基于MapReduce的Web标签SOINN聚类算法

王 洁 于颜硕 周宽久 侯 刚
(大连理工大学嵌入式系统工程系 大连 116620)

摘 要 Web标签有助于用户根据自己特定的兴趣完成信息资源的分类、组织和检索。然而,正是由于协同标记系统特有的公开性、自由化的特点,采用其对信息资源进行描述、组织、分类和检索,存在着信息描述不精确、标签组织混乱和标签语意模糊等问题。在此背景下提出了3种基于特征向量表示法(FVR)的Web标签SOINN聚类算法:基于资源的特征向量表示法、基于其他共现标签的特征向量表示法和基于全集共现标签的特征向量表示法。同时应用MapReduce框架将SOINN算法进行并行化。实验表明,当类中心数量超过2000时,3种分布式聚类FVR算法的召回率和准确度优于原始算法,可获得很好的加速比。从而证明此分布式聚类算法具有很好的可扩展性,可以用于更为海量的Web日志聚类分析系统。

关键词 Web标签聚类,SOINN算法,MapReduce
中图法分类号 TP302 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.12.043

MapReduce-based SOINN Clustering Algorithm for Web Tag

WANG Jie YU Yan-shuo ZHOU Kuan-jiu HOU Gang
(Department of Embedded Systems Engineering, Dalian University of Technology, Dalian 116620, China)

Abstract Web tag helps users to classify, organize and search internet resources according to their interests. Tag clustering can help to solve problems caused by openness and freedom of Web tag system, such as inaccurate information description, disorganized tags, ambiguity, and so on. Three tag feature vector representation (FVR) methods were presented which are resource-based FVR, other tag co-occurrence FVR and total tag co-occurrence FVR, can all apply to SOINN clustering algorithm. SOINN clustering can be parallelized by MapReduce model. Experiments show that accuracy and recall rate of three tag FVR are superior to original tag co-occurrence FVR and tag SOINN clustering by MapReduce owns optimum performance when the number of class center is more than 2000. The experimental results prove that distributed clustering algorithms proposed in this paper have good scalability which can be applied to more massive cluster Web tag analysis system.

Keywords Web tag clustering, SOINN algorithm, MapReduce

1 引言

随着计算机网络的普及,因特网已经发展成为一个蕴藏着有用知识的海量信息空间。对于一般互联网用户而言,一个关键的问题是怎样快速、准确地从网上获取有价值的信息,并从中找到自己需要的那部分。另一方面,对于互联网运营商来说,如何合理、有效地分析用户行为,将用户及其感兴趣的话题、网页进行合理聚类,以提供更好的个性化服务,从而提高企业利益。因此如何分析挖掘互联网中海量的用户行为信息,成为近年来互联网研究领域的一个重要课题。Web2.0是一种新兴的高度网络化、自由化的互联网形态,它架构在用户、内容及应用的基础上,因此吸引了大量用户,衍生出诸如社区网络、博客、播客、网络文摘、维基百科等Web2.0类应用[1]。Web标注系统是针对Web2.0环境下新的用户需求而产生的信息组织工具,它带来了全新的信息交流与资源共享方式,为传统的网络信息分类和传播方法带来了新的理念,体现出互联网所推崇的共享与写作精神,开创了互联网信息传播的新阶段。例如del.icio.us、豆瓣网等网站都采用了协同标注,用户可以根据自己的社会文化背景、专门技术和世界观,用不同标签标注资源,并利用这些用户标签完成信息资源的分类、组织和检索。然而,正是由于协同标记系统特有的公开性、自由化的特点,使得采用其对信息资源进行描述、组织、分类和检索[2]存在着信息描述不精确、标签组织混乱和标签语意模糊等问题。针对这些问题,文献[3-7]指出,聚类技术是有效的解决方法之一。

现有标记系统常采用聚类分析技术来解决标签冗余和语意模糊的问题。目前标签聚类算法[8,9]大多根据不同标签在对象中共同出现的次数来计算它们之间的相似度,但是用这

到稿日期:2013-06-25 返修日期:2013-08-29 本文受国家自然科学基金可信软件研究课题(61272174),中央高校基本科研业务费专项基金(DUT14QY32)资助。

王 洁(1979—),男,博士,讲师,主要研究领域为并行计算、FPGA验证,E-mail:wang_jie@dlut.edu.cn;于颜硕(1989—),男,硕士生,主要研究领域为并行计算、FPGA验证,E-mail:yuyanshuo@gmail.com(通信作者);周宽久(1966—),男,博士,教授,主要研究领域为可信嵌入式系统、嵌入式系统应用;侯 刚(1982—),男,博士生,主要研究领域为可信软件、复杂性理论。

种方法聚类的精确度与召回率并不高。此外，在 Web 标记系统中，用户标签是海量的，如何在如此大规模的数据中对用户行为进行挖掘是目前互联网研究领域的一个难点。随着 Google 的 MapReduce[10] 分布式平台的出现，很多单机无法完成的计算任务现在成为了可能，即便是一些复杂度很高的计算也可以在可接受的时间内完成。MapReduce 模型[11] 的基本思想是：将要执行的问题拆解成 map（映射）和 reduce（规约）操作，即先通过 map 程序将数据切割成不相关的区块，分配（调度）给大量计算机处理，以达到分布运算的效果；再通过 reduce 程序将结果汇整，输出开发者所需要的结果。

本文提出了一种新的基于 MapReduce 分布式平台的高效聚类算法，充分考虑标签的标记信息，采用基于对象的特征向量来精确地表征一个标签，利用余弦相似度公式得到较为准确的标签相似度，然后采用 SOINN 算法将用户标签进行聚类。这在很大程度上缓解了标签组织混乱、语义模糊的问题，提升了标签描述的精确性，为用户提供了更好的标签导航和浏览机制。

2 Web 标签的向量表示法

由于 Web 标记系统没有任何限制，采用其对资源进行描述、分类会产生标签组织混乱、语义模糊的问题。对于 Web 上的应用，利用这种混乱的、没有清楚语义的标签来搜索网络资源没有太大价值。因此，解决该问题的关键在于如何从这些混乱的、语义模糊的用户标签中，分析挖掘出其中隐藏的含义，从而使它们可以被用来进行更好的网络资源描述、分类和搜索。针对这些问题，现有研究大多采用的方法是对用户标签进行聚类，重新组织用户标签，这在很大程度上缓解了标签组织混乱、语义模糊的问题，提升了标签描述的精确性，为用户提供了更好的标签导航、浏览机制，但是，它们又存在聚类结果准确性不高的缺点。本章在这个背景下提出了基于特征向量表示法的 Web 标签聚类算法，将标签用一个 N 维的特征向量建模表示，并给出了 3 种不同的特征向量表示方法。

2.1 基于标签共现的聚类算法

为了计算两个标签之间的相似度，最常用的方法是计算两个标签在不同的资源中共同出现的次数[11]。其基本思想就是：如果两个标签在资源中同时出现的次数越多，则它们之间也就越相关。

其数学表达式为：

$$\begin{cases}\mathrm{Sim}(t_i,t_j)=1 (i=j)\\ \mathrm{Sim}(t_i,t_j)=\dfrac{|\{k|b_{k,i}>0\&b_{k,j}>0\}|}{\min(|\{k|b_{k,i}>0\}|,|\{k|b_{k,j}>0\}|)} (i\neq j)\end{cases} \tag{1}$$

其中，$\mathrm{Sim}(t_i,t_j)$ 表示标签 t_i 和 t_j 之间的相似度大小，$b_{k,i}$ 表示资源 r_k 被标签 t_i 标记的次数（即有 $b_{k,i}$ 个用户用标签 t_i 标记了资源 r_k）。根据采用的定义方法，当两个标签标记了相同资源或者一个标签标记的资源是另一个标签标记资源的子集时，这两个标签的相似度就为 1。

利用式(1)，可以计算所有标签两两之间的相似度，进而得到标签相似度矩阵：$TT\{1,2,\cdots,|T|\}\times\{1,2,\cdots,|T|\}$，矩阵中元素的值即为式(1)中的 $\mathrm{Sim}(t_i,t_j)$，$|T|$ 为标签总数。在此基础上利用聚类算法对标签进行聚类。尽管这种算法缓解了标签组织混乱、语义模糊等问题，但是它没有对单个标签进行建模，即没有建立一个统一的数学表达式来表征单个标签，仅能衡量标签两两之间的相似程度，其相似度计算的精确性并不高，本文的聚类实验结果也验证了这一点。

2.2 基于资源的特征向量表示法

一个资源一般由几个标签进行标注，每个标签与该资源间都存在着一定的关系，利用这种关系，我们可以反向思考，一个标签也可以由与其相关的资源来表示。基于这种思想，利用被当前标签标记过的资源构成所需要的标签特征向量。

对在标签集合 T 中的任意一个标签有：

$$V_{t_i}^{item}=[k_{t_i,1},k_{t_i,2},\cdots,k_{t_i,|R|}] \tag{2}$$

$$\begin{cases}k_{t_i,j}=0, & \text{if } t_i\cap r_j=\varnothing\\ k_{t_i,j}=b_{j,t_i}, & \text{else}\end{cases} \tag{3}$$

其中，R 表示整个资源集合，$|R|$ 表示资源的总数，b_{j,t_i} 在式(1)中有过说明，式(3)表示如果标签 t_i 未在资源 r_j 中出现，则 $k_{t_i,j}=0$；否则，$k_{t_i,j}$ 为标签 t_i 在资源 r_j 中出现的次数。

观察该特征向量，分析发现，当资源数目很多时，$V_{t_i}^{item}$ 的维数将会非常高；另一方面，一个标签真正标记过的资源数却比较少（这在实际情况中很常见，如在豆瓣网中，其资源数会达到亿量级，而一个标签标记过的资源却仅有万量级；在后文的实验中，资源的总数为 300000，而出现频率最高的标签标记过的资源总数仅为 7000 左右），造成该特征向量的极度稀疏。利用这种稀疏性，我们可以大大减少该向量的存储空间，降低向量间相似度计算的时间复杂度，将算法的时间和空间开销控制在可接受范围内。

2.3 基于其他共现标签的特征向量表示法

一个标签可以与其他一些标签共同标记一个资源，这种关系表明该标签与和它一起出现过的其他标签间存在着一定的相关性。基于这种思想，本文利用共同出现过的其他标签所组成的特征向量来表征当前标签，标签 $V_{t_i}^{tag}$ 有：

$$V_{t_i}^{tag}=[c_{t_i,1},c_{t_i,2},\cdots,c_{t_i,|T|}] \tag{4}$$

$$\begin{cases}c_{t_i,j}=0, & \text{if } i=j\\ c_{t_i,j}=|\{k|b_{k,i}>0\&b_{k,i}>0\}|, & \text{if } i\neq j\end{cases} \tag{5}$$

其中，$|T|$ 为标签的总数，式(5)表示标签 t_i 的特征向量元素值为其他标签与 t_i 共同出现的次数；由于是由其他标签来表征 t_i，因此特征向量第 i 项元素值为 0。

对比基于资源的特征向量表示法算法，该算法也具有与基于资源的特征向量表示法相同的特点——特征向量非常稀疏，这就保证了聚类操作的时间和空间复杂度也在可接受范围内。

2.4 基于其他共现标签的特征向量表示法

该表示方法是对基于其他共现标签表示法的一种优化。我们考虑以下情形，假设有两个标签 A 和 B，它们出现并且仅共同出现了 100 次，也就是说没有再和其他标签共同出现过。按照前一种表示法，A 的特征向量为[0,100]，B 的特征向量为[100,0]，它们是正交的，相似度为 0，但实际上 A 和 B 的相似度非常高。当然，在实际协同标记系统中，这种情况出现的频率很低，但是通过这样的情形我们可以得出结论：将特征向量自身那一项简单地设为 0 会降低标签相似度计算的精度（之后的实验结果也验证了这一点）。针对此问题，本文采用基于全集的共现标签表示法（考虑标签自身因素），如式(6)、式(7)所示。

$$v_{t_i}^{tag(self)}=[c'_{t_i,1},c'_{t_i,2},\cdots,c'_{t_i,|T|}] \tag{6}$$

$$\begin{cases}c'_{t_i,j}=|\{k|b_{k,i}>0\}|, & \text{if } i=j\\ c'_{t_i,j}=|\{k|b_{k,i}>0\&b_{k,j}>0\}|, & \text{if } i\neq j\end{cases} \tag{7}$$

与前一种方法的不同之处在于，对于向量自身那一项不再设置为0，而是将其设置为标签 t_i 标记过的资源数。该方法利用标签标记过的信息来表征自身，使得标签的特征向量更真实、更充分，从而有效地提高了相似度的精确度。如果再用优化过的表示方法来表征上文提到的标签 A 和 B，则 A 的特征向量为[100,100]，B 的特征向量为[100,100]，它们的特征向量完全相同，相似度为1，这个结果更符合实际情形。

3 基于 MapReduce 的 SOINN 算法

SOINN 算法中，计算复杂度最高的地方是每个对象与类中心之间的距离运算以及类中心权重和边属性的更新。在每次迭代中，假设目前网络中已经存在 n 个类中心，都需要 n 次的距离计算。很明显，一个对象与所有类中心之间的距离运算和其他对象与所有类中心之间的距离运算是不相关的，因此不同对象与类中心之间的距离运算可以并行执行。同时，类中心权重的更新与边的属性更新均是不相关的。根据这一思想，将 SOINN 算法进行了 MapReduce 化。

3.1 SOINN 算法相关参数设置

3.1.1 类中心 i 的相似性阈值

类中心的相似性阈值 T_i[12]用于判断输入的 Web 标签是否属于已知的类，是一个非常重要的参数。开始时，网络中没有类中心。在经过一段时间的学习后，输入的 Web 标签被分成小组，每一个小组由一个主要类中心及它的邻居类中心组成。小组之间又可以组成一个很大的组；每个大组之间是彼此分离的，即每一个大组为一个类。那么，相似性阈值 T_i 必须大于类内节点之间的最大距离，小于类与类之间的距离。

本文采用如下算法更新类中心 i 的相似性阈值 T_i：

流程：

(1)当输入的 Web 标签 i 被当作一个新的类中心插入网络时，它的相似性阈值 T_i 初始值为正无穷大；

(2)当类中心 i 是胜出类中心或次胜出类中心时，如果类中心 i 有邻居类中心，则 T_i 为类中心 i 与邻居类中心的最大距离，否则 T_i 为类中心 i 与网络中所有类中心的最小距离。

3.1.2 学习效率

学习速率直接决定了网络聚类分析的稳定性与速度。在本文中，我们采取了一种类似于 K-Means 算法的学习速率，即：

$$\varepsilon_1(t)=\frac{1}{t},\varepsilon_2(t)=\frac{1}{100t}$$

其中，$t=M_i$。通过这种方式，胜出类中心的学习速率会随着胜出次数的增多而越来越小，胜出类中心及其邻居类中心将越来越稳定。

3.2 基于 MapReduce 的并行算法改进

MapReduce 的数据结构主要为〈key，value〉对，必须将 SOINN 算法中的关键数据结构转化为对应的〈key，value〉对。根据 SOINN 算法的描述可知，一共有 3 种类型的数据，第一种为输入的 Web 标签，第二种为类中心，第三种为类中心互相连接的边。它们的数据结构定义如下。

3.2.1 Web 标签

对于输入的 Web 标签，key 为输入的 Web 标签的 ID，不同的标签对应唯一的 ID。value 保存 Web 标签的对应权重。本文采用基于全集共现标签的特征向量表示法来生成对应权重，Web 标签对应的〈key，value〉对具体数据结构如图 1 所示。

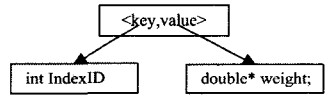

图 1 Web 标签〈key，value〉对数据结构

3.2.2 类中心

类中心〈key，value〉对数据结构如图 2 所示。

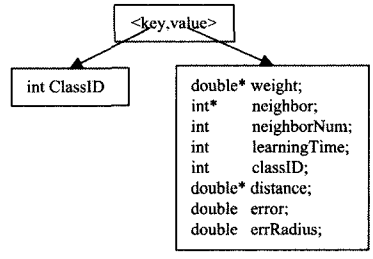

图 2 类中心〈key，value〉对数据结构

3.2.3 边

对于边来说，key 为边的 ID，具体的数据结构如图 3 所示。其中，fromID 与 toID 均为类中心的类 ID，age 为当前边的年龄。

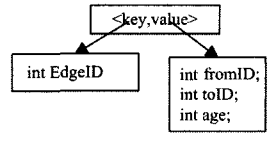

图 3 边〈key，value〉对数据结构

根据 3.1.1 节的算法描述，边的操作比较简单，得出胜出类中心与次胜出类中心后，map 任务负责边的更新，reduce 任务负责移除已经死亡的边。伪代码如下所示：

```
输入：边〈key，value〉对
输出：新的 NeighborTable、ClassTable
流程：
(1)if value. age>agedead
  删除该边
  else
  保留该边
(2)更新 NeighborTable、ClassTable
```

4 系统测试与分析

4.1 SOINN 算法测试与分析

首先，采用随机数生成方法对 SOINN 算法的聚类分析效果进行测试。

图 4 为具有 4 个类的输入数据的某一时刻的分布图，图 5 为聚类分析后得到的效果图。从图 4、图 5 可知，SOINN 聚类算法能够很好地对输入数据进行聚类分析，并能够表征类内中心的拓扑结构。

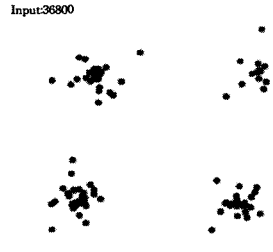

图 4 具有 4 个类的输入数据分布图

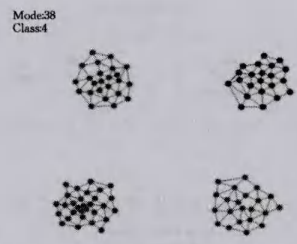

图 5　具有 4 个类的输入数据聚类分析效果图

4.2　数据集的选择

实验所采用的数据是从豆瓣网中的图书版块中爬取得到的，如图 6 所示，在每本书的下方均有所标记的标签，从每本书中抽取这些用户标记信息作为实验数据，其中包含了 442163 本书籍，91234 个标签，每本书平均被多个标签所标记。

图 6　《宇宙奇趣全集》豆瓣网标签示例

首先，为了验证该聚类算法的准确度以及方便观察，我们从这 9 万多个标签中选取了 100 个具有代表性的标签作为待聚类的标签。为了能够对实验结果有一个客观的评价，我们根据豆瓣网上的分类信息以及先验知识将这 100 个标签首先做了人工分类，然后将聚类算法得到的聚类结果与该人工分类进行对比分析，以得到聚类算法的各项性能指标，如精确度、召回率等。

4.3　实验评价标准

如 2.2 节所述，在评价一个聚类算法的聚类效果时，精确度、召回率、Rand 指标和 F1 指标是常用的评价指标。为了能够准确地反映聚类效果，这里采用使用较多的评价方法：F1 聚类评价指标。选择 F1 参数作为衡量指标是因为其容易理解并且满足了这样的需求：将相似的标签放在一起的同时，将不相似的标签区分开。

与传统的在分类评价中的 F1 参数相同，F1 聚类评价指标也是精确度与召回率的调和平均值。下面给出这种评价指标的计算方法。

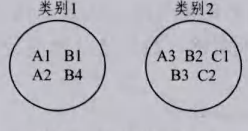

图 7　聚类分析示例

如图 7 所示，两个类别是由聚类算法得到的，而(A1，A2，A3)、(B1，B2，B3，B4)、(C1，C2)表示先验类别。考虑以下两种情形："任意两个标签组成的标签对是否属于相同的类别"；"聚类算法是否将该对标签分在了同一个类别中"。例如图 7 中，聚类算法将(A1，A2)分在了同一类别中，而将(B1，B2)分在了不同的类别中。如果只考虑图 6 中所有的 36 个标签对，可以得到 4 种情形：

正确积极的(TP：True Positives)：聚类算法将一对标签分在了同一类别中，并且在先验类别中它们也在相同的类别中，如(A1，A2)，这里一共有 4 对 TP。

错误积极的(FP：False Positives)：聚类算法将一对标签分在了同一类别中，但在先验类别中它们属于不同的类别，如(A1，B1)，这里一共有 12 对 FP。

正确消极的(TN：True Negatives)：聚类算法将一对标签分在了不同类别中，并且在先验类别中它们也属于不同的类别，如(A1，B2)，这里一共有 14 对 TN。

错误消极的(FN：False Negatives)：聚类算法将一对标签分在了不同类别中，但在先验类别中它们属于相同的类别，如(A1，A3)，这里一共有 6 对 FN。

然后我们得到：

计算精确度为：

$$precision=\frac{TP}{(TP+FP)}=\frac{4}{16}=0.25$$

召回率为：

$$recall=\frac{TP}{(TP+FN)}=\frac{4}{10}=0.4$$

F1 参数为：

$$F1=\frac{(2\times precision\times recall)}{(precision+recall)}\approx 0.308$$

4.4　实验结果分析

4.4.1　定性分析

限于篇幅，这里只列出基于全集共现标签的特征向量表示法的聚类结果，所得结果如表 1 所列。

表 1　基于全集共现标签的特征向量表示法的聚类结果

类	标签
1	神经网络、人工智能、人工神经网络、计算机、神经网络入门、AI、神经计算、聚类、科技、计算机
2	美食、素食、生活、食谱、饮食、吃
3	经济学、经济学原理、经济、教材、经济学入门、金融
4	漫画、日本漫画、日本、卡通
5	散文、随笔、散文随笔、文学
6	旅行、探险、生活、游记
7	职场、职业规划、励志、管理、人力资源、职场工作、HR
8	互联网、IT、万维网、社交、社交网络、WWW、Web、网络技术、数据挖掘
9	建筑、建筑史
10	音乐、音乐欣赏、艺术、音乐教程、音乐学、古典音乐
11	电影、电影理论、电影分析、艺术、电影欣赏
12	室内设计、家居、家具、设计、温馨
13	推理、推理小说、悬疑、犯罪
14	投资、股票、理财、证券
15	哲学、哲学史

从表 1 可以看出，SOINN 算法能够比较准确地对标签进行分类。

4.4.2　定量分析

"Original"表示传统的基于标签共现的算法，"Source-Based"表示基于资源的特征向量表示算法，"OtherBased"表示基于其他共现标签的特征向量表示算法，"AllBased"表示基于全集共现标签的特征向量表示算法。观察图 8 和图 9 可以发现，在均采用 SOINN 聚类算法的基础上，本文提出的 3 种基于对象特征向量表示算法的聚类效果都要优于传统的标签共现算法，这是由于用一个维数很高的特征向量去表征一个标签时，其精确度很高，因此这种表示方法计算出来的标签

之间的相似度比仅比较两个标签一起出现的次数的更加精确，最后所得到的聚类效果也更加优秀。

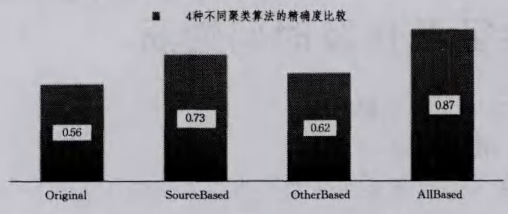

图 8　4 种不同聚类算法的精确度比较

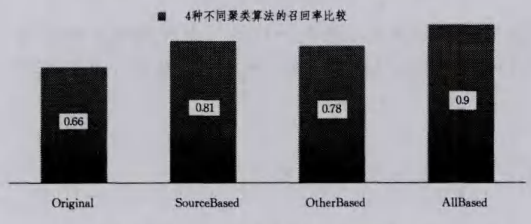

图 9　4 种不同聚类算法的召回率比较

图 10 为类中心数量随 Web 标签数量的变化情况，曲线只表示类中心数量的变化趋势，实际情况可能在某些点有波动。这是因为 SOINN 算法有类中心的插入和删除过程。

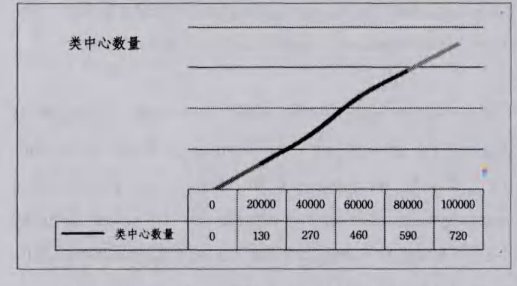

图 10　类中心数量随 Web 标签数量的变化情况

图 11 为采用 MapReduce 编程模型实现的聚类分析算法与未采用 MapReduce 编程模型的聚类算法的速度对比图。横轴表示当前类中心的数量，纵轴单位为分钟。观察该图可以发现，随着数据量的成倍增长，算法的执行时间也是接近线性增长，并获得了很好的加速比。从而证明了本文提出的分布式聚类算法具有很好的可扩展性，可以应对更为海量的 Web 日志聚类分析系统。

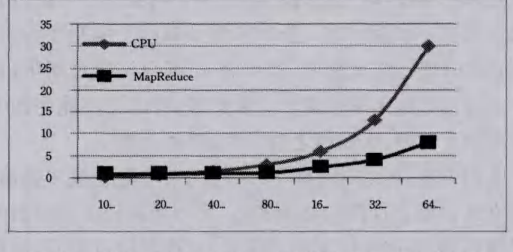

图 11　CPU 与 MapReduce 聚类算法执行时间对比图

从图 11 中还可以看出，当类中心数量小于 2000 左右时，采用 MapReduce 编程模型的聚类算法反倒比采用 CPU 慢，这是由于 MapReduce 预处理开销以及通信、同步所消耗的时间占主要部分，导致整个系统性能下降。

结束语　本文论述了一种基于 MapReduce 的 Web 标签聚类算法。该方法采用基于全集共现标签的特征向量表示法来对 Web 标签进行量化，获得了很高的精确度；采用了 SOINN 聚类分析算法，由于该算法是自增长的，无需事先定义类中心的整体规模大小，因此不会像 K-Means 算法那样产生类中心不足或者过剩的情况。同时，在该算法类中心之间加入的边连接，能够表征类内节点之间以及类与类之间的拓扑关系；最后，采用 MapReduce 编程模型对整个系统进行加速，获得了很好的加速比，使系统能够满足海量标签聚类的需求。

本文论述的算法仍然有一定的局限性。首先，若标签非常多，例如几十万以上，那么采用基于全集共现标签的特征向量表示法表征 Web 标签时，Web 标签的向量维度会达到几十万以上，这就导致系统性能急剧下降。如何对向量进行压缩，将是下一步工作的重点之一。其次，SOINN 算法的 MapReduce 实现属于粗粒度的实现，没有考虑算法内部的并行化，所以还存在一定的加速空间有待挖掘。

参 考 文 献

[1] Kamel B, Wheeler S. The emerging Web 2.0 social software: an enabling suite of sociable technologies in health and health care education1[J]. Health Information & Libraries Journal, 2007, 24(1): 2-23

[2] Li Y, An J. Analysis on the Online Public Opinion Management in the Context of Web 2.0[C]// Proceedings of 2010 International Conference on Public Administration(6th). 2010: 418-422

[3] Kipp M E I, Campbell D G. Patterms and inconsistencies in collaborative tagging systems: An examination of tagging prattices [C]// Proceedings of the American Society for Information Science and Technology. 2006: 1-18

[4] Grigory B, Philipp K, Frank S. Automated Tag Clustering: Improving search and exploration in the tag space[C]// Collaborative Web Tagging Workshop at www 2006. Edinburgh, Scotland, 2006: 15-33

[5] Paul H, Hector G. Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems[R]. Stanford, 2006

[6] Ramage D, Heymann P, Manning C D, et al. Clustering the tagged web[C]// International Conference on Web Search and Web Data Mining(WSDM). ACM, 2009: 54-63

[7] Gunarathne T, Zhang B J, Wu T L, et al. Scalable parallel computing on clouds using Twister4Azure iterative MapReduce[J]. Future Generation Computer Systems, 2013, 29(4): 1035-1048

[8] Furao S, Hui Y, Sakurai K, et al. An incremental online semi-supervised ac-tive learning algorithm based on self-organizing incremental neural network[J]. Neural Computing & Applications, 2011, 20(7): 1061-74

[9] Kawewong A, Honda Y, Tsuboyama M, et al. Reasoning on the Self-organizing Incremental Associative Memory for Online Robot Path Planning[J]. IEICE transactions on information and systems, 2010, 93(3): 569-582

[10] Ching-man A, Nicholas G, Nigel S. Contextualising Tags in Collaborative Tagging Systems[C]// 20th ACM Conference on Hypertext and Hypermedia. ACM, 2009: 251-260

[11] Gunarathne T, Zhang B, Wu T L, et al. Scalable parallel computing on clouds using Twister4Azure iterative MapReduce[J]. Future Generation Computer Systems, 2013, 29(4): 1035-1048

[12] Shen F, Osamu H. An incremental network for on-line unsupervised classification and topology learning[J]. Neural Networks, 2006, 19(1): 90-106