基于网格服务的GEP分布式函数挖掘算法

邓 松[1] 王汝传[1,2] 任勋益[1]
(南京邮电大学计算机学院 南京 210003)[1]
(南京大学计算机软件新技术国家重点实验室 南京 210093)[2]

摘 要 提出了一种基于网格服务的GEP分布式函数挖掘算法(DFMGEP-GS),它将网格服务与GEP算法相结合,既成功地实现了在网格平台下的GEP函数挖掘,又提高了每个网格节点上GEP算法的全局寻优性;同时证明了在网格环境下由局部数据模型生成全局数据模型的方法。仿真实验结果表明,对于函数类型已知的数据,随着数据集的增大,在成功挖掘到目标函数的情况下,DFMGEP-GS算法的平均耗时最少,而且随着网格节点的增加,DFMGEP-GS的收敛速度最大提高了约17倍;对于函数类型未知的复杂数据集,DFMGEP-GS算法挖掘所得到的模型的误差最小。

关键词 基因表达式编程,分布式挖掘,网格服务,函数挖掘
中图法分类号 TP393 **文献标识码** A

Distributed Function Mining for GEP on Grid Services

DENG Song[1] WANG Ru-chuan[1,2] REN Xun-yi[1]
(School of Computer, Nanjing University Post & Telecommunication, Nanjing 210003, China)[1]
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)[2]

Abstract This paper presented distributed function mining for GEP on grid services(DFMGEP-GS), which combined grid services and GEP algorithm to realize not only function finding for GEP on grid platform successfully, improve but also global optimization of GEP algorithm with every grid node. Meanwhile, it proved the method by which global data model is obtained by means of local data model on grid. By simulation experiment, it is showed that for data with known function type, and with the augmentation of datasets, average consumptive time of DFMGEP-GS is less than other three algorithms under the condition of mining target function successfully, and that with the increment of grid nodes, the convergent speed of DFMGEP-GS is improved about 17 times maximally. For very complex data with unknown function type, the error which is mined by DFMGEP-GS algorithm is minimum.

Keywords Gene expression programming, Distributed mining, Grid service, Function mining

1 引言

在实际应用中,对大量复杂的数据提炼数学模型,用变量和函数表示不同因素之间的依赖关系,有助于揭示复杂现象的内在本质。文献[1]在对图像内容进行选择、修改、锐化处理后,利用最小二乘法拟合处理这些图像所得到的数据之间的函数关系式,实验证明了该方法得到的曲线与实际曲线基本吻合。文献[2]在分析频响函数的理论值和功率谱平均估计值的误差函数的基础上,应用最小二乘法对频响函数的估计进行优化,并通过实验证明了该方法的有效性。为了解决传统的最小二乘法拟合曲面曲线的不足,文献[3]提出了基于移动最小二乘法的曲线曲面拟合方法,使得生成的曲线曲面精度高、光滑性好。但传统的回归方法一般都假定了函数类型已知,然后借助最小二乘法或者其改进的方法进行参数估计,最后确定函数关系式。上述这些方法过于依赖先验知识,含有很多主观人为因素,目前还不能解决复杂函数关系式的建立,而且对于复杂高维的样本数据,计算量较大,算法复杂度较高,计算效率低下。为此,在文献[4,5]中,作者使用遗传编程(GP)来进行数学建模,得到了比较好的结果,而且还避免了传统方法事先选定函数模型的不足。但是利用GP来挖掘函数模型,效率不高,文献[6]提出了一种新的算法——基因表达式编程(Gene Expression Programming, GEP),它和GP一样可以实现函数关系式的挖掘,而且作者在文献[6]中称其在解决复杂问题时,其效率高出传统的GA和GP算法4~6个数量级。文献[7-16]成功地将GEP以及改进的GEP算法应用于函数挖掘领域,取得了很好的效果。

到稿日期:2008-12-11 返修日期:2009-03-19 本文受国家自然科学基金(60573141和60773041),江苏省自然科学基金(BK2008451),国家高科技863项目(2006AA01Z201,2007AA01Z404,2007AA01Z478),现代通信国家重点实验室基金(9140C1105040805),江苏省高校自然科学研究计划(07KJB520083)/江苏省博士后基金(0801019C)和江苏高校科技创新计划项目(CX08B-085Z,CX08B-086Z)资助。

邓 松(1980-),男,博士生,主要研究方向为网格计算、数据挖掘等,E-mail:dsylc2006@yahoo.com.cn;王汝传(1943-),男,教授,博士生导师,主要研究方向为网格计算、计算机软件等;任勋益(1975-),男,博士,讲师,主要研究方向为网格计算、信息安全等。

以上这些算法都是在单机环境下挖掘函数关系的，而在实际应用中，很多数据集在地理上分布在不同的异构节点上，其中文献[16]提出的基于模拟退火的并行基因表达式编程算法经过修改就可以对分布式节点上的数据进行函数挖掘，但是 MPI 环境要求太高，而且基于 MPI 的编程过程很繁琐。在现有的文献中没有针对网格环境下利用 GEP 进行分布式函数挖掘的描述。鉴于在网格平台下实施数据挖掘算法的可行性，本文在文献[16]的基础上，结合网格服务的思想，提出了基于网格服务的 GEP 分布式函数挖掘算法(Distributed Function Mining for GEP on Grid Services，DFMGEP-GS)。

本文所做的主要工作为：(1)证明了网格环境下全局数据模型的存在。为了更好地汇聚局部数据模型，得到全局数据模型，文中提出了基于最小残差平方和的全局数据模型生成算法(Global Data Model on Minimum Residual Sum of Squares，GDM-MRSS)；(2)为了适应网格环境下分布式函数挖掘的需要，本文结合网格服务的思想，提出了基于网格服务的 GEP 分布式函数挖掘算法(Distributed Function Mining for GEP on Grid Services，DFMGEP-GS)；(3)进行了比较实验，并做了性能分析。

本文第 2 节介绍基于最小残差平方和的全局数据模型生成算法；第 3 节重点讲述 DFMGEP-GS 算法；第 4 节进行比较实验；最后进行总结。

2 基于最小残差平方和的全局数据模型生成算法

传统的分布式数据挖掘算法主要包括两个步骤：

(1) 局部数据分析，生成局部数据模型；

(2) 组合不同数据集上的局部数据模型，得到全局数据模型。

为了从局部数据模型得到全局数据模型，本文利用残差平方和最小的思想来融合在各个网格节点上挖掘得到的局部数据模型。在介绍算法思想之前，先看几个概念。

定义 1(k-维局部数据模型，k-Dimensions Local Data Model) 设有 n 个分布式节点，第 i 个节点上部署有 m_i，$i\in[1,n]$个样本数据，每一个样本数据都含有 $k+1$ 维属性，通过对每一个节点上的数据集进行函数挖掘分别得到 $f_i(X_1^{(i)},X_2^{(i)},\cdots,X_k^{(i)})$，其中 $X_j^{(i)}=(x_{1j}^{(i)},\cdots,x_{m_i j}^{(i)})^T$，$i\in[1,n]$，$j\in[1,k]$，则称 $f_i(X_1^{(i)},X_2^{(i)},\cdots,X_k^{(i)})$为第 i 个节点的 k-维局部数据模型。

为了简化问题描述和求解，现假设每个节点上的样本数据个数相等。

定义 2 设已知 n 个网格节点，每个节点上部署有 m 个样本数据，通过对每个节点上的数据分布式挖掘可以分别得到 n 个 k-维局部函数关系式：$f_1(X_1^{(1)},X_2^{(1)},\cdots,X_k^{(1)})$，$f_2(X_1^{(2)},X_2^{(2)},\cdots,X_k^{(2)})$，$\cdots$，$f_n(X_1^{(n)},X_2^{(n)},\cdots,X_k^{(n)})$，$\exists$ 常数 a_i，$i\in[1,n]$，使得函数表达式

$$f(X_1,X_2,\cdots,X_k)=a_1f_1(X_1^{(1)},X_2^{(1)},\cdots,X_k^{(1)})+a_2f_2(X_1^{(2)},X_2^{(2)},\cdots,X_k^{(2)})+\cdots+a_nf_n(X_1^{(n)},X_2^{(n)},\cdots,X_k^{(n)}) \quad (1)$$

成立，称该函数表达式为全局数据模型。

按照分布式挖掘传统步骤，最后的目的就是要通过局部数据模型生成实际所需要的全局数据模型。在网格平台下，首先将生成的 n 个局部数据模型传到网格客户端，然后由 n 个 k-维局部数据模型生成 p 组 $k+1$ 维测试数据，由定义 2 可知要最后得到所需要的全局数据模型，就要求解出一组常数 a_i，$i\in[1,n]$，使得 n 个 k-维局部数据模型的线性组合成立，从而使得残差平方和 $RSS=\sum_{i=1}^{p}[(f(X_1,X_2,\cdots,X_k)-x_{ik+1})]^2$ 最小，其中 p 表示数据集共有 p 条数据，$p>>n$。为了求解 a_i，$i\in[1,n]$的值，构建函数 $g(a_1,a_2,\cdots,a_n)=\sum_{i=1}^{p}[(f(X_1,X_2,\cdots,X_k)-x_{ik+1})]^2$，从而问题就转化为求解 a_i，$i\in[1,n]$使得函数 $g(a_1,a_2,\cdots,a_n)$值最小。也就是说，此时求解的全局数据模型能充分反映所有样本数据。

引理 1 若 $f_1(X_1^{(1)},X_2^{(1)},\cdots,X_k^{(1)})$，$f_2(X_1^{(2)},X_2^{(2)},\cdots,X_k^{(2)})$，$\cdots$，$f_n(X_1^{(n)},X_2^{(n)},\cdots,X_k^{(n)})\in k$ 维空间中的函数表达式，则 $f(X_1,X_2,\cdots,X_k)$为此 n 个 k 维函数表达式的线性组合，系数为 a_i，$i\in[1,n]$。总 $\exists$ 常数 a_i，$i\in[1,n]$使得函数 $g(a_1,a_2,\cdots,a_n)=\sum_{i=1}^{p}[(f(X_1,X_2,\cdots,X_k)-x_{ik+1})]^2$ 的值最小。

设已知 n 个局部数据模型，p 组测试样本数据，则构建测试样本数据矩阵

$$\begin{pmatrix} x_{11} & \cdots & x_{1k}x_{1k+1} \\ \vdots & \ddots & \vdots \\ x_{p1} & \cdots & x_{pk}x_{pk+1} \end{pmatrix} \quad (2)$$

其中，该测试数据包含 $k+1$ 个属性。前 k 个属性为输入属性，最后一个为输出属性。x_{ij} 表示第 i 个数据的第 j 个属性的数值，其中 $i\in[1,p]$，$j\in[1,k+1]$，则把式(1)代入函数 $g(a_1,a_2,\cdots,a_n)=\sum_{i=1}^{p}[(f(x_{i1},x_{i2},\cdots,x_{ik})-x_{ik+1})]^2$，可得

$$g(a_1,a_2,\cdots,a_n)=\sum_{i=1}^{p}([a_1f_1(x_{i1},\cdots,x_{ik})+\cdots+a_nf_n(x_{i1},\cdots,x_{ik})]-x_{ik+1})^2 \quad (3)$$

由已知条件可知，此时函数 $g(a_1,a_2,\cdots,a_n)$是关于$(a_1,a_2,\cdots,a_n)$的一个 n 维的二次多项式，且连续可微。由导数定义可知，要使函数 $g(a_1,a_2,\cdots,a_n)$的值最小，只要用该函数分别对$(a_1,a_2,\cdots,a_n)$求二阶导数，若其二阶导数大于 0，说明最小值存在，则令一阶导数等于 0，求解$(a_1,a_2,\cdots,a_n)$即可。

证明：把式(2)中的每一行数据代入式(3)可得

$$\begin{aligned} g(a_1,a_2,\cdots,a_n)=&([a_1f_1(x_{11},\cdots,x_{1k})+a_2f_2(x_{11},\cdots,x_{1k})+\cdots+a_nf_n(x_{11},\cdots,x_{1k})]-x_{1k+1})^2+\cdots+([a_1f_1(x_{p1},\cdots,x_{pk})+\\ &a_2f_2(x_{p1},\cdots,x_{pk})+\cdots+a_nf_n(x_{p1},\cdots,x_{pk})]-x_{pk+1})^2 \end{aligned} \quad (4)$$

此时所有的 $f_i(x_{j1},\cdots,x_{jk})$，$i\in[1,n]$，$j\in[1,p]$都为常数。令 $f_1(x_{11},\cdots,x_{1k})=C_{11}$，$f_2(x_{11},\cdots,x_{1k})=C_{21}$，$\cdots$，$f_n(x_{11},\cdots,x_{1k})=C_{n1}$，$\cdots$，$f_1(x_{p1},\cdots,x_{pk})=C_{1p}$，$f_2(x_{p1},\cdots,x_{pk})=C_{2p}$，$\cdots$，$f_n(x_{p1},\cdots,x_{pk})=C_{np}$，代入式(4)，可得

$$\begin{aligned} g(a_1,a_2,\cdots,a_n)=&([a_1C_{11}+a_2C_{21}+\cdots+a_nC_{n1}]-x_{1k+1})^2+([a_1C_{1p}+a_2C_{2p}+\cdots+\\ &a_nC_{np}]-x_{pk+1})^2 \end{aligned} \quad (5)$$

又因为函数 $g(a_1,a_2,\cdots,a_n)$是关于$(a_1,a_2,\cdots,a_n)$的一个 n 维的二次多项式，且连续可微，所以函数 $g(a_1,a_2,\cdots,a_n)$对$(a_1,a_2,\cdots,a_n)$的导数都存在

$$\frac{\partial g(a_1,a_2\cdots,a_n)}{\partial a_1}=2a_1\sum_{i=1}^{p}C_{1i}^2+2a_2\sum_{i=1}^{p}C_{1i}C_{2i}+\cdots+2a_n\sum_{i=1}^{p}C_{1i}C_{ni}-2\sum_{i=1}^{p}C_{1i}C_{ik+1} \quad (6)$$

$$\frac{\partial g(a_1,a_2\cdots,a_n)}{\partial a_1^2}=2\sum_{i=1}^{p}C_{1i}^2>0 \tag{7}$$

同理可得，$g(a_1,a_2,\cdots,a_n)$关于$(a_2,\cdots,a_n)$的二阶导数都大于0，故原函数的最小值存在。依此类推，得

$$\frac{\partial g(a_1,a_2\cdots,a_n)}{\partial a_n}=2a_n\sum_{i=1}^{p}C_{ni}^2+2a_1\sum_{i=1}^{p}C_{1i}C_{ni}+\cdots+2a_{n-1}\sum_{i=1}^{p}C_{n-1i}C_{ni}-2\sum_{i=1}^{p}C_{ni}C_{ik+1} \tag{8}$$

分别令其等于0，即可得n个方程

$$\begin{pmatrix}\sum_{i=1}^{p}C_{1i}^2 & \cdots & \sum_{i=1}^{p}C_{1i}C_{ni}\\ \vdots & \ddots & \vdots\\ \sum_{i=1}^{p}C_{1i}C_{ni} & \cdots & \sum_{i=1}^{p}C_{ni}^2\end{pmatrix}\begin{pmatrix}a_1\\ \vdots\\ a_n\end{pmatrix}=\begin{pmatrix}\sum_{i=1}^{p}C_{1i}C_{ik+1}\\ \vdots\\ \sum_{i=1}^{p}C_{ni}C_{ik+1}\end{pmatrix} \tag{9}$$

再令$A=\begin{pmatrix}\sum_{i=1}^{p}C_{1i}^2 & \cdots & \sum_{i=1}^{p}C_{1i}C_{ni}\\ \vdots & \ddots & \vdots\\ \sum_{i=1}^{p}C_{1i}C_{ni} & \cdots & \sum_{i=1}^{p}C_{ni}^2\end{pmatrix}$，$X=\begin{pmatrix}a_1\\ \vdots\\ a_n\end{pmatrix}$，

$B=\begin{pmatrix}\sum_{i=1}^{p}C_{1i}C_{ik+1}\\ \vdots\\ \sum_{i=1}^{p}C_{ni}C_{ik+1}\end{pmatrix}$，式(9)转换为$AX=B$。又因挖掘函数的随机性，总能保证$(n-1)\leqslant R(A)\leqslant n$，故方程组(9)总有解。由高斯消元法求解线性方程组(9)，可得$(a_1,a_2,\cdots,a_n)$。原命题得证。

根据引理1，本文提出了基于最小残差平方和的全局数据模型生成算法(Global Data Model on Minimum Residual Sum of Squares，GDM-MRSS)，整个算法的形式化描述如算法1所示。

算法1 基于最小残差平方和的全局数据模型生成算法(Global Data Model on Minimum Residual Sum of Squares，GDM-MRSS)

Input：n个局部数据模型：$f_i(X_1^{(i)},X_2^{(i)},\cdots,X_k^{(i)})$，其中

$X_j^{(i)}=(x_{1j}^{(i)},\cdots,x_{m_ij}^{(i)})^T,i\in[1,n],j\in[1,k]$

Output：全局数据模型$f(X_1,X_2,\cdots,X_k)$

Begin {

1)将n个局部数据模型传送到网格客户端；

2)由n个局部数据模型生成p组$k+1$维的测试样本数据，同时构建式(3)；

3)把测试样本数据代入式(3)中计算残差平方和，化简得式(5)；

4)对于求得的函数$g(a_1,a_2,\cdots,a_n)$分别对$(a_1,a_2,\cdots,a_n)$求一阶导数，并令求导结果等于0，得到n个关于$(a_1,a_2,\cdots,a_n)$的线性方程组(9)；

5)利用高斯消元法求解线性方程组(9)分别得到$a_1,a_2,\cdots,a_n$；

6)把$a_1,a_2,\cdots,a_n$代入式(1)，并返回$f(X_1,X_2,\cdots,X_k)$；}

End

整个GDM-MRSS算法的时间基本上都消耗在利用高斯消元法求解线性方程组上，故整个算法的时间复杂度约为$O(n^3)$。

3 基于网格服务的GEP分布式函数挖掘算法

根据引理1，在网格平台下，可以很好地由局部模型得到全局数据模型，解决了分布式函数挖掘的关键问题。再者，为了减少分布式数据集中进行函数挖掘所带来的通信带宽、存储以及计算能力的限制，本文在文献[16]所提出的算法的基础上，结合网格服务的思想来构建分布式的GEP函数挖掘平台，提出了基于网格服务的GEP分布式函数挖掘算法(Distributed Function Mining for GEP on Grid Services，DFMGEP-GS)。

3.1 算法思想

在分布式环境下，数据分布式存储在不同的节点上，如果把不同节点的数据集中传到指定的计算机上再进行函数挖掘，由于受到通信带宽和传输速度的影响，挖掘效率低下，而且对于复杂非线性数据集，函数挖掘的成功率较低。而网格正好提供了一个分布式计算的平台，利用网格软件把各分布式节点连接起来，既可以共享各个节点的计算和存储资源，也可以方便分布式数据的挖掘，这对于节省网络带宽，提高数据处理能力都有极大的优势。

数据的分布式有两种情况：一是待处理的数据本身就处于不同的节点上；二是待处理的海量数据需要事先存储到不同的节点上。本文重点讨论第二种情况。为了简单起见，DFMGEP-GS算法中的分布式数据集是按照水平划分的，也就是说每一个节点上的数据集所包含的数据属性一致。整个DFMGEP-GS算法的工作思路为：首先通过知识网格服务层来获取数据资源的定位和有关计算节点的信息，然后利用知识网格核心层调用相应的GEP算法服务来挖掘相应的数据集，最后对各个分节点挖掘得到的局部函数关系式聚合成一个线性组合，得到最终的函数表达式，并且通过网格应用层把该函数表达式显示给用户。

3.2 算法描述

在网格节点上运行DFMGEP-GS算法，主要需做两部分工作：第一要开发算法网格服务端；第二就是开发网格客户端调用算法网格服务。下面具体给出算法网格服务端和客户端的描述。

首先看DFMGEP-GS算法的服务端情况。为了提高算法的全局寻优性，采用了文献[16]中的基于模拟退火的GEP函数挖掘算法，将模拟退火与传统的遗传操作相结合，以提高GEP算法跳出局部最优的能力。整个算法的步骤如算法2所示。

算法2 基于模拟退火的GEP函数挖掘(Function Mining for GEP based on Simulated Annealing)

Input：样本数据集 SampleSets

Output：挖掘得到的函数表达式 FuncExpression

Begin {

1)初始化控制参数：最大适应度值 MaxFitness、最大进化代数 MaxGen 等；

2)While (fitness<=MaxFitness) or (gen<=MaxGen) {

3) 计算当前种群中每一个个体的适应度函数值 fitness；

4) 对当前种群中的每一个个体，执行模拟退火操作，从而产生局部最优个体组成新的种群；

5) 再对新产生的种群进行各种遗传操作(选择、插串、变异以及重组等)；

6) 计算新种群中每一个个体的适应度函数值 fitness；

7) 保留最佳个体；}}

8)return FuncExpression；//返回挖掘得到的最优的函数表达式。

整个算法的运行主要是由运行代数 gen 以及种群中每一

个个体执行模拟退火操作的总迭代次数 N 来控制，故整个算法的时间复杂度约为 O(gen * N)。

其次看 DFMGEP-GS 算法的网格客户端的运行过程。为了便于问题描述，整个客户端算法的形式化描述如算法 3 所示。

算法 3 基于网格服务的 GEP 分布式函数挖掘算法(Distributed Function Mining for GEP on Grid Services, DFMGEP-GS)

Input：样本数据集 SampleSets[n]

Output：全局数据模型 $f(x_1,x_2,\cdots,x_k)$；

Begin{

1)根据用户提供给知识网格高层的信息，调用知识网格核心层的服务和工具，得到所需数据库资源，从数据库资源中提取样本数据，申请核心层为该样本数据分配计算节点，并通过数据管理将样本数据分配到相应的计算节点上；

对于网格平台 n 个节点中的第 i 个节点；

2)由知识网格核心层调用算法 2；

3)保存第 i 个节点上所生成的最优局部数据模型并传送到网格客户端；

4)对于每一个网格节点并行运行 2),3)两步；

5)最后再由知识网格核心层调用算法 1 的网格服务，产生全局数据模型；

6)由知识网格高层将挖掘整个样本数据所得到的全局函数表达式的结果可视化地表达出来；}

End

整个 DFMGEP-GS 算法利用网格服务的思想来对不同节点上的数据并行进行分布式函数挖掘，从而可以大大提高挖掘函数关系的效率和成功率。

4 仿真实验和分析

为了验证 DFMGEP-GS 算法的有效性，在实验室局域网环境下做了仿真实验。整个试验平台为 Windows 2000＋WS-Core-4.0.2＋Jdk1.5＋Eclipse3.1＋Tomcat5.17，所有的程序由 Java 语言实现。

实验 1 从满足函数关系式 $y=\frac{\sin(a)\cos(b)}{\sqrt{\exp^c}}+\tan(d-e)$ 的样本数据中挖掘该函数表达式，随机产生 1000，10000，100000 条数据，分别使用 GEP、文献[16]中所提出的 GEPSA 以及并行 GEPSA，DFMGEP-GS 算法进行函数挖掘，4 种算法各运行 50 次。在运行 DFMGEP-GS 算法之前，把数据集平均分到 3 个网格节点上运行。图 1 比较了 4 种算法的平均耗时；图 2 则显示了随着网格节点个数的增加 DFMGEP-GS 算法的平均收敛代数的变化情况。

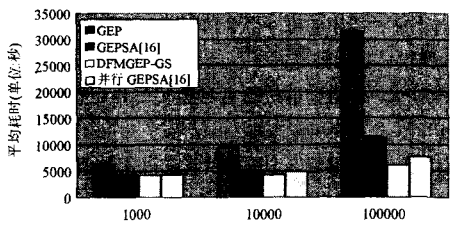

图 1 4 种算法的平均耗时比较

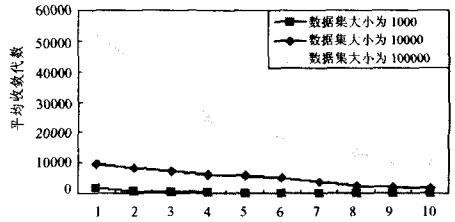

图 2 DFMGEP-GS 算法在不同网格节点个数下的收敛速度比较

图 1 表明，对于不同大小的数据集，其它 3 种算法的平均耗时都要明显优于 GEP 算法。随着数据集的不断增大，在成功挖掘到目标函数的情况下，DFMGEP-GS 比 GEPSA[16] 的平均耗时减少约 2 倍，比并行 GEPSA[16] 减少约 0.5 倍。这是因为对于处理大而复杂的非线性数据集而言，GEPSA[16] 每次计算适应度都要遍历整个样本数据集，故比 DFMGEP-GS 算法中每个节点处理一部分数据的计算量要大。而对于文献[16]中提出的为提高算法挖掘函数效率的并行 GEPSA 算法，由于每次都要在各个计算节点中计算和交换最优个体，故也比 DFMGEP-GS 算法中最后一次传送最优个体到网格客户端要费时。而且从图 1 中还可以看出，随着数据集的增大，由于考虑到各个网格节点传送数据模型到网格客户端以及传输延迟的影响，DFMGEP-GS 算法的平均耗时并没有下降，而是有所增加。图 2 表明了随着网格节点个数的增加，对于不同大小的数据集而言，DFMGEP-GS 算法的收敛速度明显提高。对于相同的数据集，DFMGEP-GS 算法的收敛速度最大提高了约 17 倍。并在实际进化过程中，DFMGEP-GS 算法在每一个节点上经过若干代进化得到的函数关系式再根据引理 1 求解化简，可得到全局函数关系式

$$y=\cos\sqrt{\sin(c)}(\cos(b)\sin(a))+\tan(d-e) \tag{10}$$

式(10)与文献[6]的相同。在文献[6]中证明了式(10)近似为 $y=\frac{\sin(a)\cos(b)}{\sqrt{\exp^c}}+\tan(d-e)$，从而证明了在网格环境下 DFMGEP-GS 算法是有效可行的。

实验 2 由于实验 1 的实验数据满足一个固定的函数关系式，但是在实际情况中，不可能所有的或者绝大部分数据都满足一个固定的函数关系式，那么每一次挖掘得到的函数关系式都有可能不同。此时，为了说明 DFMGEP-GS 算法的高效性能，必须和实验数据的实际输出进行比较，计算其平均误差，亦即挖掘得到的模型与实际数据的吻合程度。而对于挖掘 UCI 标准数据集 waveform-5000 的函数关系式来说，这是一个非常艰巨的任务。该数据集包含 40 个输入属性、3 个分类输出属性，数据集大小为 5000。挖掘该数据集得到的数学模型，既可以分析数据之间存在的内在联系，也可以作为将数据分类的依据。在运行 DFMGEP-GS 算法之前，把 waveform-5000 数据集平均分到 3 个网格节点上运行。图 3 比较了 GEP，GEPSA[16] 以及并行 GEPSA[16]，DFMGEP-GS 算法的耗时情况，图 4 则显示了随着网格节点个数的增加 DFMGEP-GS 算法的平均收敛代数的变化情况。同时为了说明 DFMGEP-GS 挖掘复杂非线性函数的有效性，表 1 比较了 GEP、文献[16]中所提出的 GEPSA 以及并行 GEPSA，DFMGEP-GS 这 4 种算法挖掘所得模型的平均绝对误差和最

大绝对误差。其中 DFMGEP-GS 算法最后挖掘得到的全局模型为

$$2.15\times\cos(\sin(\sin(\sin(\frac{e^m}{d+f+i}))))+1.564\times\sin(\sin(\sin(\cos(\sin(n)+\sin(q)))))+6.32\times\sin(\sin(\sin(\sin(\frac{-f-g+m}{d+j}))))$$

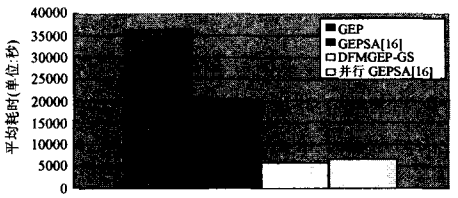

图 3　挖掘 waveform-5000 数据集时 4 种算法的平均耗时比较

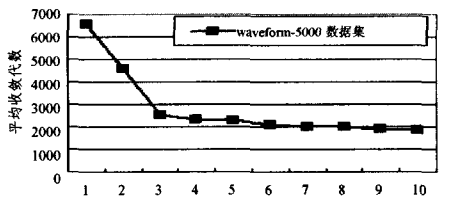

图 4　DFMGEP-GS 算法在不同网格节点个数下的收敛速度

表 1　3 种模型的误差对比

模型	平均绝对误差	最大绝对误差
GEP	15.32	65.34
GEPSA[16]	12.29	52.87
并行 GEPSA[16]	11.03	41.93
DFMGEP-GS	10.37	30.28

图 3 显示了对于实际函数类型未知的海量复杂非线性数据集而言，在挖掘到理想函数模型的基础上，DFMGEP-GS 算法的平均耗时比 GEP 以及 GEPSA[16] 要少得多，而与文献[16]中的并行 GEPSA 算法相当。但是网格平台下开发分布式并行 GEP 算法，只要把算法打包成网格服务发布到网格环境中即可，比基于 MPI 环境下开发要简单，而且增强了算法本身的可扩展性。图 4 则表明了 DFMGEP-GS 算法在挖掘实际数据集时，随着网格节点的增加，收敛速度明显加快。在实验室局域网环境下，当网格节点数为 7 时，DFMGEP-GS 算法的收敛速度基本已经达到收敛状态。这是因为针对特定大小的数据集而言，随着网格节点数的增加，平均分配到各节点的数据量下降，每个节点的算法比较容易就可以挖掘到符合小数据量的局部函数关系；当网格节点数达到一个数量时，由于数据量较小，算法导致挖掘局部函数关系的收敛速度基本没有变化。当然也不是把数据分到越多的网格节点上执行就越好，因为针对 GEP 挖掘特定的数据集而言，每个网格节点上的样本数据太少，导致最终挖掘得到的全局函数关系难以拟合大部分样本数据。而且由定义 2 知，最后得到的全局函数关系式也较为复杂。具体要把特定大小的数据分到多少网格节点上运行，本文没有做具体的考虑，这也是以后的研究方向。表 1 则表明了 4 种算法挖掘所得的模型预测的结果与实际输出的比较，可见 DFMGEP-GS 算法挖掘所得模型的平均绝对误差和最大绝对误差均为最小。

结束语　本文将网格服务的理念应用于 GEP 函数挖掘中，提出了一种基于网格服务的 GEP 分布式函数挖掘算法(DFMGEP-GS)，并证明了由局部数据模型得到全局数据模型的方法。仿真比较实验表明，在挖掘函数关系中，无论是虚拟数据还是真实数据，DFMGEP-GS 算法都比传统的 GEP，GEPSA[16] 以及并行 GEPSA[16] 的耗时要少。而且对于真实数据，使用 DFMGEP-GS 算法挖掘得到的模型的误差最小。

参 考 文 献

[1] 蒋华伟，张庆州. 基于图像曲线和最小二乘法进行函数拟合的研究[J]. 计算机应用与软件，2007，24(7)：43-44

[2] 陆冬，汤宝平，何启源，等. 模态参数识别中频响函数估计的最小二乘优化[J]. 重庆大学学报：自然科学版，2007，30(3)：6-10

[3] 曾清红，卢德唐. 基于移动最小二乘法的曲线曲面拟合[J]. 工程图学学报，2004，1：84-89

[4] 李敏强，寇纪淞，林丹，等. 遗传算法的基本理论与应用[M]. 北京：科学出版社，2002：336-343

[5] Koza J R. Genetic Programming II：automatic discovery of reusable programs[M]. MIT Press，1994

[6] Ferreira C. Gene Expression Programming：Mathematical Modeling by an Artificial Intelligence (2nd Edition) [M]. Springer，May 2006

[7] 贾晓斌，唐常杰，左劼，等. 基于基因表达式编程的频繁函数集挖掘[J]. 计算机学报，2005，28(8)：1247-1254

[8] 段磊，唐常杰，左劼，等. 基于基因表达式编程的抗噪声数据的数据挖掘方法[J]. 计算机研究与发展，2004，41(10)：1684-1689

[9] 黄晓冬，唐常杰，李智，等. 基于基因表达式编程挖掘函数关系[J]. 软件学报，2004，15(增刊)：96-105

[10] 胡建军，唐常杰，段磊，等. 基因表达式编程初始化种群的多样化策略[J]. 计算机学报，2007，30(2)：305-310

[11] 元昌安，唐常杰，左劼，等. 基于基因表达式编程的函数挖掘—收敛性分析与参差制导进化算法[J]. 四川大学：工程科学版，2004，36(6)：100-105

[12] Yuan Chang-an，Tang Chang-jie，Zuo Jie，et al. Attribute Reduction Function Mining Algorithm Based on Gene Expression Programming[C]//Proceedings of 2006 International Conference on Machine Learning and Cybernetics(ICMLC 2006). Dalian，China，August 2006：1007-1012

[13] Ferreira C. Gene Expression Programming：A New Adaptive Algorithm for Solving Problems [J]. Complex Systems，2001，13(2)：87-129

[14] Ferreira C. Automatically defined functions in Gene Expression Programming [J]. Genetic Systems Programming：Theory and Experiences，Studies in Computational Intelligence，13：21-56

[15] Ferreira C. Function Finding and the Creation of Numerical Constants in Gene Expression Programming [J]. Advances in Soft Computing- Engineering Design and Manufacturing，Spring-Verlag，2003：257-266

[16] 蒋思伟，蔡之华，曾丹，等. 基于模拟退火的并行基因表达式编程算法研究[J]. 电子学报，2005，33(11)：2017-2021