,0)	(0,1)	(2,0)	2	2	2
(1,3)	(1,2)	(3,2)	3	3	3
1	(2,0)	1	4	4	1
	(3,1)		1	5	1
	1			1	

根据一阶粗表质心求得如下二阶粗表质心，若参与度阈值为 0.3，则根据二阶粗表质心求得二阶细表质心，如二阶细表实例中的模式 AB，A 的参与率为 1，B 的参与率为 3/5，因此，模式 AB 的参与度为 $\min\{1,3/5\}=3/5=0.6>0.3$，故得到二阶模式 AB 的频繁模式。当然，若参与度阈值定为 0.7，则二阶粗表实例中模式 AB 的参与度 0.9>0.7，有必要进一步求出其相应的细表实例，而二阶粗表实例中模式 AC、BC 的参与度均为 0.6<0.7，实施剪枝。

二阶粗表质心		
A,B	A,C	B,C
(0,0)(0,0)	(0,0)(0,0)	(0,0)(0,0)
(0,0)(0,1)	(1,0)(0,0)	(0,1)(0,0)
(1,0)(0,0)	(1,0)(2,0)	(2,0)(2,0)
(1,0)(0,1)	2/3,3/5	(3,1)(2,0)
(1,0)(2,0)		(3,1)(3,2)
(1,3)(1,2)		3/5,1
1,9/10		

二阶细表质心		
A,B	A,C	B,C
1,1	1,1	1,1
2,1	2,1	2,2
2,4	3,1	3,3
3,1	3,2	5/9,1
4,5	5/9,2/5	
1,3/5		

由于二阶模式 AB、AC 均为频繁模式，因此有可能存在三阶频繁模式。根据二阶粗表质心生成三阶粗表质心，由于三阶粗表实例 ABC 的参与度 3/5=0.6>0.3，因此必须求出其相应的三阶细表实例 ABC，由于三阶细表实例 ABC 的参与度 1/10=0.1<0.4，其不是频繁的。

三阶粗表质心	三阶细表质心
A,B,C	A,B,C
(0,0)(0,0)(0,0)	1,1,1
(0,0)(0,1)(0,0)	2,1,1
(1,0)(0,0)(0,0)	3,1,1
(1,0)(0,1)(0,0)	1/3,1/10,1/5
(1,0)(2,0)(2,0)	
(1,3)(1,2)	
1,9/10,3/5	

2.2　算法的时间和空间复杂度分析

定义 1（实例收缩率）　设原数据集中总实例个数为 $insS$，应用多分辨剪枝局部聚类算法进行挖掘时，生成的总质心个数为 $cenS$，其中 cen 代表 $centroid$。则称 $(1-\frac{cenS}{insS})\times 100\%$ 为该数据集的实例收缩率。

定理 1　如果多分辨剪枝局部聚类算法的实例收缩率为 $s\%$，则(1)算法从完全 $r-1(r\geqslant 3)$ 阶模式生成完全 r 阶模式表实例的最差时间复杂度为 $O(\frac{n!}{r!(n-r)!}\cdot\frac{[k(1-s\%)]^{2r-2}}{rc^2\cdot cc^2})$，(2)其时间消耗是传统的多分辨剪枝算法的 $(1-s\%)^{2r-2}$，(3)它是全连接算法的 $\frac{(1-s\%)^{2r-2}}{rc^2\cdot cc^2}$。其中，$n$ 为总特征数，k 为各特征包含的平均实例数，rc、cc 分别为网格的行数和列数。

证明：(1)对于传统的多分辨剪枝算法，在特征实例均匀分布的前提下，任一特征分布在每一个网格中的实例数约占该特征总实例数的 $\frac{1}{rc\cdot cc}$，而每一个网格中的实例必须且仅须与其自身及邻近 8 格，共 9 格中的实例进行连接，因此，任一实例只需连接其他各个特征所的 $\frac{9}{rc\cdot cc}$ 实例，设 k 为平均任一特征所含有的实例数，这时，挖掘 1 阶模式所需时间为

nk；对于 2 模式，产生一个表实例需要比较$(\frac{9k}{rc \cdot cc})^2$ 次，2 阶表实例的最大长度为 k^2，且至多有 C_n^2 个 2 阶表实例，故生成 2 阶模式所需的最大时间为 $C_n^2(\frac{9k}{rc \cdot cc})^2$；对于 $r(r \geqslant 2)$ 阶模式，$r-1$ 阶表实例的最大长度为 k^{r-1}，生成 r 阶表实例的前 $r-2$列需要连接约$(\frac{9k^{r-1}}{rc \cdot cc})^2$ 次，对于前 $r-2$ 列所作的每一次连接，最后两列所做的相应连接次数最坏情况下为 k^r，k^r 是 r 阶表实例的最大长度。且至多存在 C_n^r 个 r 阶表实例，因此，传统的多分辨剪枝算法由 $r-1$ 阶生成 r 阶模式的最大时间消耗为 $C_n^r((\frac{9k^{r-1}}{rc \cdot cc})^2 + k^r) = O(\frac{n!}{r!\ (n-r)!} \cdot \frac{k^{2r-2}}{rc^2 \cdot cc^2})$。对于多分辨剪枝局部聚类算法，由于质心的数量只有原总实例数的 $1-s\%$，相当于将 $O(\frac{n!}{r!\ (n-r)!} \cdot \frac{k^{2r-2}}{rc^2 \cdot cc^2})$中的 k 改为 $k(1-s\%)$，因此，多分辨剪枝局部聚类算法由 $r-1$ 阶生成 r 阶模式的最大时间复杂度为 $O(\frac{n!}{r!\ (n-r)!} \cdot \frac{[k(1-s\%)]^{2r-2}}{rc^2 \cdot cc^2})$；(2) $(\frac{n!}{r!\ (n-r)!} \cdot \frac{[k(1-s\%)]^{2r-2}}{rc^2 \cdot cc^2}) / (\frac{n!}{r!(n-r)!} \cdot \frac{k^{2r-2}}{rc^2 \cdot cc^2}) = (1-s\%)^{2r-2}$；(3)全连接算法，可视为传统多分辨剪枝算法在 $rc=1, cc=1$ 的特殊情形，因此由(2)、(3)即可得证。

定理 2 如果多分辨剪枝局部聚类算法的实例收缩率为 $s\%$，则算法从完全 $r-1(r \geqslant 3)$阶模式生成完全 r 阶模式表实例的最差空间复杂度为 $O(\frac{n!}{(r-1)!(n-r)!} \cdot k^r(1-s\%)^r)$，其空间消耗是全连接算法和传统的多分辨剪枝算法的$(1-s\%)^r$。

证明：证明过程与定理 1 类似，在此从略。

由定理 1 和定理 2 可知，与传统的全连接算法及多分辨剪枝算法相比，多分辨剪枝局部聚类算法在时间和空间效率方面均有明显优势。

3 实验分析

实验目的：验证与经典高效的多分辨剪枝算法相比，新算法的高效率、稳定性，及高准确率。

实验环境：所有代码采用 C# 编写，实验在英特尔 CPU Core i3-2100 @3.10GHz 双核、4GB 内存的 PC 机上进行。

实验数据：实验数据分为 1 个真实数据 RDS(real data set)和 4 个混杂数据 MDS1～MDS4(mixture data sets)。其中真实数据来自中国内地某自然保护区的植物物种分布数据；混杂数据 MDS 为在真实数据区域中有目的地按原特征所含实例数的比例进行随机投点，以增大特征实例分布的密度。

3.1 在真实数据集上的实验

在真实数据集 RDS 上的实验分为两部分，首先考查随着距离阈值变化、算法的相关性能；然后考查算法随参与度阈值变化的相关性能。其中，RDS 包含 12 个特征，总实例数为 53000 个，分布在 50000 * 4000 平方米的区域内。

3.1.1 因距离阈值变化算法的性能

如图 4 所示，在参与度阈值 $PI\_th=0.8$ 的情况下，随着距离阈值从 5 增加到 20，与新算法相比，传统的多分辨剪枝算法所消耗的时间明显要多，呈现出骤然上升的指数级增长态势。而两个新算法的时间曲线几乎与 x 轴重合在一起。

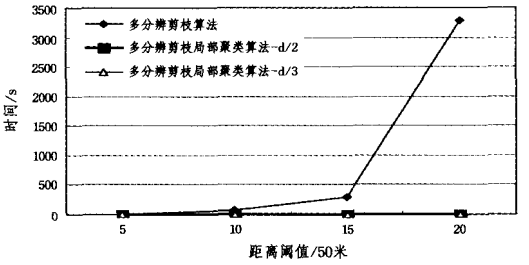

图 4 距离阈值变化 3 个算法的时间消耗

为了进一步分析两种新算法的时间效率，将图 4 放大后得图 5，从图 5 可以看出，多分辨剪枝局部聚类算法－$d/2$ 的时间效率更优，但两算法的最大时间消耗还处在 2.636 秒以内的低水平。

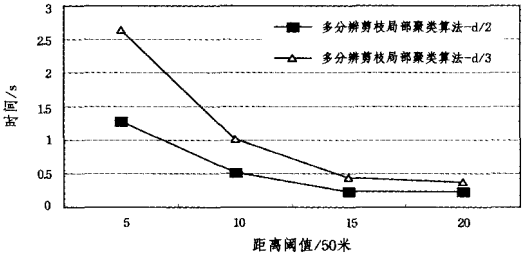

图 5 距离阈值变化 2 个算法的时间消耗

如图 6 所示，传统的多分辨剪枝算法的准确率均为 100%，而多分辨剪枝局部聚类算法－$d/2$ 的准确率也达 83.2%以上的水平，多分辨剪枝局部聚类算法－$d/3$ 的准确率更高达 88.4%以上。

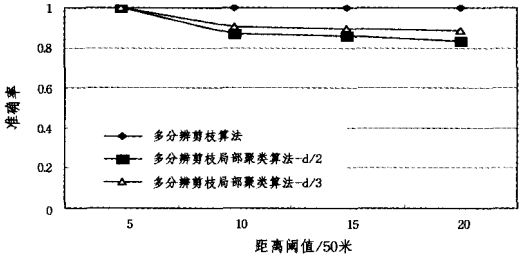

图 6 距离阈值变化 3 个算法的准确率

3.1.2 因参与度变化算法的性能

如图 7 所示，在距离阈值 $d=10$ 的情况下，随着参与度阈值从 0.8 减小到 0.2，与新算法相比，传统的多分辨剪枝算法所消耗的时间明显要多，呈现出骤然上升的指数级增长态势。而两个新算法的时间曲线几乎与 x 轴重合在一起。

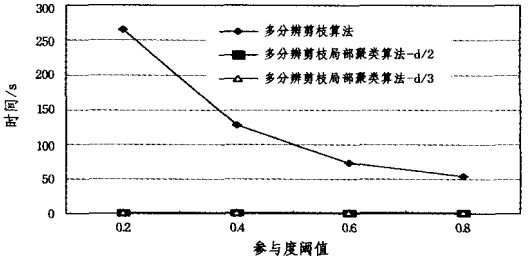

图 7 参与度阈值变化 3 个算法的时间消耗

为了进一步分析两种新算法的时间效率，将图 7 放大后得图 8，从图 8 可以看出，多分辨剪枝局部聚类算法－$d/2$ 的

时间效率更优，但两算法的最大时间消耗还处在 1.206 秒以内的低水平。

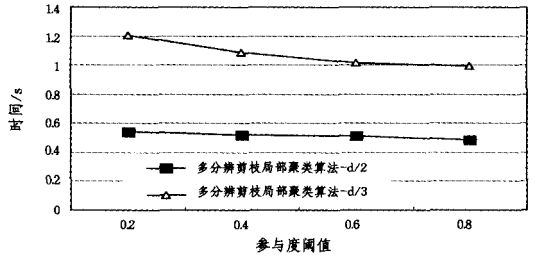

图 8　参与度阈值变化 2 个算法的时间消耗

如图 9 所示，传统的多分辨剪枝算法的准确率均为 100%，而多分辨剪枝局部聚类算法－$d/2$ 的准确率也达 86.9%以上，多分辨剪枝局部聚类算法－$d/3$ 的准确率更高达90.9%以上。

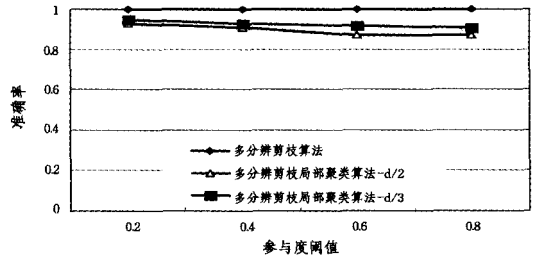

图 9　参与度阈值变化 3 个算法的准确率

从以上实验可以看出，与传统的多分辨剪枝算法相比，多分辨剪枝局部聚类算法具有明显的效率优势，且具有较高的准确率。与多分辨剪枝局部聚类算法－$d/3$ 相比，多分辨剪枝局部聚类算法－$d/2$ 具有一定的效率优势，但前者的准确率更优。之所以出现这种情形，是因为多分辨剪枝局部聚类算法－$d/3$ 采取距离阈值更小的聚类方式，产生质心的颗粒度更小，质心更多。

3.2　在混杂数据集上的实验

如图 10 所示，在混杂数据 MDS1－MDS4 集中进行实验，参与度阈值 $PI\_th=0.6$，距离阈值 $d=20$，随着总实例数从 50 万个增加到 200 万个，2 个多分辨剪枝局部聚类算法的时间消耗增长缓慢，基本呈线性性。值得一提的是，相对于传统算法所能承受的实验数据量，这时数据量已经相当庞大，传统算法因为内存溢出面无法挖掘出最终结果，因此，该小节中的实验无法将新算法与传统算法做比较。

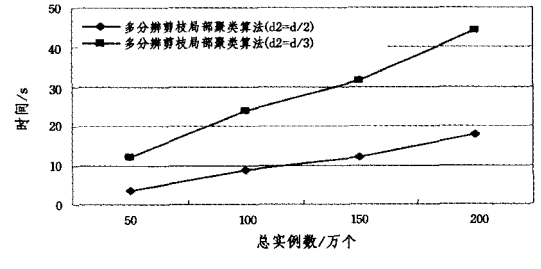

图 10　总实例数变化 2 个算法的时间消耗

结束语　本研究提出了 MP_LC 算法，与传统的多分辨剪枝算法相比，新算法具有明显的效率优势：较好地解决了面对海量数据，传统算法常常发生的内存溢出问题，进而较好地避免了由于过度消耗空间资源而无法挖掘出最终结果的尴尬；算法的准确率较高；能较好地解决现实生活中存在的实际问题。

当然，本研究也存在一些不足，如相关阈值均人为预先给定，难以做到科学合理。因此，未来将在本研究基础上，引入自动产生阈值的思想，以进一步提高新算法的应用价值。

参 考 文 献

[1] Gao Yun-jun, Zheng Bai-hua. Continuous Obstructed Nearest Neighbor Queries in Spatial Databases[C]// ACM SIGMOD. NY, USA, 2009: 577-590

[2] Ruggieri S. Frequent regular itemset mining [C]// KDD. ACM, Washington DC, USA, 2010: 263-272

[3] Shekhar S, Huang Yan. Discovering Spatial Co-location Patterns: A Summary of Results [C]// Proc. of the 7th International Symposium on Advances in Spatial and Temporal Databases. CA, USA, 2001: 236-240

[4] Al-Naymat G. Enumeration of maximal clique for mining spatial Co-Location Patterns [C]// 6th ACS/IEEE International Conference on Computer Systems and Applications. Doha, Qatar, 2008: 126-133

[5] Celik M, Kang J M, Shekhar S. Zonal Co-Location Patterns Discovery with Dynamic Parameters[C]// 7th IEEE International Conference on Data Mining. Omaha NE, USA, 2007: 433-438

[6] Huang Yan, Xiong Hui, Shekhar S, et al. Mining confident Co-Location rules without a support threshold[C]// ACM Symposium on Applied Computing. Florida, USA, 2003: 497-418

[7] Huang Yan, Shekhar S, Xiong H. Discovering Co-location Patterns from Spatial Data Sets: A General Approach [J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16 (12): 1472-1485

[8] Koperski K, Han J. Discovery of Spatial Association Rules in Geographic Information Databases[C]// Proc. of the 4th International Symposium on Spatial Databases. Portland, Maine, 1995: 47-66

[9] Yoo J S, Shekhar S. A partial Join Approach for Mining Co-location Patterns [C]// the 12th Annual ACM International Workshop on Geographic Information Systems. NY, USA, 2004: 241-249

[10] Yoo J S, Shekhar S, Celik M. A Join-Less Approach for Co-Location Pattern Mining: A Summary of Results [C]// The 5th IEEE International Conference on Data Mining. NY, USA, 2005: 813-816

[11] Wang L, Bao Y, Lu J, et al. A New Join-less Approach for Co-location Pattern Mining [C]// The 8th IEEE International Conference on Computer and Information Technology. Sydney, Australia, 2008: 197-202

[12] Wang Li-zhen, Bao Yu-zhen, Lu Zhong-yu. Efficient Discovery of spatial co-location patterns using the iCPI-tree [J]. The Open Information Systems Journal, 2009, 3 (1): 69-80

[13] Wang Li-zhen, Zhou Li-hua, Joan L. An order-clique-based approach for mining maximal co-locations [J]. Information Sciences, 2009, 179 (19): 3370-3382

[14] Xiao X, Xie X, Luo Q. Density-based co-location pattern discovery[C]// Proc. of the 16th ACM International Conference on Advances in Geographic Information Systems. Irvine, California, 2008: 11-20

[15] Huang Yan, Pei J, Xiong H. Mining Co-location Patterns with

Rare Events from Spatial Data Sets[J]. Geoinformatica, 2006, 10(3): 239-260

[16] Xiao X, Xie X, Luo Q. Density-based co-location pattern discovery[C]// Proc. of the 16th ACM International Conference on Advances in Geographic Information Systems. Irvine, California, 2008: 11-20

[17] Huang Yan, Pei J, Xiong H. Mining Co-location Patterns with Rare Events from Spatial Data Sets [J]. Geoinformatica, 2006, 10(3): 239-260

[18] Wang Li-zhen, Wu Ping-ping, Chen Hong-mei. Finding Probabilistic Prevalent Co-locations in Spatially Uncertain Data Sets [J]. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2011, 25(4): 790-804

[19] Sheng C, Hsu W, Lee M. Discovering Spatial Interaction Patterns [M]. Berlin: Springer, 2008: 95-109

[20] 欧阳志平，王丽珍，陈红梅. 模糊对象的空间 Co-Location 模式挖掘研究[J]. 计算机学报，2012(10): 1947-1956

[21] Wang Li-zhen, Chen Hong-mei, Zhao Li-hong, et al. Efficiently Mining Co-Location Rules on Interval Data [C]// ADMA 2010, Part I. LNCS 6440, 2010: 477-488

（上接第 300 页）

3 实验结果

文本采用中科院 ICTCLAS 对中文文档进行分词处理，针对自然语言处理面向真实语料与面向实例化的趋势，文本的测试基于 300 篇真实文本。这些文本是由设计爬虫从新浪新闻频道随机爬取的各类新闻。文本的 LDA 模型的主题个数 k 设定为 200，超参数根据经验[17]设定为 α 为 0.25，β 为 0.01，迭代次数设定为 200。并设定句子字数的阈值为 10，不选取低于该阈值的句子作为最终文档摘要的候选句。同时设定摘要字数的上限值为 200，在计算句子权重之后，根据权重选取权重从高到低的一个或多个句子作为文档的摘要，选取句子的个数依赖于已经选取的句子的字数，使最终的文档摘要总字数小于我们设定的上限值。最后将每篇文档的摘要与文档的内容与文档标题进行对比，并判断摘要与文档内容的相关程度。

文本采用人工打分对摘要结果进行评测，打分分为 3 个标准，分别是准确反映主题、基本反映主题、没有很好反映主题。同时，本文为了减少人为差异对最终统计结果的影响，最终的结果为去掉最高和最低项之后的均值。具体的测试结果如表 1 所列。

表 1 摘要测试结果

分类标准	评价结果	比例
准确反映主题	184	61.33%
基本反映主题	90	30.00%
没有很好反映主题	26	8.67%

结束语 文本提出了基于主题的文档摘要算法，通过主题得到文档中不同词语的生成概率。同时在得到了词语生成概率之后，本文对句子进行了概率建模，从而引入了信息熵来对句子的权重进行度量。为了验证该方法的效果，本文随机爬取了新浪新闻频道的若干新闻，并进行了实验。实验结果表明，在合适的模型参数的情况下，该方法抽取的摘要能较好地概括文档的主要内容。

参 考 文 献

[1] Luhn, Hans P. The automatic creation of literature abstracts [J]. IBM Journal of research and development, 1958, 2(2): 159-165

[2] Edmundson, Harold P, Wyllys R E. Automatic abstracting and indexing—survey and recommendations[J]. Communications of the ACM, 1961, 4(5): 226-234

[3] Edmundson, Harold P. New methods in automatic extracting [J]. Journal of the ACM(JACM), 1969, 16(2): 264-285

[4] Pollock, Joseph J, Zamora A. Automatic abstracting research at chemical abstracts service[J]. Journal of Chemical Information and Computer Sciences, 1975, 15(4): 226-232

[5] Paice, Chris D. The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases [C]// Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval. Butterworth & Co., 1980

[6] Salton, Gerard, et al. Automatic text structuring and summarization[J]. Information Processing & Management, 1997, 33(2): 193-207

[7] Blair-Goldensohn, Sasha, et al. Columbia university at duc 2004 [C]// Proceedings of the Document Understanding Conference, DUC-2004. Boston, USA, 2004

[8] 王继成，武港山. 一种篇章结构指导的中文 Web 文档自动摘要方法[J]. 计算机研究与发展，200340(3): 398-405

[9] 张奇，黄萱菁，吴立德. 一种新的句子相似度度量及其在文本自动摘要中的应用[J]. 中文信息学报，2005, 19(2): 93-99

[10] 尹存燕，戴新宇，陈家骏. Internet 上文本的自动摘要技术[J]. 计算机工程，2006, 32(3): 88-90

[11] 张云涛，龚玲，王永成. 基于综合方法的文本主题句的自动抽取[J]. 上海交通大学学报，2006, 40(5): 771-774

[12] 纪文倩，等. 一种基于 LexRank 算法的改进的自动文摘系统[J]. 计算机科学，2010, 37(5): 151-154

[13] 罗文娟，等. 权衡熵和相关度的自动摘要技术研究[J]. 中文信息学报，2011, 25(5): 9-16

[14] 任昭春，马军，陈竹敏. 基于动态主题建模的 Web 论坛文档摘要[J]. 计算机研究与发展，2013, 49(11): 2359-2367

[15] 刘平安. 基于 HLDA 模型的中文多文档摘要技术研究[D]. 北京：北京邮电大学，2013

[16] http://zh.wikipedia.org/wiki/隐含狄利克雷分布

[17] Blei, David M, Ng A Y, et al. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003, (3): 993-1022

[18] Wei X, Croft W B. LDA-based document models for ad-hoc retrieval[C]// Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2006: 178-185