维普资讯 http://www.cqvip.com

最佳模型拟合的并行算法*)

梁 立 肖 飞 郜 松
(云南师范大学计算机科学与信息学院 昆明 650092)

摘 要 提出最佳函数模型数据拟合的并行算法。每个处理机用不同的函数模型拟合相同的数据,再由主机根据每种函数模型的残差选择最佳的函数模型。该算法不需要事先给出函数模型而由计算机盲目搜索。

关键词 数据拟合,函数集,函数模型,离散数据,并行算法

Parallel Algorithm of Data Fitting with the Best Function Model

LIANG Li XIAO Fei GAO Song
(School of Computer Science and Information Technology, Yunnan Normal University, Kunming 650092)

Abstract The parallel algorithm of data fitting with the best function model is proposed. The same data are fitted by different function models in different CPU, and then Host chooses the best function model according to errors.

Keywords Data fitting, Function set, Function model, Disperse data, Parallel algorithm

在数据拟合中,首先需要确定一个函数模型,再确定函数模型的待定参数,最后得到一个逼近函数。函数模型的确定,通常根据离散数据的规律或解决问题的经验,或者利用高次多项式作为近似函数模型。函数模型的确定是数据拟合的关键,也是数据拟合困难之所在。如果利用计算机搜索最适合的函数模型,将为数据拟合奠定基础。

文[1]中是将数据划分到各个处理机上进行相同模型拟合,再处理边界。本文是将相同的数据全部传给每个处理机,每个处理机用不同的模型进行拟合。

文[2]研究了神经网络在函数逼近中的应用,可以用于每个处理机的拟合算法中。

给出一个函数集合,由该集合中的函数生成一些函数模型,从其中找一个最好的函数模型进行数据拟合。但生成的函数模型的数量是惊人的,如果以线性组合的方式生成函数模型,也将是 2^n 规模(n 是函数集合的函数个数),处理是困难的。

利用并行算法,每个处理机用一个函数模型进行拟合,把函数模型的逼近程度反馈给主机,再由主机选择一个逼近程度最好的函数模型,最后用这个最好的函数模型进行数据拟合。

1 函数模型的生成

事先给定一个函数集合,由该集合中的元素生成不同的函数模型。如果函数集合中的元素是线性无关的,可利用线性组合方式生成函数模型,而且可以类似最小二乘法拟合函数模型,但在实际问题中往往是苛刻的。

把几类典型的初等函数放在一个集合中,例如 $S=\{1,x,x^2,\sin x,\sin x,e^x,\ln x\}$,如果用线性组合方式生成函数模型,算法较简单,但可能得不到最精确的函数模型,如果用任意组合方式,函数模型的数量将是无穷的。

利用遗传算法随机生成函数模型,能有效得到非常好的函数模型。实验表明,一般情况下,线性组合方式也能得到较好的函数模型,但比遗传算法简单得多。

用线性组合方式生成函数模型,需要设计较好的初等函数集合。例如,除 $\sin x$ 外还需要 $\sin(\omega x+\phi)$,除 e^x 外还需要 e^{ax+b} 等等,总之,凡可能涉及的函数形式都有必要放入函数集合中。当然也可根据问题本身的规律选择最有可能的函数,以减少函数集的规模。

2 最优函数模型的选择

用逼近函数值 $f(x_i)$ 与原始函数值 y_i 之间的差(残差)$\delta_i=y_i-f(x_i)$ 衡量逼近程度的标准。常用的衡量准则有以下三种[3]:

(1) 残差绝对值之和最小:$\Sigma|\delta_i|=\min$

(2) 残差的最大绝对值最小:$\max|\delta_i|=\min$ (最佳一致逼近)

(3) 残差的平方和最小:$\Sigma\delta_i^2=\min$(最佳平方逼近)

理论上使用第三种准则最方便讨论,但程序计算时,第一种计算也不复杂。不管用哪一种准则数据拟合得到的逼近函数都有残差,每个处理机用相同的标准反馈给主机,选残差(残差之和、残差的最大绝对值或者残差的平方和)最小者作为最优函数模型。

3 最优数据拟合的并行算法

主机由函数集合生成多个函数模型分别提交给不同的处理机,每个处理机用相同的逼近准则拟合同一批离散数据,得到一个逼近函数,把这个逼近函数的残差反馈给主机,由主机选择最优函数模型,再由主机用最优函数模型进行最终数据拟合,如图 1 所示。

算法的耦合度非常低,可以大大减少通信数据量,利用通信速度相对较慢的 PC 集群系统实现也较理想[4]。如果处理机数量比函数模型少,主机可循环发送剩下的函数模型给完成任务的处理机继续拟合,直到函数模型完全处理结束,主机才进行选择最佳者,最后由主机再拟合最佳函数(最后的拟合

*)基金项目:云南省教育厅科研基金资助项目(5Y1053D)。梁 立 副教授,研究方向:算法分析,并行计算。

维普资讯 http://www.cqvip.com

主机做了子进程的重复工作，这是为了减少通信量）。

算法的通信量主要有以下几部分开销：主机一次性广播离散数据给每个处理机，然后分别发送不同的函数模型，处理机每拟合一个模型后把残差反馈给主机。

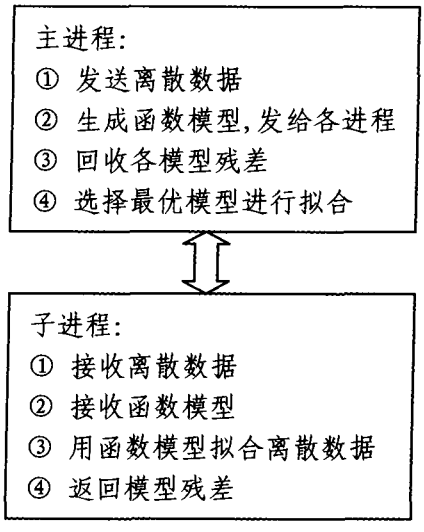

图 1　并行算法工作图

4　最优数据拟合实例

有离散数据如表 1[3]：

表 1　离散数据表

x_i	0.2	0.3	0.4	0.5	0.6	0.7	0.8
y_i	3.16	2.38	1.75	1.34	1.00	0.74	0.56

把这些点依次连接起来如图 2 所示。

从图 2 的形态看，可以视为二次函数或指数函数；如果考虑数据的误差，将其视为直线也无妨。这里讨论函数形态只是为了尽量减少函数模型的数量，在实际应用时完全没有必要人为限制函数模型的范围，可以盲目搜索，但花费更多的 CPU 时间是盲目搜索的代价。

这三个函数模型由三个处理机都使用最佳平方逼近拟合这些数据，结果如表 2。

从表 2 可以得到，三个最佳平方逼近函数中，残差的平方和最小者为指数函数，因此，主机在这三个函数模型中选择指数模型，得到最佳逼近函数 $f(x)=5.636e^{-2.891x}$。

当然，如果函数模型再多一些，可选择的函数模型就更广一些，也就可能得到更精确的逼近函数。

每个处理机上处理一个函数模型时，可调用现成的拟合软件，比如 Matlab 就是比较理想的拟合平台。本文并行算法实验使用了 MPI 编程调用 Matlab。

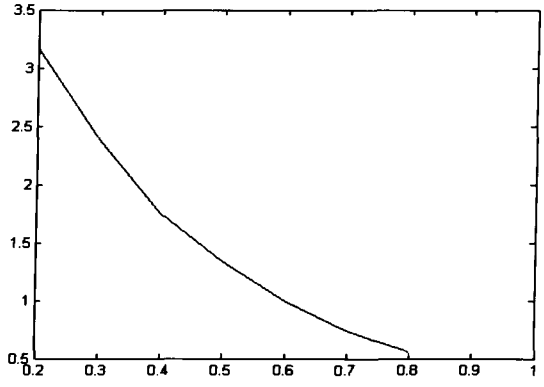

图 2　离散数据的折线图

表 2　三种函数拟合的比较

函数模型	最佳平方逼近函数 $f(x)$	残差的平方和
一次函数 $y=ax+b$	$f(x)=-4.225x+3.674$	0.3047
二次函数 $y=ax^2+bx+c$	$f(x)=5.94x^2-10.17x+4.921$	8.3×10^{-3}
指数函数 $y=ae^{bx}$	$f(x)=5.636e^{-2.891x}$	8.9447×10^{-4}

参 考 文 献

1　章勤，李品，肖舸. 海量数据的曲线拟合并行算法及实现[J]. 华中理工大学学报，2000，28(10)：85～87

2　翟振华，李凯. 神经网络在函数逼近问题中的应用研究[J]. 计算机工程，2001(5)

3　丁丽娟. 数值计算方法[M]. 北京：北京理工大学出版社，2001

4　都志辉，等. 高性能计算并行编程技术[M]. 北京：清华大学出版社，2001

（上接第 293 页）

DB2，MYSQL 等常用数据库平台的通用类型定义文件；

2）按系统模块划分，用 5 个实体配置文件 entitymodel_stock. xml、entitymodel_store. xml、entitymodel_dispatch. xml、entitymodel_lease. xml、entitymodel_other. xml 分别来描写采购、库存、调拨、租赁四个模块以及其他的一些实体；

3）根据实体配置文件，对 entitygroup. xml 文件进行相应的修改，即添加新定义的实体到 entitygroup. xml 文件；

4）配置 entityengine. xml 文件，主要是实体文件名称和数据库连接串参数，包括数据库访问地址，数据库名称，用户名，密码等。

在做好以上配置后，在以后的编程中就可以直接访问实体配置文件进行数据的访问，而如何完成这一过程则由实体引擎处理。特别是当用户的数据模型需要修改时，需要修改的也仅仅是实体配置文件，这就极大地减轻了程序的维护工作量。

结束语　OFBiz 框架是一种非常优秀的基于 J2EE 的 MVC 应用框架，尽管它试图把所有的基础、公共和特殊应用都涵盖进来，增加了其复杂性，阻碍了 OFBiz 的应用，但其实体引擎的思想及其所带来的良好的异构数据库间的可移植性仍然值得系统分析设计人员的学习。本文在总结了 OFBiz 框架技术及其实体引擎工作原理的基础上，结合"网络化物流管理系统"项目，真正实现了平台无关应用程序的构建，实践证明取得了良好的效果。

参 考 文 献

1　http://sourceforge.net/project/ofbiz

2　http://incubator.apache.org/ofbiz/

3　冷山述，余一，陆倜. 企业信息化框架 OFBiz 的研究. 北京机械工业学院学报，2004(3)

4　王晓楠. MVC 的设计与实现. 计算机系统应用，2004(3)