

一种面向社会化标签的并行双聚类算法

孙 闻,韩立新

(河海大学 计算机与信息学院,江苏 南京 211100)

摘 要:随着社会化网络的快速发展,社会化数据呈现爆炸式增长,挖掘社会化数据的局部信息成为有效利用社会化数据的研究热点。相对于传统聚类方法,双聚类能够更好地挖掘社会化数据中的局部信息。较高的计算复杂度成为使用双聚类挖掘大数据集中局部信息的关键问题。通过对几何双聚类产生过程的研究与分析,提出了一种改进的并行几何双聚类方法。该方法通过过滤子双聚类合并过程中产生无效的子双聚类,降低算法的计算量,而且利用多核计算机的优势,使用并行算法,从而提高双聚类算法的效率。

关键词:社会化数据;并行;几何双聚类;子双聚类

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2015)05-0033-04

doi:10.3969/j.issn.1673-629X.2015.05.009

A Parallel Biclustering Algorithm for Social Tagging

SUN Wen,HAN Li-xin

(College of Computer and Information,Hohai University,Nanjing 211100,China)

Abstract:With the rapid development of social networks,a great number of social data can be acquired.Extracting the local information has become the focus in the research of social data.Compared with traditional clustering method,biclustering can better exploit the local information of the social data.However,the computational complexity of biclustering is high,which is the bottleneck of mining the local information.Based on researching and analyzing the process of generating biclustering,propose an improved parallel geometric biclustering method.By filtering the invalid biclustering which was generated in the process of combining sub-bicluster,the complexity can be reduced.Moreover,by using the advantages of multi-core processors and the parallel algorithm,can improve the efficiency of biclustering on social data.

Key words:social data;parallel;geometric biclustering;sub-biclustering

0 引 言

随着社会化网络的不断发展,呈爆炸式增长的海量数据给互联网带来了丰富的社会化信息。社会化标注是网络用户的一种自发的、自主的对网络资源的描述行为。但是,社会化标注的自由性同时也导致了标注数据的二义性、无层次性等问题^[1],这些问题给用户从社会化标注数据中获取有用信息带来了较大的障碍。因此,挖掘社会化标注数据中的有效信息成为一个研究热点。

双聚类的概念被 Cheng 和 Church 首次提出,用于基因表达信息的分析^[2]。现在,双聚类成为一种倍受研究者青睐的数据挖掘方式,被广泛应用于生物医药^[3]、金融^[4]、图像处理^[5]以及社会化数据挖掘^[6]等领域。

社会化标注数据的有效信息往往隐藏在局部数据中,而传统聚类方法只能挖掘出数据的全局信息,因而在社会化标注数据中传统聚类方法很难挖掘到有用的局部信息^[7]。与传统聚类方法不同,双聚类同时从数据矩阵的行和列两个维度进行聚类,挖掘数据的局部信息。然而,双聚类是一个 NP-hard 问题^[8],具有较高的计算复杂度。与其他双聚类算法相比,几何双聚类在进行数据的相干模式检测中,可以获得较高精度,并且具有相对较小的计算复杂度^[9]。但是几何双聚类较高的时间复杂度仍然是其应用于大数据集的瓶颈。

此外,由于各种领域中数据的海量增长,使得高性能计算在大规模数据挖掘中获得了越来越广泛的应用^[10]。并行计算是将任务分配到不同的线程中,这些

收稿日期:2014-06-18

修回日期:2014-09-22

网络出版时间:2015-02-23

基金项目:国家自然科学基金资助项目(60971088)

作者简介:孙 闻(1990-),男,硕士研究生,通讯作者,研究方向为信息检索、数据挖掘;韩立新,教授,博士生导师,研究方向为信息检索、模式识别、数据挖掘。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150223.1257.058.html>

线程由多个 CPU 来执行计算工作,从而加快了计算速度^[11]。并行计算作为高性能计算的有效工具,被广泛应用于文本分类^[10]、关联规则分析^[12]、双聚类^[13]等。

文中主要针对以上提出的问题,讨论了几何双聚类在社会化标注数据中的应用,通过过滤无效的子双聚类,减少了子双聚类的出现,从而减少了算法的计算量,并且利用多核计算机的并行计算优势,提高了几何双聚类算法的效率,从而提出了一种基于改进的并行几何双聚类的标签聚类方法。

1 几何双聚类

1.1 几何双聚类工作流程

几何双聚类工作流程如图 1 所示。

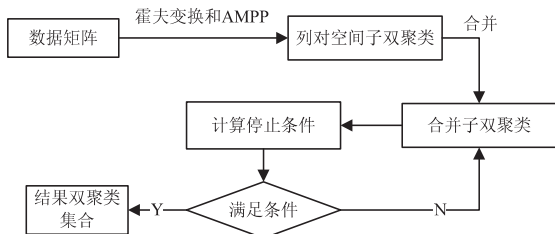


图 1 几何双聚类算法工作流程

AMPP(Additive and Multiplicative Pattern Plot)判断数据矩阵中任意两列之间的加法相干性或乘法相干性,霍夫变换将二维数据矩阵映射为霍夫空间的列对,在霍夫空间的每一个列对都视为一个最小双聚类。AMPP 可以区分在列对空间形成的子双聚类属于加法相干性模式还是乘法相干性模式^[9]。在数字图像处理中,霍夫变换(Hough Transform, HT)是模式识别领域中对二值图像进行线性检测的有效方法^[14]。而在列对空间中进行的线性检测是几何双聚类算法中至关重要的一步。列对空间中的每个列对都是一个最小双聚类单元,以最小双聚类单元为基础,通过使用几何双聚类算法合并较小的双聚类,从而产生较大的双聚类。在合并过程中,双聚类的行数逐渐减小,列数逐渐增加,算法将过滤掉那些行数小于某一特定阈值的双聚类。在算法停止后,就可以获得有效的双聚类的集合。

定义 1: D 为一个社会化标注数据矩阵,每一行代表一个标签(t_i),每一列代表一个资源(r_j),矩阵的元素 d_{ij} 代表第 j 个资源被第 i 个标签标注的次数。 c_i 、 c_j 分别为数据矩阵 D 中的第 i 列和第 j 列,若对于 c_i 和 c_j 的对应位置的非零元素,都满足 $c_{im} = c_{jm} + C$,其中 C 为某一常量,则列对 c_{ij} 属于加法相干模式;若对于 c_i 和 c_j 的对应位置的非零元素,都满足 $k = \arctan(c_{jm}/c_{im})$,其中 k 为某一常量,则列对 c_{ij} 属于乘法相干模式。

假设有一 m 行 n 列的数据矩阵,该矩阵通过霍夫变换转换为一个包含 $n(n-1)/2$ 列的矩阵。如图 2

所示,一个 5 行 4 列的数据矩阵被转换为一个 5 行 6 列的列对矩阵。例如,列对 $C_{1,2}$ 代表数据矩阵中第 1 列和第 2 列进行合并的结果。 $C_{1,2}$ 中值为 1 的位表示在原数据矩阵中第 1 列和第 2 列对应的位置具有加法相干性模式。在合并处理过程中,所有满足条件的列对都会被用来合并为有效的双聚类。

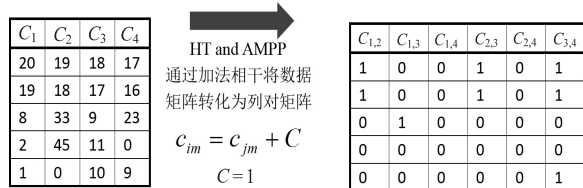


图 2 数据矩阵到列对空间的映射过程

1.2 几何双聚类合并过程

传统几何双聚类算法^[9]的合并过程如算法 1 所示。

算法 1: 传统几何双聚类合并算法。

输入: 列对矩阵 $M_c, C_{i,j} \in M_c$, 其中 $1 \leq i \leq n, 1 \leq j \leq n$ 且 $i \neq j$; TH_1 , 可以接受的双聚类中最少包含的行数;

输出: 所有满足条件的双聚类集合 M'_c 。

```

1 for all  $C_{i,j} \in M_c$ 
2 if number of 1 in  $C_{i,j} \geq TH_1$  then
3  $M'_c \leftarrow C_{i,j}$ 
4 end if
5 end for
6 for  $b_1$  in  $M'_c$  do
7 for  $b_2$  in  $M'_c$  do
8 if  $b_{1\_columns} \not\subseteq b_{2\_columns}$  and  $b_{2\_columns} \not\subseteq b_{1\_columns}$  do
9 if  $b_{1\_rows} \cap b_{2\_rows} \neq \emptyset$  and  $\setminus$ 
 $b_{1\_columns} \cap b_{2\_columns} \neq \emptyset$  then
10  $B_{combine} \{rows, columns\} = \{b_{1\_rows} \& b_{2\_rows}, \setminus b_{1\_columns} \setminus$ 
 $b_{2\_columns}\}$ 
11 if  $\# \{B_{combine\_rows}\} \geq TH_1$  then
12  $M'_c \leftarrow B_{combine} \cup M'_c$ 
13 end if
14 end if
15 end if
16 end for
17 end for
  
```

若 M 为一 m 行 n 列的数据矩阵,则经过变换后的列对矩阵 M_c 包含 $n(n-1)/2$ 个列对, $C_{i,j}$ 为数据矩阵第 i 列与第 j 列在列对矩阵中对应的列对向量。因为在合并过程中,双聚类的行数是不断减小的,若作为双聚类最小单元的列对的行数小于 TH_1 ,则由其合并而来的双聚类也会小于 TH_1 。因此,算法 1 中,在 1~5 行过滤掉所有不满足条件的列对;6~17 行为算法对双聚类集合中的双聚类进行两两合并的过程,直到不再产生新的双聚类,算法结束。第 8 行与第 9 行是对所有双聚类 b_1 和 b_2 进行合并,其中要求 b_1 与 b_2 中包

含的列有非空交集, b_1 中至少存在一列不在 b_2 中, b_2 中至少存在一列不在 b_1 中, 而且 b_1 与 b_2 的行有非空交集。在第 11 行中, $\# \{ B_{\text{combine_rows}} \}$ 表示双聚类 B_{combine} 中包含的行数。

1.3 算法并行性分析

几何双聚类算法最耗时的处理是双聚类的合并过程。文献[15]分析了几何双聚类算法的合并过程具有可并行性, 充分利用现有多核 CPU 的优势, 通过在同一时间合并多个双聚类, 可以加快双聚类的合并速度。

2 基于多核 CPU 的并行几何双聚类算法

传统几何双聚类算法^[9]在合并双聚类的过程中, 会产生大量的无效中间双聚类, 导致算法的复杂度增加, 而且串行化的传统几何双聚类算法在效率上很难满足处理大数据集的需求。文中针对以上问题, 对传统几何双聚类算法进行了改进, 过滤掉无效的双聚类, 减少中间双聚类的产生, 从而降低了算法的计算复杂度, 并且利用并行计算的优势, 提高了算法的执行效率。

2.1 社会化数据的映射策略

由于社会化标注数据本身的稀疏性, 以及矩阵中大部分数据集中在 0、1、2 等较小的数值上, 且伴随着数值的增大, 标注的数据量快速减小。在原始数据上, 很难发现有效的数据相干性模式。为了使数据的相干性更加明显, 文中将社会化标注数据进行一次映射, 使其更加适合于几何双聚类处理。映射公式如式(1)所示。

$$c'_{i,j} = \begin{cases} 0, c_{i,j} = 0 \\ 1, \arg_1 \leq c_{i,j} < \arg_2 \\ 2, \arg_2 \leq c_{i,j} < \arg_3 \\ 3, \arg_3 \leq c_{i,j} < \arg_4 \\ 4, c_{i,j} \geq \arg_4 \end{cases} \quad (1)$$

其中, $c_{i,j}$ 为原始数据矩阵中第 i 行第 j 列的数据; $c'_{i,j}$ 为映射后的数据矩阵第 i 行第 j 列的数据; \arg_1 、 \arg_2 、 \arg_3 、 \arg_4 为映射可调参数。映射后的数据保持了原始数据中数据的相对大小关系, 而且增强了数据的相干性。

2.2 改进的几何双聚类算法

在传统几何双聚类算法^[9]中, 合并较小的双聚类时会产生大量的中间双聚类, 而这些中间双聚类大部分都是一些已有双聚类的子双聚类, 即包含于已有双聚类。假设, 在 M_c 中存在一个双聚类 $C_{1,2,3,4}$, 它是通过合并双聚类 $C_{1,2,3}$ 与双聚类 $C_{2,4}$ 得到的。然而, 传统几何双聚类^[9]仍然会产生双聚类 $C_{1,3,4}$ 和双聚类

$C_{2,3,4}$ 。双聚类包含的列数越多, 其子双聚类也就会越多, 也就意味着会产生更多的中间双聚类。这些中间双聚类会导致算法的计算量大幅度增加, 大量的 CPU 时间将会被浪费在计算这些中间双聚类上。因此, 改进的算法在合并双聚类的过程中, 丢弃那些无用的中间双聚类。以图 2 中的列对矩阵为例, 如图 3 所示, 在 Step1 中由 $C_{1,2}$ 与 $C_{2,4}$ 合并为 $C_{1,2,4}$, Step2 中 $C_{1,2,4}$ 与 $C_{3,4}$ 合并为 $C_{1,2,3,4}$, Step3 中 $C_{2,3}$ 与 $C_{3,4}$ 合并为 $C_{2,3,4}$, 而此时已经存在 $C_{1,2,3,4}$, $C_{2,3,4}$ 就变成了一个多余的双聚类, Step4 的计算也就没有了意义。

通过过滤多余的双聚类, 可以减少集合 M_c 中双聚类的个数, 从而能够在保持双聚类的完整信息的情况下, 降低算法的计算复杂度。

Step1: $C_{1,2} \& C_{2,4} \rightarrow C_{1,2,4}$ (11 000)
Step2: $C_{1,2,4} \& C_{3,4} \rightarrow C_{1,2,3,4}$ (11 000)
Step3: $C_{2,3} \& C_{3,4} \rightarrow C_{2,3,4}$ (11 000)
Step4: $C_{2,3,4} \& C_{1,2} \rightarrow C_{1,2,3,4}$ (11 000)

图 3 传统几何双聚类合并过程

2.3 并行几何双聚类算法

多核处理器已经成为硬件发展的主流趋势, 多核处理器的总线控制技术, 以及内存通信技术的发展为解决分布式结构网络通信的问题提供了一种有效的解决方案^[12]。各种新型体系结构的不断涌现^[16], 加快了并行计算的发展速度。在双聚类合并过程中, 每对双聚类的合并都是相互独立的。利用多核处理器的并行优势, 可以加快双聚类合并过程的计算, 缩短算法收敛时间。具体算法如算法 2 所示。

算法 2: 并行几何双聚类算法。

输入: 列对矩阵 M_c , $C_{i,j} \in M_c$, 其中 $1 \leq i \leq n, 1 \leq j \leq n$ 且 $i \neq j$; TH_1 , 可以接受的双聚类中最少包含的行数;

输出: 所有满足条件的双聚类集合 M_c 。

```

1 for all  $C_{i,j} \in M_c$ 
2 if number of 1 in  $C_{i,j} \geq TH_1$  then
3 Temp  $M_c \leftarrow C_{i,j}$ 
4 end if
5 end for
6 Copy Temp  $M_c$  to global Memory  $M_c$ 
7 //保存已经处理过的双聚类的索引的全局矩阵
8 Mindex  $\leftarrow \{ \}$ 
9 for  $b_1$  in  $M_c$  and  $b_1$  not in Mindex do
10 Mindex  $\leftarrow$  Mindex  $\cup b_1\_columns$ 
11 for  $b_2$  in Temp  $M_c$  do
12 if  $b_1\_columns \not\subseteq b_2\_columns$  and  $b_2\_columns \not\subseteq b_1\_columns$ 
do
13 if  $b_1\_rows \cap b_2\_rows \neq \emptyset$  and  $b_1\_columns \cap b_2\_columns \neq \emptyset$ 
then
14  $B_{\text{combine}} \{ rows, columns \} = \{ b_1\_rows \& b_2\_rows, b_1\_columns \cup b_2\_columns \}$ 

```

```

15 if  $B_{combine\_rows} \geq TH_1$  then
16 if  $\# \{ b \mid B_{combine\_columns} \subseteq b\_columns, \forall b \in M'_c \} = 0$ 
then
17  $M'_c \leftarrow B_{combine} \cup M'_c$ 
18 end if
19 end if
20 end if
21 end if
22 end for
23 end for
    
```

在算法 2 中,1~5 行与算法 1 中相同。与传统几何双聚类^[9]不同之处在于,传统几何双聚类使用的是串行算法,而算法 2 使用的是并行算法,可以同时合并多个双聚类,从而减少算法执行时间。第 6 行,创建全局内存 M'_c ,并将所有满足条件的列对拷贝到该全局内存中;第 8 行在全局内存中创建一个空的集合,该集合用于保存所有处理完的双聚类的索引,即双聚类包含的列的标号的集合;9~23 行为算法并行执行部分,每个线程独立运行该段程序,每个线程处理两个双聚类的合并操作。此外,算法 2 在算法 16~18 行,对无效的中间双聚类进行了过滤。第 16 行中, $\# \{ b \mid B_{combine_columns} \subseteq b, \forall b \in M'_c \}$ 表示在 M'_c 中,所有列标号集合包含 $B_{combine_columns}$ 的双聚类的个数,如果该值为 0,则 $B_{combine}$ 不是 M'_c 中任何双聚类的子双聚类,可以将其作为新的双聚类添加到双聚类集合 M'_c 中。

3 实验及分析

实验数据采用 HetRec2011 Datasets 中的 delicious-2k 数据集,实验使用数据是从该数据集中抽取了 2 000 个资源,2 000 个标签,以及这 2 000 个标签在这 2 000 个资源上的标注数据。抽取出来的数据集中共有 49 472 条标注记录,在数据矩阵中有 34 067 个非零元素,稀疏度为 0.851 7%。实验环境为 Intel 4 核 3.4 GHz CPU(Intel Core i7-2600),8 G RAM, Linux 64 位操作系统,算法使用 python 实现。

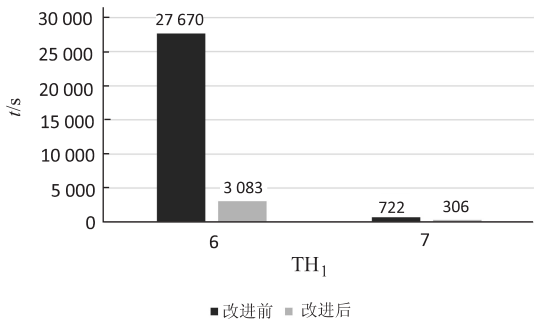


图 4 算法改进前后效率对比

图 4 中对比了 TH₁ 分别为 6 和 7 时,改进后的算

法与传统几何双聚类算法^[9]的性能。当 TH₁ 为 6 时,传统几何双聚类算法^[9]产生的双聚类个数为 24 109,算法执行时间为 27 670 s;改进后的算法产生的双聚类个数为 7 196,算法执行时间为 3 083s。当 TH₁ 为 7 时,传统几何双聚类算法^[9]产生的双聚类个数为 3 942,算法执行时间为 722 s;改进后的算法产生的双聚类个数为 2 401,算法执行时间为 306 s。通过改进算法,大部分无效的双聚类被过滤掉了,双聚类集合中双聚类的个数大为减少,算法的执行效率获得了较大的提升。

图 5、图 6 中分别对比了 TH₁ 为 5、6,线程数分别为 1、2、3、4 时算法的性能。如图所示,并行算法的性能相对于串行算法获得了一定的提升。

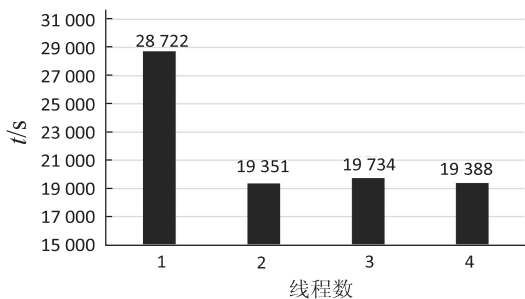


图 5 TH₁ 为 5 时算法效率对比

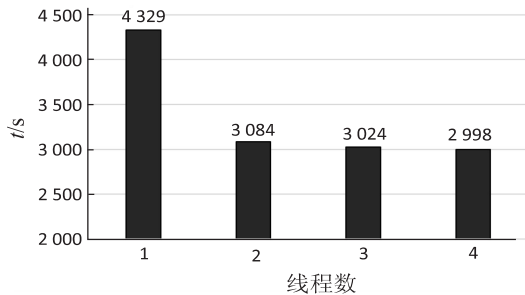


图 6 TH₁ 为 6 时算法效率对比

4 结束语

由于社会化数据的快速增长,数据维度的快速增长,传统的聚类方法难以挖掘数据的局部信息,双聚类方法虽然能够较好地发现隐藏在局部数据中的信息,但较大的计算复杂度成为双聚类方法应用的一大障碍。文中提出了一种改进的并行几何双聚类算法,通过过滤无效的子双聚类,有效地控制中间数据的数量,从而显著降低了算法的计算量,并且结合多核处理器的并行优势,在几何双聚类合并过程中,并行化子双聚类的合并操作,从而提升算法的执行效率。

参考文献:

[1] 魏建良,朱庆华. 社会化标注理论研究综述[J]. 中国图书馆学报,2009(6):88-96.

的地层缺失等复杂地质情况具有良好的适应性。

参考文献:

- [1] 李 鹏,张 芸,王 宁,等. 密集段及其在区域地层对比中的意义[J]. 石油地质与工程,2012,26(6):5-9.
 - [2] 丁 艳,许长福,何幼斌,等. 克拉玛依油田六中区下克拉玛依组精细地层对比研究[J]. 长江大学学报:自然科学版,2011,8(8):32-34.
 - [3] 邓宏文,王洪亮,李熙喆. 层序地层地层基准面的识别,对比技术及应用[J]. 石油与天然气地质,1996,17(3):177-184.
 - [4] 邵才瑞,李洪奇,张福明,等. 人工智能地层对比专家系统原理[J]. 石油物探,2000,39(1):77-84.
 - [5] 杨合群,李 英,赵国斌,等. 新疆—甘肃—内蒙古衔接区地层对比及其意义[J]. 西北地质,2009,42(4):60-75.
 - [6] 刘衍民,牛 奔,赵庆祯. 多目标优化问题的粒子群算法仿真研究[J]. 计算机应用研究,2011,28(2):458-460.
 - [7] 刘海江,黄 炜. 基于粒子群算法的数控加工切削参数优化[J]. 同济大学学报:自然科学版,2008,36(6):803-806.
 - [8] Durillo J J, Garcia-Nieto J, Nebro A J, et al. Multi-objective particle swarm: optimizers an experimental comparison [C]//Proc of 5th international conference on evolutionary multi-criterion optimization. Nantas:Spring,2009:495-509.
 - [9] 王丽萍,江 波,邱飞岳. 基于决策偏好的多目标粒子群算法及其应用[J]. 计算机集成制造系统,2010,16(1):140-148.
 - [10] 杨 帆,胡春平,颜学峰. 基于蚁群系统的参数自适应粒子群算法及其应用[J]. 控制理论与应用,2010,27(11):1479-1488.
 - [11] Abido M A. Multi-objective particle swarm optimization for environmental/economic dispatch problem[J]. Electric Power System Research,2009,79(7):1105-1113.
 - [12] Worasuchee C. A particle swarm optimization with stagnation-detection and dispersion [C]//Proceedings of IEEE conference of evolutionary computation. [s. l.]: IEEE,2008:424-429.
 - [13] 钱 志,胡心红,杨宏伟,等. 综合利用多种测井曲线进行地层划分与对比[J]. 石油仪器,2008,22(5):46-47.
 - [14] 孙 湘,周大为,张希望. 惯性权重粒子群算法模型收敛性分析及参数选择[J]. 计算机工程与设计,2010,31(18):4068-4071.
 - [15] 刘志雄,梁 华. 粒子群算法中随机数参数的设置与实验分析[J]. 控制理论与应用,2010,27(11):1489-1496.
- +++++
- (上接第 36 页)
- [2] Cheng Y, Church G M. Biclustering of expression data [C]//Proc of ISMB. [s. l.]:[s. n.],2000:93-103.
 - [3] Gan X, Liew A, Yan H. Discovering biclusters in gene expression data based on high-dimensional linear geometries [J]. BMC Bioinformatics,2008,9(1):209-210.
 - [4] Liu S, Chen Y, Yang M, et al. Biclustor algorithm and used in market analysis [C]//Proc of second international workshop on knowledge discovery and data mining. Moscow: IEEE,2009:504-507.
 - [5] de Franca F O, Bezerra G, von Zuben F J. New perspectives for the biclustering problem [C]//Proc of IEEE congress on evolutionary computation. Vancouver, BC: IEEE,2006:753-760.
 - [6] Han Lixin, Yan Hong. A fuzzy biclustering algorithm for social annotations [J]. Journal of Information Science,2009,35(4):426-438.
 - [7] 张 敏,戈文航. 双聚类研究与进展[J]. 微型机与应用,2012,31(4):4-6.
 - [8] Madeira S C, Oliveira A L. Biclustering algorithms for biological data analysis: a survey [J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics,2004,1(1):24-45.
 - [9] Zhao Hongya, Liew Alan W C, Xie Xudong, et al. A new geometric biclustering algorithm based on the Hough transform for analysis of large-scale microarray data [J]. Journal of Theoretical Biology,2008,251(2):264-274.
 - [10] 向小军,高 阳,商 琳,等. 基于 Hadoop 平台的海量文本分类的并行化[J]. 计算机科学,2011,38(10):184-188.
 - [11] 蔡佳佳,李名世,郑 锋. 多核微机基于 OpenMP 的并行计算[J]. 计算机技术与发展,2007,17(10):87-91.
 - [12] 吴建章,韩立新,曾晓勤. 一种基于多核微机的闭频繁项集挖掘算法[J]. 计算机应用与软件,2013,30(3):44-46.
 - [13] Wei Shen, Xie Chengjun, Liu Guixia, et al. A novel biclustering with parallel genetic algorithm [C]//Proc of IEEE international conference on human health and biomedical engineering. [s. l.]:[s. n.],2011:900-903.
 - [14] 孙丰荣,刘积仁. 快速霍夫变换算法 [J]. 计算机学报,2001,24(10):1102-1109.
 - [15] Liu B, Wai C, Wang D, et al. Design exploration of geometric biclustering for microarray data analysis in data mining [J]. IEEE Transactions on Parallel and Distributed Systems,2014,25(10):2540-2550.
 - [16] 卢风顺,宋君强,银福康,等. CPU/GPU 协同并行计算研究综述[J]. 计算机科学,2011,38(3):5-9.

一种面向社会化标签的并行双聚类算法

作者: [孙闻](#), [韩立新](#), [SUN Wen](#), [HAN Li-xin](#)
作者单位: [河海大学 计算机与信息学院, 江苏 南京, 211100](#)
刊名: [计算机技术与发展](#) 
英文刊名: [Computer Technology and Development](#)
年, 卷(期): 2015(5)

引用本文格式: [孙闻](#). [韩立新](#). [SUN Wen](#). [HAN Li-xin](#) [一种面向社会化标签的并行双聚类算法](#)[期刊论文]-[计算机技术与发展](#) 2015(5)