Journal of Computer Applications
计算机应用,2014,34(11):3073 -3077
ISSN 1001-9081
CODEN JYIIDU
2014-11-10
http://www.joca.cn

文章编号:1001-9081(2014)11-3073-05
doi:10.11772/j.issn.1001-9081.2014.11.3073

基于 MapReduce 的并行化模糊划分算法

张广蓉[1*],陈庆奎[1,2],章　刚[2],赵海燕[1],高丽萍[1],霍　欢[1]
(1.上海理工大学 光电信息与计算机工程学院,上海 200093;　2.上海理工大学 管理学院,上海 200093)
(*通信作者电子邮箱 island_rong@126.com)

摘　要:针对大规模项目资源库中项目资源信息无序而导致无法准确快速找出项目资源库中所需资源的问题,提出了基于 MapReduce 的并行化模糊聚类划分算法。该算法首先抽象原始项目资源特征属性并标准化;其次,根据标准化后的特征属性建立项目相似矩阵,运用矩阵分块思想分割矩阵;然后,利用 MapReduce 技术处理分块矩阵并合并结果;最后,运用阈值评判划分成若干个有序的项目组。与 *K*-means 算法和遗传算法的对比实验结果证明:该算法具有较高的准确率和查全率,并且在大规模数据计算时能够得到较高的加速比,可以有效准确地划分项目资源。

关键词:MapReduce;模糊划分;并行计算;大规模数据
中图分类号:TP338.6　　**文献标志码**:A

Parallel fuzzy partition algorithm based on MapReduce

ZHANG Guangrong[1*], CHEN Qingkui[1,2], ZHANG Gang[2], ZHAO Haiyan[1], GAO Liping[1], HUO Huan[1]
(1. *School of Optical Electrical Information and Computer Engineering, University of Shanghai for Science and Technology, Shanghai* 200093, *China*;
2. *Business School, University of Shanghai for Science and Technology, Shanghai* 200093, *China*)

Abstract: It is difficult for users to find the needed items from a large-scale project resource repository because the project resources in it are disordered, so a parallel fuzzy partition algorithm based on MapReduce was proposed. The algorithm firstly abstracted and standardized characteristic attributes of original project resource. Then a similarity matrix was established based on the standardized characteristic attributes of the project, and it was segmented by using block matrix. MapReduce was used to process the block matrix and merge the results. Finally, the algorithm obtained the partition results according to the threshold. The contrast experiment among the proposed algorithm, *K*-means algorithm and genetic algorithm shows that the proposed algorithm has higher accuracy and recall, it can achieve better speedup in large-scale data calculation and divide project resources effectively and accurately.

Key words: MapReduce; fuzzy partition; parallel computing; large-scale data

0　引言

随着经济全球化,公共服务平台[1-3]成为支持企业间密切协作的重要渠道。科技对接平台是公共服务平台的关键核心,其主要功能之一是将海量项目资源库中的项目资源进行有序划分,使无序资源有序化,有效地为用户提供服务。图 1 展示了平台架构中的大规模项目资源划分过程。

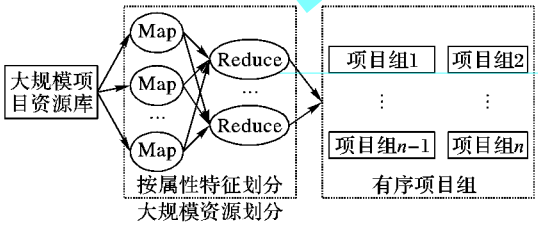

图 1　大规模资源划分过程示意图

由图 1 可知,面对大规模项目资源库中无序的项目资源,如何根据项目的特征将大规模项目资源库划分为若干个有序项目组,使得其能更有效地为用户提供服务是本文研究重点。

目前对项目进行划分的研究已有很多。文献[4]采用了模糊聚类的方法运用活动环境聚类寻找相似环境聚类改善了传统推荐算法中的硬聚类和硬分解问题;文献[5]总结分析了模糊聚类的特点以及存在的问题;文献[6]提出了基于 MapReduce 模型的 AP(Affinity Propagation)算法提高了传统算法的效率。但是,文献[4-6]所提出的传统模糊聚类对大规模数据聚类效率太低。文献[7]基于 MapReduce 的并行遗传算法对算法的准确性有所提高,但由于运用遗传算法,算法的效率不高。文献[8]运用了 *K*-means 在 MapReduce 框架下的并行化进行了自适应聚类,改善了算法运行的结果,采用基于 MapReduce 的聚类算法在效率上有所提高,但是对于形状不规则数据集聚类效果不佳。

收稿日期:2014-06-05;**修回日期**:2014-08-04。　**基金项目**:国家自然科学基金资助项目(60970012);高等学校博士学科点专项科研博导基金资助项目(20113120110008);上海教委创新重点项目(13ZZ112);上海市一流学科建设项目(XTKX2012);上海市工程中心建设项目(GCZX14014);上海重点科技攻关项目(09511501000, 09220502800)。

作者简介:张广蓉(1989-),女,上海人,硕士研究生,主要研究方向:大规模数据处理;　陈庆奎(1966-),男,上海人,教授,博士生导师,主要研究方向:网络计算、云计算、并行计算;　章刚(1981-),男,上海人,博士研究生,主要研究方向:网络计算;　赵海燕(1975-),女,河南温县人,副教授,主要研究方向:服务计算;　高丽萍(1980-),女,山东烟台人,副教授,主要研究方向:计算机支持协同工作与协同计算;　霍欢(1979-),女,辽宁沈阳人,副教授,主要研究方向:XML 数据流。

计算机应用 www.joca.cn

针对上述提到的问题,本文提出了基于 MapReduce 的并行化模糊划分方法,首先,将平台中的项目数据标准化,建立相似矩阵;然后运用 MapReduce 对相似矩阵采用并行化分块矩阵乘法求等价矩阵,根据阈值将项目划分成若干个相似度较高的组,完成大规模项目资源库项目划分问题。

1 基于 MapReduce 的并行化模糊划分

模糊聚类[9-10]是将模糊数学引入聚类分析,采用隶属度的概念确定样本间亲疏关系的聚类方法,它适合处理客观世界中大量存在的界限不分明的聚类问题,而对接平台中项目的划分融合问题正存在着中介性,因此可以用模糊聚类描述这类划分问题。但是在处理大规模资源时,使用模糊聚类算法的时间性能难以令人满意。因此本文提出基于 MapReduce 的并行化模糊聚类划分算法。对于项目模糊聚类可大致分为 4 个步骤:1)属性数据标准化;2)建立模糊相似矩阵;3)基于 MapReduce 矩阵分块乘法求模糊等价闭包;4)根据阈值划分项目。

1.1 属性数据标准化

建立数据矩阵并去除数据矩阵中的量纲,从而得到标准化的数据。

1.1.1 建立数据矩阵

将平台中所有的项目放在一个数据集中,设数据集为 $\boldsymbol{D}=\{\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_n\}$,$n$ 为数据集中项目的总个数,每个项目都由 m 个属性来表示项目的特征。向量 $\boldsymbol{D}_i=(x_{i1},x_{i2},\cdots,x_{im})(i=1,2,\cdots,n)$ 表示第 i 个项目的特征描述。所以,得到了项目的 $n\times m$ 原始数据矩阵:$\boldsymbol{D}=[\boldsymbol{D}_i\mid \boldsymbol{D}_i=(x_{i1},x_{i2},\cdots,x_{im}),1\leqslant i\leqslant n]$。

1.1.2 数据标准化

在项目集中用户项目的 m 个特征属性都有不同的量纲,为了方便计算比较不同量纲下的特征属性,需要把不同量纲的特征属性根据模糊聚类矩阵的要求规范到区间[0,1]内。采用平移极差变换,计算过程如下:

1)平移标准差。

$$x_{ik}'=(x_{ik}-\overline{x_k})/s_k;\ i=1,2,\cdots,n;k=1,2,\cdots,n \tag{1}$$

其中:

$$s_k=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ik}-\overline{x_k})^2}$$

$$\overline{x_k}=\frac{1}{n}\sum_{i=1}^{n}x_{ik}$$

完成平移标准差以后,就消除了变量的量纲影响,可以用来进行特征属性的比较,但是 x_{ik}' 不一定在区间[0,1]内。

2)平移极差变换。

$$x_{ik}''=\frac{x_{ik}'-\min\limits_{1\leqslant i\leqslant n}\{x_{ik}'\}}{\max\limits_{1\leqslant i\leqslant n}\{x_{ik}'\}-\min\limits_{1\leqslant i\leqslant n}\{x_{ik}'\}} \tag{2}$$

经过了平移极差变换后,对于每个属性变量都可以得到 $0\leqslant x_{ik}''\leqslant 1$,而且已经消除了量纲的影响。

1.2 建立模糊相似矩阵

要被分类的原始项目集为 $\boldsymbol{D}=\{\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_n\}$,每个项目描述为 $\boldsymbol{D}_i=(x_{i1},x_{i2},\cdots,x_{im})(i=1,2,\cdots,n)$,设 $\boldsymbol{L}(l_1,l_2,\cdots,l_m)$ 为资源权值向量,其中 l_i 表示第 i 个属性的权值,$l_1+l_2+\cdots+l_m=1$。关于权值的选取会在后续的章节中具体讨论。要把属性特点相近的项目归为一类,就要求出任意两个项目 $\boldsymbol{D}_i$、$\boldsymbol{D}_j$ 之间的相似度 $r(\boldsymbol{D}_i,\boldsymbol{D}_j)$,记为 r_{ij},显然有 $r_{ii}=1$ 和 $0\leqslant r_{ij}=r_{ji}\leqslant 1$。这样由项目相似度 r_{ij} 就可以构建一个用户项目集 $\boldsymbol{D}$ 的项目之间的一个模糊相似矩阵 $\boldsymbol{R}$。

$$\boldsymbol{R}=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{bmatrix}$$

其中:$r_{ii}=1;0\leqslant r_{ij}=r_{ji}\leqslant 1;1\leqslant i,j\leqslant n$。

相似度有很多种求解方法,本文这里给出绝对值指数法来求模糊相似度 r_{ij},绝对值指数法如下:

$$r_{ij}=\exp\left(-\sum_{k=1}^{m}l_k\mid x_{ik}-x_{jk}\mid\right) \tag{3}$$

其中:r_{ij} 表示 x_i 与 x_j 的相似系数,方法适用于 $r_{ij}>0$ 的情况,$1\leqslant i\leqslant n,1\leqslant k\leqslant m$。

1.3 基于 MapReduce 矩阵分块乘法求模糊等价闭包

有了模糊相似矩阵 $\boldsymbol{R}$ 还是不够的,此时矩阵 $\boldsymbol{R}$ 还没有传递性,也就是说如果把项目与项目之间的相似度大于等于阈值 α 的项目看作是一类的,那么当 $r_{ij}\geqslant\alpha$ 且 $r_{jk}\geqslant\alpha$ 时不一定就有 $r_{ik}\geqslant\alpha$,即当项目 x_i 与项目 x_j 是同类且项目 x_j 与项目 x_k 是同类的时候,项目 x_i 与项目 x_k 未必是同类的,所以在相似矩阵的基础上还要求出等价闭包(记为 $\boldsymbol{R}^*$),使它具有传递性。由于矩阵 $\boldsymbol{R}$ 是相似矩阵,但是不一定是等价矩阵,因此必须计算等价关系矩阵 $\boldsymbol{R}^*$。

计算相似矩阵的传递闭包一般采用平方法[11],即:当 $\boldsymbol{R}^{2k}=\boldsymbol{R}^k\times\boldsymbol{R}^k=\boldsymbol{R}^k$ 时,$\boldsymbol{R}^*=\boldsymbol{R}^k$ 就是所求的等价关系矩阵。

1.3.1 MapReduce 模型

传统的矩阵乘法的都是通过求左矩阵行与右矩阵的列的内积来求解矩阵的乘积。但是当项目数据集的数据量大、属性多时,目前的矩阵乘法不能处理大规模的数据。因此将矩阵分块可以实现大规模矩阵乘法,就是将两个大矩阵相乘转换为若干个小矩阵相乘,这样可以有效地简化矩阵相乘的运算。

因此本文在计算等价关系矩阵时,使用 MapReduce[12]进行计算。MapReduce 通常会把输入数据切分为若干独立的数据块,并以〈*key*,*value*〉的形式为输入。MapReduce 的处理过程[13]分为 Map 过程和 Reduce 过程,Map 函数和 Reduce 函数可由用户自定义。Map 过程对输入的〈*key*,*value*〉进行处理得到中间结果〈*key*1,*value*1〉,然后 Reduce 过程将中间结果处理合并,最后输出结果。在 MapReduce 过程中,所有节点独立运行相同的 Map 函数和 Reduce 函数处理不同的数据块。

1.3.2 基于 MapReduce 矩阵乘法的实现

矩阵划分有多种方法,本文采用将若干连续行和若干连续列的交叉部分划分为一块作为小矩阵进行计算。

计算机应用 www.joca.cn

相似矩阵 $\boldsymbol{R}=(r_{ij})$ 是一个 $n\times n$ 的矩阵,则等价相似矩阵 $\boldsymbol{R}^*$ 也是一个 $n\times n$ 的矩阵。文献[14]已经证明矩阵分块相乘结果与原矩阵相乘结果相同。因此将矩阵 $\boldsymbol{R}$ 划分为若干个 $s\times s$ 等大小的矩阵,假设分块后的矩阵 $\boldsymbol{R}$ 是 $S\times S$,则按照这个规则 $\boldsymbol{R}$ 矩阵可以表示为:

$$\boldsymbol{R}=\begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1S} \\ R_{21} & R_{22} & \cdots & R_{2S} \\ \vdots & \vdots & & \vdots \\ R_{S1} & R_{S2} & \cdots & R_{SS} \end{bmatrix}$$

为了方便阐述下文的算法,假设这两个矩阵分别为 $\boldsymbol{A}$ 和 $\boldsymbol{B}$,结果为 $\boldsymbol{C}$ 矩阵。其中左矩阵 $\boldsymbol{A}$ 和右矩阵 $\boldsymbol{B}$ 分别为:

$$\boldsymbol{A}=\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1S} \\ A_{21} & A_{22} & \cdots & A_{2S} \\ \vdots & \vdots & & \vdots \\ A_{S1} & A_{S2} & \cdots & A_{SS} \end{bmatrix}$$

$$\boldsymbol{B}=\begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1S} \\ B_{21} & B_{22} & \cdots & B_{2S} \\ \vdots & \vdots & & \vdots \\ B_{S1} & B_{S2} & \cdots & B_{SS} \end{bmatrix}$$

矩阵 $\boldsymbol{C}$ 中的元素 C_{ij} 是由 $\boldsymbol{A}$ 中第 i 行与 $\boldsymbol{B}$ 中第 j 列元素依次对应相乘并相加得到的结果。

$$C_{ij}=\boldsymbol{A}_i\cdot\boldsymbol{B}_j=\sum_{k=1}^{S}A_{ik}\cdot B_{kj} \tag{4}$$

因此将大规模矩阵分块化,形成若干个小矩阵,计算小矩阵的乘积然后合并,可以准确地得到大规模矩阵的乘积。

1.3.3　Map 函数的设计

矩阵 $\boldsymbol{A}$ 中的每一块都需要和矩阵 $\boldsymbol{B}$ 中对应位置的块依次相乘,这些块与块之间的相乘运算都是独立的,不受彼此影响的,因此块与块之间的对应相乘可以由 Map 函数来完成。输入数据为 $\langle(i,j),\boldsymbol{A}_i^{\mathrm{T}}\rangle$ 和 $\langle(i,j),\boldsymbol{B}_j\rangle$ 即为左矩阵的行号和列号以及对应行中的元素和对应列中的元素。每个 Map 函数都需要完成相同 *key* 值时 *value* 的相乘,这样就可以独立地完成块与块之间元素的相乘。输出结果为 $\langle(i,j),\boldsymbol{A}_i^{\mathrm{T}}\cdot\boldsymbol{B}_j\rangle$,其中 (i,j) 为元素的位置,i 是对应的行号,j 是对应的列号,$\boldsymbol{A}_i^{\mathrm{T}}\cdot\boldsymbol{B}_j$ 为左矩阵 $\boldsymbol{A}$ 的第 i 行与右矩阵 $\boldsymbol{B}$ 的第 j 列的积的结果。

1.3.4　Reduce 函数的设计

在 Reduce 函数执行之前,系统会对 Map 函数输入结果中相同 *key* 的 *value* 进行合并。

Reduce 函数的任务是完成相同行列元素的相加。输入的数据为 $\langle(i,j),elements(i,j)\rangle$,其中:$(i,j)$ 为结果矩阵 $\boldsymbol{C}$ 中元素的位置,i 为行号,j 为列号,在合并阶段相同 *key*(即 *key* 都是 (i,j))的 *value* 会被放入同一个 $elements(i,j)$ 中。Reduce 函数需要将相同 *key* 的 $elements(i,j)$ 中的乘积进行求和。

如果矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 为 $\boldsymbol{R}^k$,那么此时结果矩阵 $\boldsymbol{C}$ 即为 $\boldsymbol{R}^{2k}$,当 $\boldsymbol{R}^{2k}=\boldsymbol{R}^k\times\boldsymbol{R}^k=\boldsymbol{R}^k$ 时,这时 $\boldsymbol{R}^*=\boldsymbol{R}^k$ 就是所求的等价关系矩阵;如果不相等则返回继续用 MapReduce 计算,直到 $\boldsymbol{R}^{2k}=\boldsymbol{R}^k\times\boldsymbol{R}^k=\boldsymbol{R}^k$。图 2 为基于 MapReduce 的乘法运算过程。

图 2　基于 MapReduce 的乘法运算过程

1.4　根据阈值划分项目

在模糊聚类分析中,对于相同的模糊等价矩阵 $\boldsymbol{R}^*$,若取不同的阈值 $\alpha(0\leqslant\alpha\leqslant1)$,可以得到不同的项目分类,因此要根据具体项目集的特点来确定一个阈值 α。阈值 α 反映了系统对于项目之间相似度的采纳程度,相似度在这个阈值内的项目被认为是特征相似的项目,即当项目之间的相似度大于等于阈值 α 时,这两个项目就被认为是相似的。

1.4.1　权值与阈值的确定

设项目资源库中的一个项目资源为 P,项目的特征属性对项目 P 有不同的影响,所以用 $\boldsymbol{L}(l_1,l_2,\cdots,l_m)$ 来表示特征属性对 P 的影响,l_i 表示第 i 个特征属性对项目 P 的影响度,m 是特征属性的数目。向量 $\boldsymbol{L}$ 需要规约到区间[0,1]内,可以用下面的公式进行规约:

$$l_i'=l_i\Big/\sum_{j=1}^{m}l_j;\ 1\leqslant i\leqslant m$$

这样 l_i 为第 i 个特征属性对项目 P 的影响度,即为第 i 个特征属性的权值。影响度向量 $\boldsymbol{L}$ 变换为权值向量 $\boldsymbol{L}'(l_1',l_2',l_3',\cdots,l_m')$。由于特征属性对项目的影响存在差异,故需要用户给出一个最低误差容忍度向量[15] $\boldsymbol{T}(t_1,t_2,\cdots,t_m)$,$t_i$ 为第 i 个特征属性的最低误差容忍度,且 $0\leqslant t_i\leqslant1$,t_i 值越大说明该因素越重要,要求其误差就越小。故可以计算项目资源的划分阈值 α,计算公式如下:

$$\alpha=\sum_{i=1}^{m}l_i'\times t_i$$

由于 $\sum_{i=1}^{m}l_i'=1$,且 $0\leqslant t_i\leqslant1$,因此 $\alpha\leqslant1$。故可以利用 α 来作为划分项目的阈值。

计算机应用 www.jisuanjiyingyong.com

1.4.2 划分项目

对所得的闭包矩阵 $\boldsymbol{R}^*$ 和根据用户需求而确定的阈值 α 进行项目划分[16]，划分过程可以如下计算：

$$\boldsymbol{M}_L = \boldsymbol{R}^* - \boldsymbol{M}_\alpha \tag{5}$$

其中：$\boldsymbol{M}_\alpha = [t_{ij} \mid t_{ij} = \alpha, 1 \leqslant i, j \leqslant n]$，$\boldsymbol{M}_L$ 和 $\boldsymbol{M}_\alpha$ 均为 $n \times m$ 矩阵。

从矩阵 $\boldsymbol{M}_L$ 中清除值不大于 0 的元素，剩下的元素的下标所对应的计算节点，即为同一个项目组，因为 $\boldsymbol{M}_L$ 具有对称性，因此要去掉重复的元素，最后得到的元素下标就是同一个项目组的项目编号。

2 算法描述

算法的核心是使用基于 MapReduce 的并行化模糊聚类，将所有数据划分成若干个相似度较高的组，完成一个大规模项目划分问题。算法步骤如下：

输入 原始项目数据。

输出 有序的项目组。

Step1 建立原始项目数据集矩阵。项目集为 $\boldsymbol{D} = \{\boldsymbol{D}_1, \boldsymbol{D}_2, \cdots, \boldsymbol{D}_n\}$，其中 $\boldsymbol{D}_i = (x_{i1}, x_{i2}, \cdots, x_{im})$，表示第 i 个项目的特征描述，则得到了 $n \times m$ 的项目原始数据矩阵。

Step2 标准化项目数据。对于原始矩阵 $\boldsymbol{D}$ 中的每一个元素 x_{ik}，根据式(1) 计算元素的平移标准差 x_{ik}'，消除项目属性的量纲，由式(2) 得到元素的平移极差变换 x_{ik}''，使数据根据模糊矩阵的要求规范到区间[0,1] 内。

Step3 建立模糊相似矩阵。根据式(3) 计算任意两个项目 $\boldsymbol{D}_i$、$\boldsymbol{D}_j$ 之间的相似度 r_{ij}，由相似度 r_{ij} 建立模糊相似矩阵 $\boldsymbol{R}$。

Step4 由 MapReduce 并行计算等价闭包，将相似矩阵分块，运用 MapReduce 对相似矩阵采用并行化矩阵乘法求等价矩阵完成聚类划分，即：当 $\boldsymbol{R}^{2k} = \boldsymbol{R}^k \times \boldsymbol{R}^k = \boldsymbol{R}^k$，此时 $\boldsymbol{R}^* = \boldsymbol{R}^k$ 就是所求的等价矩阵。

Step5 根据阈值划分等价矩阵，对 Step4 得到的等价闭包 $\boldsymbol{R}^*$ 和定义的阈值 α 根据式(5) 计算得到最终的划分矩阵 $\boldsymbol{M}_L$，$\boldsymbol{M}_L$ 中值大于 0 的元素所对应的下标就是同一项目组中项目的标号，由等价闭包的传递性可以求出在同一个项目组中所有项目的标号。

Step6 得到有序的项目组，完成项目资源划分，算法结束。

3 实验与分析

3.1 实验环境

实验采用 4 台普通计算机搭建的 Hadoop 集群系统进行的，Hadoop 版本为 Hadoop-0.20.2。每台计算机使用的操作系统是 Ubuntu10.04。实验均在 PC 上完成，配置如下：CPU 为 Intel Core2 Duo CPU E7300 2.66 GHz，内存为 3.00 GB DDR2 RAM，操作系统为 Windows 7，编程语言为 Java。

3.2 测试指标

下面分别从聚类的准确率 *PRE*、查全率 *REC*[17] 来分析聚类的质量，以及算法的加速比 *Sp* 来衡量基于 MapReduce 的分块模糊聚类方法并行化的性能和效果。

各测试指标定义如下：

$$PRE = \frac{1}{n}\sum_{i=1}^{n}\frac{a_i}{a_i + b_i}$$

$$REC = \frac{1}{n}\sum_{i=1}^{n}\frac{a_i}{a_i + c_i}$$

其中：n 表示聚类的个数，a_i 表示准确分到第 i 类的项目个数，b_i 表示错误分到第 i 类的项目个数，c_i 表示应该被分到第 i 类却没有被分到的项目个数。

加速比是同一个任务在单处理器系统和并行处理器系统中运行消耗的时间的比率，其定义为 $Sp = T1/Tp$，其中：$T1$ 是单机传统的模糊聚类算法的运行时间，Tp 是基于 MapReduce 的分块模糊聚类方法的运行时间。

3.3 实验结果与分析

本文采用的数据集包括了 60 000 多个科技服务项目，项目包含 70 维属性，分别是项目的技术分类、行业、技术水平、技术成熟度、信息状态、发布时间以及预计交易价格等。

实验 1 准确率测试及查全率测试。

本实验共进行 4 组，实验中选取的数据样本 G1 ~ G4 分别有 1 000，10 000，30 000，60 000 个项目数据，在权值分配一定的情况下使用 *K*-means 算法、遗传算法和基于 MapReduce 的模糊聚类算法对项目进行划分。由于模糊聚类的结果受阈值选取的影响，因此实验室用多个不同的阈值进行实验，对于每个阈值分别求出聚类的准确率和查全率，计算得出基于 MapReduce 的模糊聚类算法在不同数据集上的平均准确率和查全率。实验结果如表 1 ~ 2 所示。

表 1 不同数据集下算法准确率的比较 %

算法	G1	G2	G3	G4
K-means 算法	87.75	70.93	64.43	60.32
遗传算法	88.92	76.75	74.93	68.84
模糊聚类	90.34	80.93	76.73	70.91

表 2 不同数据集下算法查全率的比较 %

算法	G1	G2	G3	G4
K-means 算法	87.75	77.53	74.27	71.46
遗传算法	92.92	82.79	80.93	75.84
模糊聚类	93.34	84.73	82.73	77.91

由表 1 ~ 2 可看出：在处理小规模数据集时，三种算法的准确率和查全率都在 85% 以上都是比较理想的，但是当数据规模提高到 10 000 个数据样本时，*K*-means 算法所得到的准确率和查全率相对其他两种聚类的结果明显不太理想，遗传算法相对于 *K*-means 算法明显有所改善。这是由于当数据量增大时，项目集中自成一类的项目数量也在增多，数据集中会出现很多非球形的不规则的类簇，而 *K*-means 算法对于非球形簇并没有很好的聚类效果。基于 MapReduce 的模糊聚类所得到的准确率和查全率明显高于其他两种聚类算法。

实验 2 算法性能测试。

实验仍进行 4 组，数据样本同实验 1。所得数据规模与加速比的关系如图 3 所示。

计算机应用 www.joca.cn

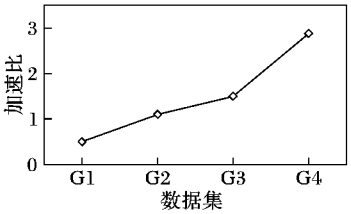

图 3　数据规模与加速比的关系

从图 3 可看出:当数据样本数为 1 000 时,加速比 *Sp* 的值较小只有 0.49,说明当数据量不大时,基于 MapReduce 的分块模糊聚类方法不如单机传统的模糊聚类的运行速度快,因为 MapReduce 需要启动各个任务节点分块计算,而单机计算省去了这部分时间,而数据计算的规模不大,无法充分地利用各个节点的资源来计算,节点与节点之间的通信合并数据消耗的时间占了很大比例,导致数据规模较小的时候单机的模糊聚类计算反而优于基于 MapReduce 的分块算法。但是随着数据规模的增大,基于 MapReduce 的分块法的优点就可以体现出来了,当数据规模扩大到 60 000 时,加速比 *Sp* 的值接近 3,即基于 MapReduce 的分块模糊聚类所需的时间仅为单机模糊聚类计算的 1/3。

4　结语

本文提出了基于 MapReduce 的并行化分块模糊划分算法,解决了大规模项目资源组中无序问题。与当前一些算法例如基于 MapReduce 的 *K*-means 算法和遗传聚类算法进行比较,该算法在准确率及查全率方面都有较好的聚类结果,并且在大规模数据的计算时能够得到较高的加速比。今后工作重点将会围绕如何对数据矩阵进行分块展开。

参考文献:

[1] CHEN J, WANG S, SUN L. Flexible model and architecture of public service platform for business-related multi-industrial chain collaboration[J]. Computer Integrated Manufacturing Systems, 2011, 17(1): 177 – 185. (陈静, 王淑营, 孙林夫. 面向柔性的业务关联的多产业链协作公共服务平台模型和架构[J]. 计算机集成制造系统, 2011, 17(1): 177 – 185.)

[2] WANG S. Integrated framework of collaborative commercial platform for manufacturing industrial chain[J]. Journal of Southwest Jiaotong University, 2008, 43(5): 643 – 647. (王淑营. 面向制造业产业链的协同商务平台集成框架[J]. 西南交通大学学报, 2008, 43(5): 643 – 647.)

[3] AGRAWAL R, BAYARDO R J, Jr, GRUHL D, *et al.* Vinci: a service-oriented architecture for rapid development of Web applications [C]// Proceedings of the 10th International Conference on World Web. New York: ACM Press, 2001: 355 – 365.

[4] ZHANG F, CHANG J, ZHOU Q. Context-aware recommendation algorithm based on fuzzy C-means clustering[J]. Journal of Computer Research and Development, 2013, 50(10): 2185 – 2194. (张付志, 常俊风, 周全强. 基于模糊 C 均值聚类的环境感知推荐算法[J]. 计算机研究与发展, 2013, 50(10): 2185 – 2194.)

[5] ZHANG M, YU J. Fuzzy partitional clustering algorithms[J]. Journal of Software, 2004, 15(6): 858 – 868. (张敏, 于剑. 基于划分的模糊聚类算法[J]. 软件学报, 2004, 15(6): 858 – 868.)

[6] XU X, XIAO Y. KBAC: *K*-means based adaptive clustering for massive dataset[J]. Journal of Chinese Computer Systems, 2012, 33(10): 2268 – 3372. (徐晓旻, 肖仰华. KBAC: 一种基于 *K*-means 的自适应聚类[J]. 小型微型计算机系统, 2012, 33(10): 2268 – 3372.)

[7] JIA R, GUAN Y, LI Y. Parallel *K*-means clustering algorithm based on MapReduce model[J]. Computer Engineering and Design, 2014, 35(2): 657 – 660. (贾瑞玉, 管玉勇, 李亚龙. 基于 MapReduce 模型的并行遗传 *K*-means 聚类算法[J]. 计算机工程与设计, 2014, 35(2): 657 – 660.)

[8] LU W, DU C, WEI B, *et al.* Distributed affinity propagation clustering based on MapReduce[J]. Journal of Computer Research and Development, 2012, 49(8): 1762 – 1772. (鲁伟明, 杜晨阳, 魏宝刚, 等. 基于 MapReduce 的分布式近邻传播聚类算法[J]. 计算机研究与发展, 2012, 49(8): 1762 – 1772.)

[9] YU J, HUANG H. A new weighting fuzzy c-means algorithm [C]// FUZZ 2003: Proceedings of the 12th IEEE International Conference on Fuzzy Systems. Piscataway: IEEE Press, 2003, 2: 896 – 901.

[10] RUNKLER T A, KATZ C. Fuzzy clustering by particle swarm optimization [C]// Proceedings of the 2006 IEEE International Conference on Fuzzy Systems. Piscataway: IEEE Press 2006: 601 – 608.

[11] ZHANG H, DING F, JIANG L. A collaborative filtering recommendation method based on fuzzy clustering[J]. Computer Simulation, 2005, 33(12): 144 – 147. (张海燕, 丁峰, 姜丽红. 基于模糊聚类的协同过滤推荐方法[J]. 计算机仿真, 2005, 33(12): 144 – 147.)

[12] JIN C, VECCHIOLA C, BUYYA R. MRPGA: an extension of MapReduce for parallelizing genetic algorithms [C]// eScience 2008: Proceedings of the Fourth IEEE International Conference on eScience. Piscataway: IEEE Press, 2008: 214 – 221.

[13] ZHAO H, YANG S, CHEN Z, *et al.* Optimization of range queries and analysis for MapReduce systems[J]. Journal of Computer Research and Development, 2014, 51(3): 606 – 617. (赵辉, 杨树强, 陈志坤, 等. 基于 MapReduce 模型的范围查询分析优化技术研究[J]. 计算机研究与发展, 2014, 51(3): 606 – 617.)

[14] SUN Y, CHEN Y, GUAN X, *et al.* Approach of large matrix multiplication based on Hadoop[J]. Journal of Computer Applications, 2013, 33(12): 3339 – 3344, 3358. (孙远帅, 陈垚, 官新均, 等. 基于 Hadoop 的大矩阵乘法处理方法[J]. 计算机应用, 2013, 33(12): 3339 – 3344, 3358.)

[15] LIU B, CHEN Q. Fuzzy clustering partition model for computer cluster in cloud computing[J]. Computer Science, 2011, 38(10): 157 – 160, 168. (刘伯成, 陈庆奎. 云计算中的集群资源模糊聚类划分模型[J]. 计算机科学, 2011, 38(10): 157 – 160, 168.)

[16] WU H, WANG X, CHENG Y, *et al.* Advanced recommendation based on collaborative filtering and partition clustering[J]. Journal of Computer Research and Development, 2011, 48(S3): 205 – 212. (吴泓辰, 王新军, 成勇, 等. 基于协同过滤与划分聚类的改进推荐算法[J]. 计算机研究与发展, 2011, 48(增刊 3): 205 – 212.)

[17] XIAO Y, YU J. Semi-supervised clustering based on affinity propagation algorithm[J]. Journal of Software, 2008, 19(11): 2803 – 2813. (肖宇, 于剑. 基于近邻传播算法的半监督聚类[J]. 软件学报, 2008, 19(11): 2803 – 2813.)

计算机应用 www.joca.cn