一种基于 MapReduce 的知识聚类与统计机制

徐小龙* 李永萍
(南京邮电大学计算机学院 南京 210003)

摘 要：网络文献知识库中的海量资源及其分类的粗粒度，导致学习者容易在文献检索和阅读过程出现认知迷航和知识过载问题。该文提出一种基于 MapReduce 的知识聚类与统计机制：首先，提出基于 MapReduce 的共现矩阵构建算法 MR-CoMatrix；其次，将共现矩阵与相似度系数结合构建相似度矩阵；然后，通过 Z Scores 对相似度矩阵进行标准化；最后，使用离差平方和法(Ward's method)对相似度矩阵进行聚类，生成树状的知识聚类谱系图；基于聚类结果，提出基于 MapReduce 的知识文献统计算法 MR-Statistics，对每个分类的知识属性进行统计。实验结果表明：将 MR-CoMatrix 和 MR-Statistics 方法应用于网络文献知识库进行知识聚类和统计，达到较理想的聚类精度和计算效率，实现了细粒度知识聚类和多维统计，同时减少了时间开销。

关键词：数据挖掘；聚类；知识；共现矩阵；统计；MapReduce
中图分类号： TP311.5 **文献标识码**： A **文章编号**：1009-5896(2016)01-0202-07
DOI: 10.11999/JEIT150247

Knowledge Clustering and Statistics Based on MapReduce

XU Xiaolong LI Yongping
(*College of Computer*, *Nanjing University of Posts and Telecommunications*, *Nanjing* 210003, *China*)

Abstract: The large scale and the coarse classification granularity of resources in literature knowledge bases lead to disorientation and overloading when learners retrieve and read papers. This paper proposes a mechanism of knowledge clustering and knowledge statistics based on MapReduce. Firstly, this paper presents a Co-occurrence Matrix building algorithm based on MapReduce (MR-CoMatrix). Secondly, it makes combination of the co-occurrence matrix and similarity coefficient to build the similarity matrix. Thirdly, the similarity matrix is standardized with Z scores. Finally, knowledge clusters are constructed with the Ward's method. After knowledge clustering, this paper introduces a knowledge Statistics algorithm based on MapReduce (MR-Statistics) to dig the hidden information in each cluster. The experimental results show that the literature knowledge base with MR-CoMatrix and MR-Statistics can realize the accurate and fine clustering, multi-dimension statistics, computational efficiency, and less cost of time.

Key words: Data mining; Cluster; Knowledge; Co-occurrence matrix; Statistics; MapReduce

1 引言

目前国内外的网络文献知识库系统均聚集了海量的知识文献，为科技工作者提供了快速查阅国内外科技文献，进行高层次知识学习的平台。然而网络文献知识库中海量资源分类的粗粒度，导致学习者容易在文献检索和阅读过程出现认知迷航(disorientation)和知识过载(overloading)问题。以中国知识资源总库(China National Knowledge Infrastructure, CNKI)为例，系统主要是按照学科门类进行分类，而没有按照学科下属的知识领域进行更细粒度的分类。通过知识聚类将知识对象按其属性进行整合与统计，不但可以更细致地对文献进行分类，而且可以揭示知识发展规律与知识间联系等潜在的有价值信息[1]。

高效的聚类算法是实现知识聚类与分类导航的关键[2−4]。聚类集成技术[5,6]对聚类成员的结果再进行聚类：首先得到对象之间的共现矩阵(co-occurrence matrix)或相似度矩阵(similarity matrix)，然后使用聚类算法对矩阵进行聚类得到结果。合理地构造共现矩阵和相似度矩阵成为提高聚

收稿日期：2015-02-12；收回日期：2015-10-08；网络出版：2015-11-17
*通信作者：徐小龙 xuxl@njupt.edu.cn
基金项目：国家自然科学基金(61202004, 61472192)，教育部科技发展中心网络时代的科技论文快速共享专项研究(2013116)，江苏省高校自然科学研究计划(14KJB520014)
Foundation Items: The National Natural Science Foundation of China (61202004, 61472192), The Special Fund for Fast Sharing of Science Paper in Net Era by CSTD (2013116), The Natural Science Fund of Higher Education of Jiangsu Province (14KJB520014)

类算法精确度的关键。STREHL 等运用图划分算法[7]对相似度矩阵聚类；基于相似矩阵的谱聚类算法对相似矩阵进行变换得到拉普拉斯矩阵，然后对其特征向量进行聚类[8−11]；徐森等人[6]基于文献[12]设计了基于相似度矩阵的聚类集成谱算法。CHENG 等人[13]通过 Pajek 构建关键词共现矩阵，进行分层聚类。

共现矩阵的构造过程比较复杂。文献[14]采用传统谱聚类中的方法构造相似矩阵，但是没有充分利用样本点分布特征隐含的先验信息，构造效果不够理想；文献[15]通过向量构建、矩阵转置运算、矩阵相乘运算来构造矩阵，文献[16]通过迭代使用贪心算法寻找满足条件的非周期性相关系数来搜索最优的向量，构造(0,1)编码矩阵。如果矩阵的规模太大不适合全部放入内存时，在单机上执行任务将非常缓慢甚至难以实现。因此，对大规模文献进行知识聚类需要更为有效的共现矩阵构建方法以及具有强大处理和存储能力的分布式计算平台。

针对上述问题，本文对网络文献知识库系统中的知识文献聚类与统计机制展开研究，并引入 MapReduce 分布式处理模型，利用集群计算系统实现对知识的高效聚类和统计，主要贡献包括：

(1)提出基于 MapReduce 的共现矩阵构建算法(MapReduce-based Co-occurrence Matrix building algorithm, MR-CoMatrix)，利用 MapReduce 统计知识属性两两共现的频次，并与哈希图结合构建共现矩阵，解决了单个计算节点有限内存难以存储与处理大矩阵导致的无法统计或统计效率降低等难题。

(2)提出基于相似度矩阵实现对知识的聚类的方法，首先通过 Z Scores 对相似度矩阵进行标准化，然后使用离差平方和法对相似度矩阵进行知识聚类，高效生成树状的知识聚类谱系图。

(3)提出基于 MapReduce 的知识文献统计算法(MapReduce-based knowledge Statistics, MR-Statistics)，在知识聚类后，对每一个类簇中的知识属性进行统计，以挖掘出有价值的信息。

2　基于 MapReduce 的共现矩阵构建方法

2.1 MapReduce

MapReduce[17−21] 作为能够对大数据集进行运算的分布式并行编程模式，适用于数据密集型自动并行化计算。MapReduce 将大规模数据集的操作任务分解后分发给主节点管理的各计算节点执行，最后将结果进行汇总。MapReduce 的作业执行过程主要分为映射(Map)和归约(Reduce)两个步骤，“映射”负责把任务分解成多个任务，“归约”负责把分解后多个任务处理的结果进行汇总。

2.2 知识对象建模

通常每一篇文献都有标题、作者、关键词、发表时间等属性，本文对文献及其属性进行建模，用知识对象来表示科学文献，用知识属性来表示文献的属性。根据以上的表述，每个对象实例包含 6 个知识属性，用一个 6 元组来表示知识文献模型：

$$P=(T,A,S,I,K,Y) \tag{1}$$

式(1)中，T 代表标题，A 代表作者，S 代表作者所在的高校或者研究所，I 代表发文机构，K 代表关键词，Y 代表发表时间。文献对象以式(1)的 6 元组形式写入文本文件，供 MapReduce 处理。

2.3 MR-CoMatrix 算法

在“映射”阶段，从文本集 P 中提取待统计的属性 K；然后，由于每一个属性 K 对应一篇文献的全部关键词，因此通过 IK Analyser 对 K 进行中文分词并作为“键”。MapReduce 确保相同“键”的所有值都在“归约”阶段“汇聚”，这样，“归约”只需将同一个“键”下的所有“值”累加求和即可得到最终的“键-值”对，即“关键词-词频”。以上的工作由一个单独的子任务(子任务 1)来完成，将其结果保存在关键词文本集中(设为 T1)。K 表示为

$$K=w_1;w_2;\cdots;w_n \tag{2}$$

式中，w 为 K 中的关键词。

由于每一个属性 K 对应一篇文献的全部关键词。在构建共现矩阵之前，先要对文献数据进行预处理，提取其中的属性 K，这由一个单独的子任务(子任务 2)完成。预处理过后，所有的 K 值会输出到文件中，以便于后续子任务处理。第 3 个子任务(子任务 3)以第 2 个子任务的输出为输入。在“映射”阶段，由每行起始位置相对于文件起始位置的偏移量和该行的 K 构成输入的“键-值”对，由共现的词语对和整数 1 作为输出的“键-值”对。整个过程可以有两个嵌套的循环直接完成：外层循环迭代所有的关键词，作为“词对”中左边的那个词，即第 1 个词，内层循环迭代第 1 个词右边的所有词，作为“词对”中右边的那个词。MapReduce 计算框架保证相同“键”下的所有“值”都在“归约”中汇聚。这样，“归约”的任务就是将相同键的所有值求和作为值，输出最终的“键-值”对，每一个键值对都与共现矩阵中的元素对应。由 MapReduce 计算出的共现信息通过哈希图“迁移”到 2 维数组。

“迁移”的过程分为两步进行：首先，哈希图提供映射操作，为每一个关键词映射唯一的整型标签；其次，通过一个循环判断“词对”中的每个单词是否均有标签；最后获取这两个标签作为 2 维数组的下标，并将单词对的共现频次作为数组的值。

通过以上步骤，将 Hadoop 分布式文件系统(Hadoop Distributed File System, HDFS)上的线性信息“迁移”到 2 维数组里。“迁移”过后，得到关键词的共现矩阵。MR-CoMatrix 算法的工作流程如图 1 所示。

步骤 1　子任务 2 从每个知识对象中提取出属性 K 输出到 Hadoop 分布式文件系统作为子任务 3 的输入；

步骤 2　子任务 3 根据子任务 2 的输出信息计算关键词的两两共现频次；

步骤 3　子任务 1 统计关键词的词频；

步骤 4　通过哈希图将共现词对存入 2 维数组。

2.4 相似度矩阵构建

Ochiai系数是一种用于衡量相似度的指标，也称作Ochiai-Barkman相似系数或者Otsuka-Ochiai相似系数。在本文中，两个关键词的相似度被定义为两个关键词的共现频次与这两个关键词单独出现次数的几何平均值之商：

$$O(w_1,w_2)=\frac{n(w_1\cap w_2)}{\sqrt{n(w_1)\times n(w_2)}} \tag{3}$$

式(3)中，w_1, w_2 表示两个关键词，$n(w_1\cap w_2)$表示关键词共现的次数，$n(w_1)$和 $n(w_2)$表示每个关键词单独出现的次数。关键词的共现矩阵表示为 $\boldsymbol{M}[w_1,w_2]$。

由共现矩阵 $\boldsymbol{M}[w_1,w_2]$和 Ochiai 相似度系数就可以计算出相似度矩阵 $\boldsymbol{SM}[w_1, w_2]$：

$$\boldsymbol{SM}[w_1,w_2]=\frac{\boldsymbol{M}[w_1,w_2]}{\sqrt{\boldsymbol{M}[w_1,w_1]\times\boldsymbol{M}[w_2,w_2]}} \tag{4}$$

3 知识聚类与统计

3.1 基于相似度矩阵的知识聚类

本文基于 Z Scores[22]对聚类的数据进行标准化，选择离差平方和法(Ward's method)对相似度矩阵进行聚类。Z Scores 是一种对数据标准化处理的方法，将原始数据通过标准偏差公式进行标准化处理，即样本的原始数据 Rs(raw score)和平均数据 Ms (mean score) 之差与标准偏差 Sd(standard deviation)的商。Z Scores 的定义式为

$$Z=(\mathrm{Rs}-\mathrm{Ms})/\mathrm{Sd} \tag{5}$$

将关键词之间的相似度进行标准化处理，可以消除数值相差大造成的负面影响，使得聚类效果稳定可靠。样本的原始数据即为知识属性的相似度。知识属性的相似度矩阵 $\boldsymbol{SM}$ 表示为

$$\boldsymbol{SM}=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-11} & x_{n-12} & \cdots & x_{n-1n} \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix} \tag{6}$$

其中，$\boldsymbol{SM}$ 的每一行$[x_{n1}\cdots x_{nn}]$ 为一个知识属性的相似度向量，其中的元素 x 表示该知识属性与所有知识属性的相似度，则每一个向量的相似度均值为

$$\overline{x}_i=\frac{1}{n}\sum_{j=1}^{n}x_{ij} \tag{7}$$

标准差表示为

$$S_i=\sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(x_{ij}-\overline{x}_i)^2} \tag{8}$$

基于知识属性相似度的 Z Scores 标准化为

$$x_{ij}^{*}=\frac{x_{ij}-\overline{x}_i}{S_i} \tag{9}$$

凝聚聚类(agglomerative clustering)在生成聚类谱系图时实际使用较多，离差平方和法是一种基于经典的平方和准则的凝聚聚类方法，在实际应用中分类效果较好[23]。离差平方和法采用层次聚类的

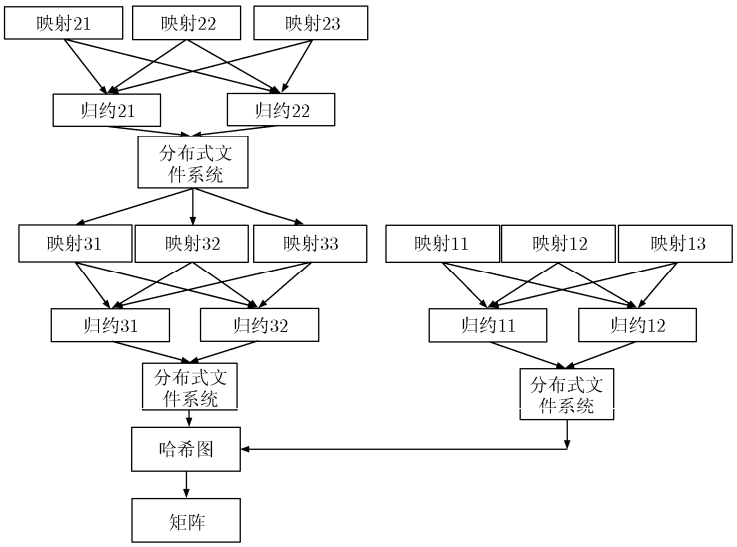

图 1　MR-CoMatrix 算法流程图

思想来合并簇，基本思想是：首先，训练集中的每一个对象都是一个簇；然后，不断计算簇与簇之间的距离，合并距离最近的两个簇；最终，所有的对象汇聚在一个簇中[24]。两个簇之间的距离定义为[25]

$$d_W(C_1,C_2)=\frac{n_1n_2}{n_1+n_2}\|\overline{x}-\overline{y}\|_2^2 \qquad (10)$$

式(10)中，d_W 表示两个簇间的距离，$C_1=\{x_i\}_{i=1}^{n_1}$ 代表一个簇，$C_2=\{y_j\}_{j=1}^{n_2}$ 代表另一个簇，n_1 和 n_2 表示每个簇中的对象个数，$\overline{x}$ 和 $\overline{y}$ 表示两个簇的质心，$||\cdot||_2$ 表示欧几里得距离。M 的每一行 $[x_{n1}x_{n2}\cdots x_{nn}]$ 为一个知识属性的相似度向量，即每一个知识属性均由一个向量表示，$||\cdot||_2$ 由向量间的欧氏距离得出。

3.2 MR-Statistics

聚类后对每一个分类进行知识统计，可以获得更多的规律性知识。以关键词的聚类为例，对每一个分类中的高频关键词出现过的文献的发文量进行统计，可以分析出该关键词的发展情况，从而判断出该主题的研究现状和发展趋势。

MR-Statistics 的处理流程如图 2 所示：在“映射”之前，先通过一个解析类来解析每一个论文实例 P，得到待统计的知识属性。“映射”阶段首先定义了若干个计数器，然后通过解析类的对象解析 P 中的属性，接着对解析的属性字段进行判断，如果该属性的值符合条件，则相应的计数器值加“1”；“归约”阶段遍历所有分片数据列表，合并列表上的相同计数器的值，最后返回所有计数器的对象。

4　实验验证与分析

4.1 实验环境

我们在实验室局域网环境中搭建多节点集群测试环境。计算节点的操作系统均为 CentOS 6.5，内核版本为 Linux 2.6.32-431.el6.i686 GNOME 2.28.2; CPU 为 Pentium(R) Dual-Core CPU E5200@2.50 GHz，内存为 1 GB，硬盘为 160 GB; Hadoop 的版本为 2.4.1, JDK 的版本为 jdk1.7.0_45。

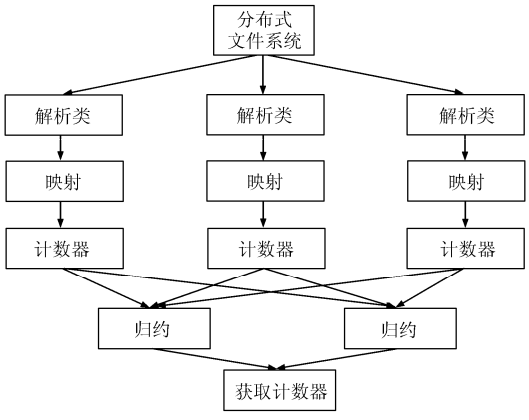

图 2　MR-Statistics 流程图

4.2 实验与性能分析

本文首先在相同数据源的情况下，将本文提出的 MR-CoMatrix 算法和 MR-Statistics 算法应用于单节点与多节点集群环境中进行对比实验，获得的实验结果如图 3 和图 4 所示。

由图3可以看出，随着文件大小规模的递增，不管是单节点还是多节点计算环境，算法运行的时间也随之增加；当输入数据规模较小时，单节点和多节点的运行时间相差很小，表明利用集群对于小规模数据进行聚类和统计算法并没有性能优势；当文件大小达到一定规模并进一步扩大时，两者的差距开始逐渐拉大，单节点下算法的计算时间迅速增大，而多节点下算法的计算时间则是平缓增加，说明基于MapReduce的集群计算平台，算法的数据处理能力随着数据规模的递增逐渐增强，达到预期效果。

由图4可以看出，当数据规模较小时，在单节点还是多节点计算环境中算法性能相近；随着文件规模增大，单节点还是多节点环境下，算法的运行时间也随之递增，当文件大小达到一定规模并进一步扩大时，两者的差距开始逐渐拉大，单节点下算法计算时间迅速增大，增幅比较明显，而多节点下算法的计算时间增幅则比较平缓，这同样说明引入MapReduce和集群计算环境对大规模文献聚类和统计处理性能的提升优势。

第2组实验的目的是观察在单节点和多节点环境下算法计算速度与数据规模之间的关系，实验结果如图5，图6所示。从图中可以看出，随着知识对象的增加，计算速度并不是一直随之提升。开始时，随着数据的增加，不管是单节点还是多节点，计算速度都在提高，并且多节点计算速度的上升幅度明显比单节点要快，但当知识对象的条数达到一定规模时，单节点的计算速度达到一个最大值后开始下降，多节点的速度提升也变得缓慢。分析原因，主要是因为对于单节点，系统的所有资源都是有限制的，包括CPU资源和内存资源：开始的时候，CPU和内存都有空闲，所以节点的计算能力会随着数据量的增大而提升；但是，当CPU和内存的资源不断地被占用，达到饱和状态后，系统资源不能及时得到合理分配，此时就会耗费大量的开销在资源和数据的分配上，导致节点的计算速度下降；对于多节点集群来说，系统资源相对来说较为充裕，因此计算速度也会相应地增强，但是随着数据量的增多，上升速度也会开始变得平缓。

第3组实验的目的是观察多节点情况下不同算法的性能表现。本文在多节点集群环境下将MR-CoMatrix与Pairs和Stripes进行了实验和对比分析；

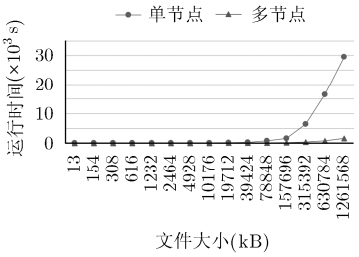

图 3 共现矩阵构建算法单节点与多节点实验结果对比图

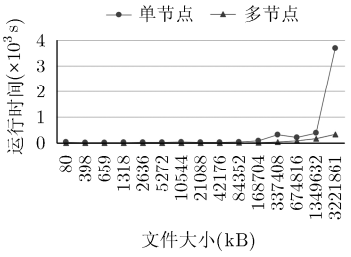

图 4 知识文献统计算法单节点与多节点实验结果对比图

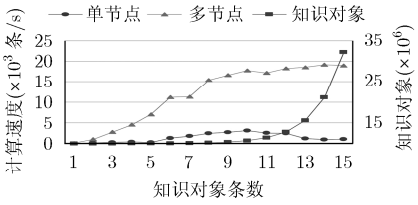

图 5 共现矩阵构建算法的实验结果图

并将MR-Statistics算法与朴素WordCount进行实验和对比分析，实验结果如图7和图8所示。

从图7的实验结果可以看出，MR-CoMatrix算法在时间开销性能上稍慢于Stripes算法，但优于Pairs算法。Stripes算法利用关联数组来存放每一个词语的同现信息，然后再将所有相同键的关联数组到“归约”阶段一起处理，这样做的好处是为组合器提供了更多的机会来聚集中间结果，因此在网络传输上减少了优化和排序的时延，从而减少了算法的运行时间。MR-CoMatrix算法的处理模式会生成很多的“键-值”对，同一个键的所有值在“归约”阶段汇总，“归约”任务是对每一个键值对进行统计，通过复杂的键进行分布式的计算自然没有批处理的效率高。但是，从算法的空间复杂度和可扩展性来说，由于Stripes算法使用关联数组，存在更多的序列化和反序列化操作，因此算法复杂度高于MR-CoMatrix算法；而且由于Stripes算法产生的键值对大小依赖整个数据集，当数据集很大时，产生的键值对会非常复杂，不便于算法的扩展。从图8的实验结果可以看出，当数据规模比较小时，WordCount在时间上的开销与本文提出的MR-Statistics性能相近，但是其时间增长率 $\Delta(t)$ 大于MR-Statistics；随着数据规模的逐渐增大，MR-Statistics算法的运行时间明显优于WordCount。这主要是因为MR-Statistics维护了MapReduce内置的计数器特性，减少了“归约”的工作过程，从而提高了计算效率。

具体而言，MapReduce 的原理决定了每一个计数器的值都是完整传输，而不是增量传输，当一个子任务完成后，MapReduce 框架将基于“映射”和“归约”来统计这些计数器，产生一个最终的结果。在 MR-Statistics 算法中，由于“归约”的工作被简化，因此算法的运行时间主要取决于“映射”阶段的数据分片和计数器的传输；数据规模较小时，“映射”阶段的数据分片消耗了较多的时间，也影响了计数器的传输，但随着数据规模增大，“映射”在处理大数据的优势开始发挥作用，而算法本身的开销主要集中在“映射”阶段，因此提高了算法效率。

4.3 应用效果

本文将MR-CoMatrix和MR-Statistics方法应用于网络文献知识库进行知识聚类和统计，生成的树状的知识聚类谱系图如图9所示。本文的数据来源是2009年～2014年发表的4195篇云计算领域的文献，共计7622个关键词，限于篇幅，本文只选取排名前50位的高频关键词，基于本文的方法生成50×50的相似度矩阵。

由于树状图中包含的关键词较多，本文将50个关键词分成了8个簇：

Cluster1: MapReduce, Hadoop, 并行计算，HBase，大数据，数据挖掘，数据分析；Cluster2：任务调度，粒子群算法，负载均衡，资源调度，蚁群算法，资源分配，遗传算法；Cluster3：云存储，数据安全，隐私保护，访问控制；Cluster4：分布式文件系统，HDFS，智能电网，调度，算法，数据处理，、数据存储，分布式计算；Cluster5：云服务，云制造，物联网；Cluster6：图书馆，信息服务，云计算技术，大数据时代；Cluster7：云安全，信息安

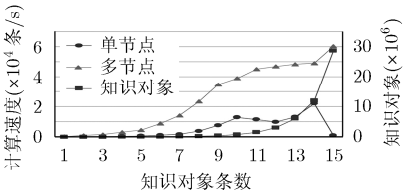

图 6 知识文献统计算法实验结果图

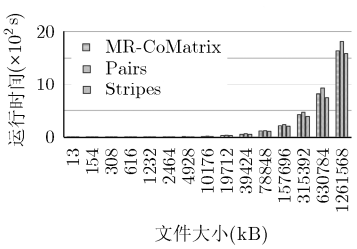

图 7 共现矩阵构建算法实验结果对比图

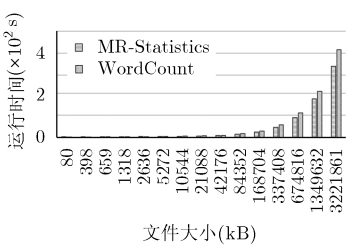

图 8 知识文献统计算法实验结果对比图

全，分布式，云平台，服务质量，聚类，私有云，电子商务，数字图书馆，知识服务，资源共享；Cluster8: SaaS, IaaS，云计算，虚拟化，虚拟机，数据中心。

对聚类结果分析可知：首先，被聚在一起的关键词彼此间存在一些价值联系，距离较近的簇间存在某些相似性，比如Cluster1内部的关键词都是与Hadoop相关的研究和应用，Cluster1和Cluater2之间的研究主题存在相关性，显示出目前很多算法基于Hadoop平台来改进算法在大数据处理方面的性能；其次，距离较远的关键词在目前的研究中相关性不是很大，但也不是绝对没有相关性，比如Cluster1和Cluster6的距离较远，只说明两个分类研究的侧重点不同，不能说二者之间没有关系。从效果看，聚类结果符合目前云计算领域的研究现状，说明根据本文算法生成的相似度矩阵比较符合实际的研究情况。以Cluster1中的关键词MapReduce为例，运用MR-Statistics对包含该词的论文的年度发文量进行统计分析，2009年～2014年间MapReduce主题的相关论文呈逐年上升趋势，尤其是从2012年后，几乎是线性增长，并且上升得非常迅速，说明我国的科研人员在云计算领域的研究已经逐步从概念性的探索阶段发展到运用具体的应用阶段；另一方面也说明云计算领域的研究进入快速发展时期，在未来一段时间都将是计算机领域的一个研究热点。

5　结束语

本文引入了MapReduce分布式处理模型，首先提出基于MapReduce的共现矩阵构建算法MR-CoMatrix；其次，基于聚类结果，提出基于MapReduce的知识对象统计算法MR-Statistics，对每一个分类中的知识属性进行统计；最后将提出的MR-CoMatrix和MR-Statistics应用于网络知识系统，实现了细粒度知识聚类和多维知识统计，在计算精度、计算效率和时间开销上均有良好的性能表现。但是，MR-CoMatrix算法在计算效率上仍然有值得提升的空间，这也是后续研究中值得关注的地方。

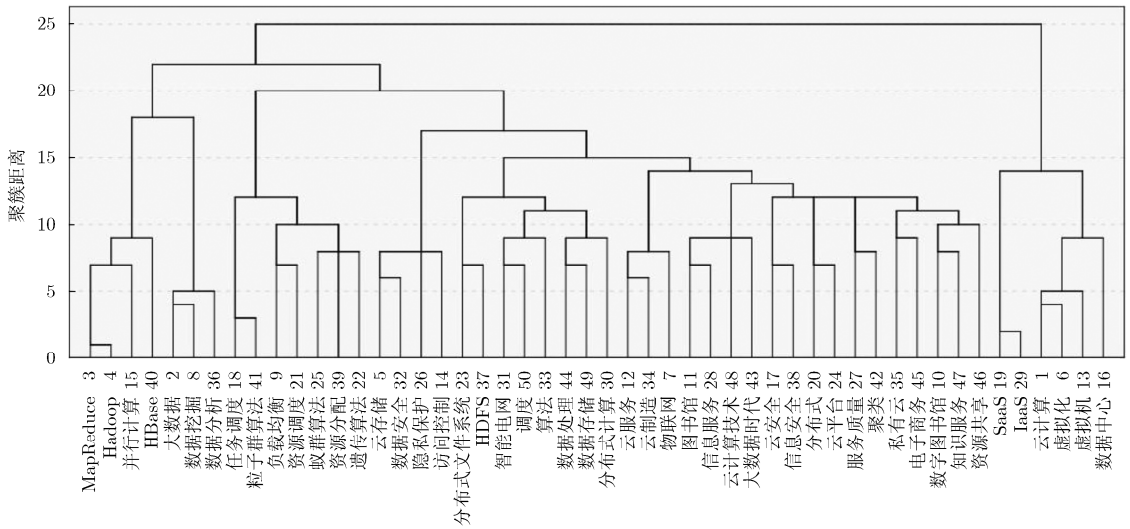

图 9 云计算领域知识聚类谱系图

参 考 文 献

[1] SERET A, VERBRAKEN T, and BAESENS B. A new knowledge-based constrained clustering approach: theory and application in direct marking[J]. *Applied Soft Computing*, 2014, 24(3): 316–327.

[2] 朱林, 雷景生, 毕忠勤, 等. 一种基于数据流的软子空间聚类算法[J]. 软件学报, 2013, 24(11): 2610–2627.
ZHU Lin, LEI Jingsheng, BI Zhongqin, *et al.* Soft subspace clustering algorithm for streaming data[J]. *Journal of Software*, 2013, 24(11): 2610–2627.

[3] ZHU Lin, CHUNG Fulai, and WANG Shitong. Generalized fuzzy C-means clustering algorithm with improved fuzzy partitions[J]. *IEEE Transactions on Systems, Man, and Cybernetics*, 2009, 39(3): 578–591.

[4] 张敏, 于剑. 基于划分的模糊聚类算法[J]. 软件学报, 2004, 15(6): 858–866.
ZHANG Min and YU Jian. Fuzzy partitional clustering algorithms[J]. *Journal of Software*, 2004, 15(6): 858–866.

[5] 徐森, 周天, 于化龙, 等. 一种基于矩阵低秩近似的聚类集成算法[J]. 电子学报, 2013, 41(6): 1219–1223.
XU Sen, ZHOU Tian, YU Hualong, *et al.* Matrix low rank approximation-based cluster ensemble algorithm[J]. *Acta Electronica Sinica*, 2013, 41(6): 1219–1223.

[6] 徐森, 卢志茂, 顾国昌. 使用谱聚类算法解决文本聚类集成问题[J]. 通信学报, 2010, 31(6): 58–66.
XU Sen, LU Zhimao, and GU Guochang. Spectral clustering algorithm for document cluster ensemble problem[J]. *Journal on Communications*, 2010, 31(6): 58–66.

[7] ZHU Wenxing, CHEN Jianli, and LI Weiguo. An augmented Lagrangian method for VLSI global placement[J]. *The Journal of Supercomputing*, 2014, 69(2): 714–738.

[8] ZHOU F, TORRE F D L, and HODGINS J K. Hierarchical aligned cluster analysis for temporal clustering of human motion[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35(3): 582–596.

[9] MASHSHI S, NIU G, MAKOTO Y, *et al.* Information-maximization clustering based on squared-loss mutual information[J]. *Neural Computation*, 2014. 26(1): 84–131.

[10] YU Feili, CAO Liangliang, FERIS R S, *et al.* Designing Category-level attributes for discriminative visual recognition [C]. Preceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, 2013: 771–776.

[11] 李建元, 周脚根, 关佶红. 谱图聚类算法研究进展[J]. 智能系统学报, 2011, 6(5): 405–414.
LI Jianyuan, ZHOU Jiaogen, and GUAN Jihong. A survey of clustering algorithms based on spectra of graphs[J]. *CAAI Transactions on Intelligent Systems*, 2011, 6(5): 405–414.

[12] LU Zhimao and ZHANG Qi. Clustering by data competition [J]. *Science China* (*Information Sciences*), 2013, 56(1): 1–13.

[13] CHENG Bo, WANG Minhong, MϕRCH A I, *et al.* Research on e-learning in the workplace 2000–2012: A bibliometric analysis of the literature[J]. *Educational Research Review*, 2013, 11: 56–72.

[14] 孔万增, 孙志海, 杨灿. 基于基本间隙与正交特征向量的自动谱聚类[J]. 电子学报, 2010, 38(8): 1880–1891.
KONG Wanzeng, SUN Zhihai, and YANG Can. Automatic spectral clustering based on eigengap and orthogonal eigenvector[J]. *Acta Electronica Sinica*, 2010, 38(8): 1880–1891.

[15] CARPENTIER S, SOLE A D, and KAC V G. Rational matrix pseudodifferential operators[J]. *Selecta Mathematica*, 2014, 20(2): 403–419.

[16] JUGL E, KUHWALD T, and IVERSEN K. Algorithm for construction of (0,1)-matrix codes[J]. *Electronics Letters*, 1997, 33(3): 226–229.

[17] 李建江, 崔健, 王聃, 等. MapReduce 并行编程模型研究综述[J]. 电子学报, 2011, 39(11): 2635–2642.
LI Jianjiang, CUI Jian, WANG Dan, *et al.* Survey of MapReduce parallel programming model [J]. *Acta Electronica Sinica*, 2011, 39(11): 2635–2642.

[18] FERRERA P, PRADO I D, PALACIOS E, *et al.* Tuple MapReduce and pangool: an associated implementation[J]. *Knowledge and Information Systems*, 2014, 41(2): 531–557.

[19] 陈吉荣, 乐嘉锦. SingleMapReduce：单一输出 HDFS 文件的 MapReduce 编程模型[J]. 华南理工大学学报, 2014, 42(5): 135–142.
CHEN Jirong and LE Jiajin. SingleMapReduce: a MapReduce programming model based on single output file of HDFS[J]. *Journal of South China University of Technology*, 2014, 42(5): 135–142.

[20] 王肇国, 易涵, 张为华. 基于机器学习特性的数据中心能耗优化算法[J]. 软件学报, 2014, 25(7): 1432–1447.
WANG Zhaoguo, YI Han, and ZHANG Weihua. Power saving based on characteristics of machine learning in data center[J]. *Journal of Software*, 2014, 25(7): 1432–1447.

[21] 易小华, 刘杰, 叶丹. 面向 MapReduce 数据处理流程开发方法[J]. 计算机科学与探索, 2011, 5(2): 161–168.
YI Xiaohua, LIU Jie, and YE Dan. Development method of MapReduce oriented data flow processing[J]. *Journal of Frontiers of Computer Science and Technology*, 2011, 5(2): 161–168.

[22] ROWBERRY J. Z Scores[M]. New York: Springer Science + Business Media, 2013: 3419–3420.

[23] VARIN T and BUREAU R. Clustering files of chemical structures using the Szekely-Rizzo generalization of Ward's method[J]. *Journal of Molecular Graphics and Modelling*, 2009, 28(2): 187–195.

[24] LEE A. Minkowski generalizations of Ward's method in hierarchical clustering[J]. *Journal of Classification*, 2014, 31(2): 194–218.

[25] MURTAGH F and LEGENDRE P. Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion?[J]. *Journal of Classification*, 2014, 31(3): 274–295.

徐小龙： 男，1977 年生，教授，研究方向为计算机软件、分布式计算、信息安全、移动计算技术等.

李永萍： 女，1989 年生，硕士，研究方向为知识系统、数据挖掘技术等.