国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第15卷第2期 1993年6月 Vol. 15 No. 2

C_p 准则下选择最优子集的并行算法*

胡庆军 吴 翊
(系统工程与应用数学系)

摘 要 C_p 准则[1]是目前颇受重视的一种变量选择准则。本文针对大型线性回归模型，推导了从所有可能子集中用 C_p 准则选择最优子集的（乘除法）运算次数，提出了 C_p 准则下变量选择的并行算法。给出了在 YH－1 和 YH－2 向量巨型计算机上运行该算法的模拟结果且获得了 15 倍左右的向量加速比 s/v，体现了该算法的优越性。

关键词 线性回归模型，变量选择准则，子集回归，并行计算，向量加速比
分类号 O212. 1

考虑含 k 个自变量的线性回归模型

$$Y = X\beta + e \tag{1}$$

$$E(e) = 0, \mathrm{cov}(e,e) = \sigma^2 I \tag{2}$$

式中 $Y_{n\times 1}$，$X_{n\times k}$均为已知矩阵且 X 列满秩，$\beta_{k\times 1}$为未知参数向量，$e_{n\times 1}$为不可测的误差向量，σ^2 未知，$I_{n\times n}$为单位矩阵。当模型含有常数项时，本文约定 X 的第一列元素皆为 1，这时作变量选择总是把常数项选入回归模型。对设计矩阵 X 分块

$$X = (X_p \vdots X_t), \beta = (\beta_p' \vdots \beta_t)'$$

$$Y = X_p\beta_p + X_t\beta_t + e \tag{3}$$

式中 $p+t=k$，X_p 为 X 的 p 列，X_t 为其余 t 列，即(1)与(3)等价，β_p 和 β_t 为对应的分块。如果 $X_t\beta_t$ 对 Y 影响很小或根本没有影响，(3)式近似为

$$Y = X_p\beta_p + e \tag{4}$$

记

$$\hat{\beta}_{LS} = (X'X)^{-1}X'Y \tag{5}$$

$$\hat{\beta}_{PLS} = (X_p'X_p)^{-1}X_p'Y \tag{6}$$

$$\mathrm{RSS} = Y'(I - P_X)Y \tag{7}$$

$$\mathrm{RSS}_p = Y'(I - P_{X_p})Y \tag{8}$$

$$\hat{\sigma}^2 = \frac{\mathrm{RSS}}{n-k}, p_X = X(X'X)^{-1}X', p_{X_p} = X_p(X_p'X_p)^{-1}X_p' \tag{9}$$

$$C_p = \frac{\mathrm{RSS}_p}{\hat{\sigma}^2} + 2p - n \tag{10}$$

称(1)或(3)为全模型，(4)为选模型，X_p 为对应的自变量子集的设计矩阵，所有可能子集

* 1991 年 7 月 16 日收稿

回归(简称子集)有 2^k-1 个。(1)、(4)对应的 β、β_p 的最小二乘估计(简记为 LSE)分别为(5)、(6)式,相应的残差平方和分别为(7)、(8)式。C_p[1](变量选择)准则是指:以 C_p 统计量愈小愈好的原则来判别选模型(4)的优劣,即从 2^k-1 个子集中选一个子集使对应的 C_p 值达最小,并称为 C_p 准则下的最优子集,对应的模型(4)称为最优选模型。

C_p 准则是近年来颇受重视的一种变量选择准则[1],当前为变量选择所设计的计算方法都是串行算法,对于大型模型(如 $k\geqslant 20$)(1)式,要求最优子集,其计算量是惊人的甚至是无法实现的。关于用 C_p 准则作变量选择,就计算量而言,有二类方法求最优子集:其一是计算所有 2^k-1 个子集的 RSS_p 以获得最优子集[2,3];其二是按分支有界的思想,通过计算一部分 RSS_p 就能从全部 2^k-1 个子集中找到最优子集[2,4]. 文[3]指出在一般情况下前者与后者计算量相当。

本文基于第一类方法针对大型模型(1)式就 C_p 准则下首次提出一种选择最优子集的并行算法,这种算法能充分发挥并行机(即向量计算机或多处理机系统的计算机)的并行操作特性。

1 理论依据和串行算法描述

定义 设 $A=(a_{ij})_{n\times n}$,若 $a_{ii}\neq 0$,定义方阵 $B=(b_{ij})_{n\times n}$,称由 A 到 B 的这种变换为以 i 为枢轴号的 S 运算(扫描运算),记为 $B=S_iA$,其中

$$\begin{cases} b_{ii}=\dfrac{1}{a_{ii}}, b_{ij}=\dfrac{a_{ij}}{a_{ii}}, b_{ji}=-\dfrac{a_{ji}}{a_{ii}}, j\neq i \\ b_{kl}=a_{kl}-\dfrac{a_{ki}a_{il}}{a_{ii}}, k\neq i, l\neq i \end{cases} \tag{11}$$

若方阵 $B=(b_{ij})_{n\times n}$ 为

$$b_{kl}=\begin{cases} a_{kl}-\dfrac{a_{ki}a_{il}}{a_{ii}}, i+1\leqslant k, l\leqslant n \\ a_{kl}, \qquad\qquad \text{其它} \end{cases} \tag{12}$$

则称这种变换为以枢轴号 i 的 Gauss 消去运算(简称 G 运算),记为 $B=G_iA$.

$$A\triangleq\begin{pmatrix} X'X & X'Y \\ Y'X & Y'Y \end{pmatrix}_{(k+1)\times(k+1)} \tag{13}$$

定理 1 设 $1\leqslant i_1<i_2<\cdots<i_p\leqslant k$,对(13)式 A 施以 S 运算 S_{i_1}, S_{i_2}, …, S_{i_p},得到 $B=S_{i_p}\cdots S_{i_2}S_{i_1}A$,则 B 的结构为:

(1) b_{i_1k+1}, b_{i_2k+1}, …b_{i_pk+1} 为选入子集 x_{i_1}, x_{i_2}, …, x_{i_p} 的回归系数的 LSE,即(6)式。

(2) b_{k+1k+1} 为对应子集的残差平方和,即(8)式。

定理 2 设 $1\leqslant i_1<i_2<\cdots<i_p\leqslant k$,对(13)式 A 施以 G 运算 G_{i_1}, G_{i_2}, …, G_{i_p},得到 $B=G_{i_p}\cdots G_{i_2}G_{i_1}A$,则 B 的第 $(k+1,k+1)$ 个元素 b_{k+1k+1} 是选入子集 x_{i_1}, x_{i_2}, …, x_{i_p} 的残差平方和,即(8)式。

定理 1 和定理 2 的证明见文献[1].

对于大型模型(1)式,由(10)式知求 C_p 准则下最优子集的计算量在于生成 2^k-1 个子集下标集和计算 2^k-1 个 RSS_p 及 2^k-1 个 C_p 值的比较。第一部分工作采用字典式子

集计算次序[2]，第二部分由定理2知，用G运算产生2^k-1个RSS_p，第三部分由(9)式$\hat{\sigma}^2$及(10)式产生2^k-1个C_p值并求C_p最小值及存贮对应的子集下标集。注意，利用字典式子集计算次序求一个子集对应的RSS_p时，只需以该子集下标集最大下标为枢轴号作一次G运算，且由(13)式A的对称性知G运算只需对上(或下)三角块施行即得RSS_p，又注意到由l个元素$\{1,2,\cdots,l\}$组成不同子集(但含元素l)共有2^{l-1}个，由此即得如下定理。

定理3 对(13)式A施以G运算求2^k-1个RSS_p的计算量(乘、除法总次数)：

$$N \triangleq \sum_{l=1}^{k} 2^{l-1} \frac{(k+1-l)(k+4-l)}{2} \tag{14}$$

推论
$$N=6\cdot 2^k-\frac{(k+3)(k+4)}{2} \tag{15}$$

证明 由(14)式得

$$\begin{aligned} N &= (k+1)(k+4)\sum_{l=1}^{k}2^{l-2}-(2k+5)\sum_{l=1}^{k}l2^{l-2}+\sum_{l=1}^{k}l^2 2^{l-2} \\ &= (k+1)(k+4)\frac{2^k-1}{2}-(2k+5)\frac{(k-1)2^k+1}{2} \\ &\quad +\frac{1}{2}[2^k(k^2-2k+3)-3] \end{aligned}$$

化简上式即得(15)式。

由(15)式知求每个RSS_p的计算量平均不到6个乘除法操作，由此说明生成2^k-1个子集下标集和找2^k-1个C_p值的最小者的工作量是整个计算过程中不可忽略的。

2 并行算法

设向量计算机的向量长度$L=2^s$，s为正整数，令$k_1=k-s$，k为全模型(1)式的自变量个数。并行算法的基本思想是形成一组(L个)"基矩阵"，在同一时刻求L个子集及对L个"基矩阵"同时作G运算求对应的RSS_p. 定义如下数组：$\{a^{(h)}_{l,i,j}\}_{L\times(k+1)\times(k+1)}$，$h=1,2,\cdots,k_1+1$；$(b_l)_{L\times 1}$，$(c_l)_{L\times 1}$，$(d_l)_{L\times 1}$，$(e_l)_{L\times 1}$，$(f_{l,i})_{L\times s}$，$(g_{l,i})_{L\times k_1}$，$(V_i)_{(k_1+1)\times 1}$. 数组$\{a^{(h)}_{l,i,j}\}$存放(13)式及$G$运算的各种结果矩阵，$(f_{l,i})$、$(g_{l,i})$存放变量子集的下标集，$(c_l)$、$(e_l)$存放对应子集的$C_p$值，$(b_l)$、$(d_l)$存放对应子集所含变量的个数，$(V_l)$是字典式子集生成次序的工作数组。这里称$\{a^{(1)}_{l,i,j}\}_{(k+1)\times(k+1)}$，$l=1,2,\cdots,L$，为一组($L$个)"基矩阵"。并行算法步骤：

第一步，将矩阵(13)式存入数组$\{a^{(1)}_{1,i,j}\}_{(k+1)\times(k+1)}$，并对(13)式依次以$1,2,\cdots,k$为枢轴号作$S$运算，所得矩阵的最后一列分别为(5)和(7)式，由(9)式得$\hat{\sigma}^2$.

第二步，对s个自变量x_1，x_2，$\cdots$，x_s由字典式子集计算次序生成$L-1$个子集下标集，依次存入数组$f_{l,i}$，$i=1$，2，$\cdots$，s，$l=2$，3，$\cdots$，L. 对应的子集所含自变量的个数依次存入数组b_l，$l=2$，3，$\cdots L$，且以对应的子集下标集最大下标为枢轴号作G运算(对矩阵(13)式)，所得矩阵依次存入数组$\{a^{(1)}_{l,i,j}\}_{(k+1)\times(k+1)}$，$l=2$，$3$，$\cdots$，$L$. 注意这一步共作$L-1$次$G$运算，这一过程就是所谓形成一组"基矩阵"的过程。这时$a^{(1)}_{l,k+1,k+1}$，$l=2$，$\cdots$，L为对应子集的RSS_p，算出对应的C_p值存入数组c_l，$l=2$，3，$\cdots$，L. 取$c_1=10^{10}$(大正数)，取$b_1=0$，$f_{1,i}=0$，$i=1$，2，$\cdots$，s.

第三步，并行计算（此段“=”表示将右边值赋给左边）：

$$v_1=0,m=1,d_l=0,g_{l,i}=0,l:1\sim L,i:1\sim k_1$$

(1) $m=m+1$，$v_m=v_{m-1}+1$ (16)

(2) $I_1=v_{m-1}+1$，$I_2=v_m+1$，$I_3=v_m+s$

$$\begin{cases} a_{l,i,j}^{(I_2)}=a_{l,i,j}^{(I_1)}-\dfrac{a_{l,i,I_3}^{(I_1)}a_{l,j,I_3}^{(I_1)}}{a_{l,I_3,I_3}^{(I_1)}}, & \begin{matrix} l:1\sim L \\ i:I_3+1\sim k+1 \\ j:I_3+1\sim i \end{matrix} \\ e_l=\dfrac{a_{l,k+1,k+1}^{(I_2)}}{\hat{\sigma}^2}+2(b_l+m-1)-n, & l:1\sim L \end{cases} \tag{17}$$

若

$$\begin{cases} e_l\leqslant C_l,\text{则} \\ d_l=m-1,c_l=e_l \qquad l:1\sim L \\ g_{l,j-1}=v_j+s,j:2\sim m \end{cases} \tag{18}$$

若 $v_m<k_1$，则转向(1)；否则 $m=m-1,v_m=v_m+1$ (19)

若 $m>1$，则转向(2)；否则我们已从 2^k-1 个子集中得到一组(L 个)子集，其中包含了最优子集，此组子集的下标集在数组$(f_{l,i})$、$(g_{l,j})$中，对应的 C_p 值在数组(c_l)中。然后只要找这 L 个子集对应的最小 C_p 值，即为所求的最优子集。

第四步，设第三步所得最优子集下标集为 i_1，i_2，…，i_{p_0}，则对矩阵(13)式依次以 i_1，i_2，…，i_{p_0}为枢轴号作 S 运算，所得矩阵的最后一列的第 i_1，i_2，…，i_{p_0}行元素即为最优子集对应的回归系数 β_{i_1}，β_{i_2}，…，$\beta_{i_{p_0}}$的 LSE，即(6)式。算法结束。

3 模拟计算及注释

某工程工具误差系数分离模型是一个含 30 个自变量(不含常数项)的线性回归模型(1)式，且 $e\sim N(0,\sigma^2 I)$，其中 $n=900$. 今从已知 X 中取前 k_i 列派生出新的设计阵，记为 $X^{(k_i)}$，取 $\sigma^2=1$，并产生 $N(0,1)$随机数 $\hat{e}$，对应的回归系数 $\beta^{(k_i)}$用模拟真值代替，由 $X^{(k_i)}$，$\beta^{(k_i)}$及 $\hat{e}$ 产生 $Y=X^{(k_i)}\beta^{(k_i)}+\hat{e}$，这里 $k_i=20$，21，22，23，24、30. 于是我们得到一组形如(1)式的模型，其中 k_i 即所构造模型含自变量的个数。针对以上六个模型，分别用 C_p 准则选择最优子集，就本文的并行算法和传统的串行算法在 YH－1 计算机上作向量运算和标量运算，得向量执行时间（v－cpu）和标量执行时间（s－cpu）表 1；同时对 $k=30$ 的模型就本文算法在 YH－2 巨型机上做了单机和整机（四台处理机并行处理）运算，计算时间见表 2. 由于篇幅有限，有关最优子集的回归系数的 LSE 未列出。

表 1 YH－1 上的 cpu 时间（秒）（L=2^7）

k_i	20	21	22	23	24	30
$v-cpu$	4.5	9.5	17	32	66	4245
$s-cpu$	65	135	256	490	995	
s/v	14.4	14.2	15.1	15.3	15.1	

表 2 ($k=30$)YH－1 与 YH－2 运行时间(秒)

YH－1		YH－2	
L=2^7	L=2^6	单机	四机
4245	5523	2006	518

注 1 表 1 和表 2 中的 L 表示做向量运算时算法中采用的向量长度(YH－1 机的向

量长度 L=2^7，YH−2 机的 L=2^6)；表 1 中 s/v 为向量加速比(串行计算与并行计算的时间之比)；表 1 中 $k_i=30$ 的模型未做标量运算，但从表中可估其 s−cpu 将约用 60000 多秒。表 1 显示本算法在 YH−1 上运行其向量加速比可达 15 左右；表 2 给出了 $k=30$ 的模型用本算法在 YH−1 和 YH−2 上运行的效果，对 YH−1 采用两种向量长度的运算，对 YH−2($L=2^6$)给出了单机(用一台处理机)和整机(四台处理机同时作并行运算)的运行效果。表 2 指出本文算法用于 $k=30$ 的模型在 YH−2 上找最优子集仅需 518 秒就能完成。表 1 和表 2 充分显示了该并行算法的优越性，也体现该算法用于 C_p 准则的变量选择的可行性。

注 2　本文算法的特点：其一使(17)、(18)式完全并行化；其二使得(16)、(19)式从 2^k-1 次标量操作降到 $2^{k-s}-1$ 次标量操作，也即使得 2^k-1 个子集的下标集的生成降到只要生成 $2^{k-s}-1$ 个子集的下标集。这就是说本文算法能充分发挥向量机的向量操作特性。本文算法用于某工程的分离模型是 YH−2 考机课题之一，运行结果如表 2.

注 3　本文算法可移植到关于用其它准则[1](如 Akaike 信息量准则 $AIC_p \triangleq n \cdot ln(RSS_p)+2 \cdot p$ 和预测平方和准则——$PRESS_p$ 准则)的变量选择，其并行思想仍是形成一组(L 个)"基矩阵"，在同一时刻求 L 个子集及对 L 个"基矩阵"同时作 S 运算或 G 运算，这里 L 与前同。本文首次提出变量选择的并行算法，这种算法用于大型线性回归模型的变量筛选，对于巨型向量计算机来说，其实际意义是明显的。

参 考 文 献

1　陈希孺，王松桂. 近代回归分析，安徽教育出版社，1987：151～210

2　Furnival G M and Wilson R W M. Regression by Leaps and Bounds. Technometrics, 1974, 16: 499～511

3　Furnival G M. All Possible Regressions with Less Computation. Technometrics, 1971, 13: 403～408

4　Hocking R R and Leslie R N. Selection of the Best Subset in Regression Analysis. Technometrics, 1967, 9: 531～540

A Parallel Algorithm of Selecting the Best Subset on C_p Criterion

Hu Qingjun　Wu Yi

(Department of System Engineering and Applied Mathematics)

Abstract

The C_p criterion[1] is recently paid much attention to as an important one of variable selection. A parallel algorithm and the number of operations (multiplications and divisions) of selecting the best subset from all possible subsets, with regard to a large−scale linear regression model under C_p criterion, are presented. The simulation results for the algorithm on the vector super−computers YH−1 and YH−2 are given. The vector speed−up, ratio, s/v, approximates to fifteen and the advantages of the algorithm are shown.

Key words　linear regression model, variable selection criterion, subset regression, parallel computation, vector speed−up ratio