常用算法及应用程序

中国科学院成都计算机应用研究所
第 四 研 究 室

科学技术文献出版社重庆分社

常用算法及应用程序

中国科学院成都计算机应用研究所

科学技术文献出版社重庆分社

常用算法及应用程序

中国科学院成都计算机应用研究所第四研究室 编辑

科学技术文献出版社重庆分社 出版

重庆市市中区胜利路91号

四川省新华书店重庆发行所 发行

重庆市印制第一厂 印刷

开本：787×1092毫米1/32 印张：8.5 字数：19万

1984年3月第一版 1984年3月第一次印刷

科技新书目：87-227 印数：12177

书号：15176·570 定价：0.90元

前言

随着电子计算机的应用日益广泛，广大科技工作者、工程技术人员和管理干部等都希望学到一些常用算法和应用程序。为此，我们结合科研工作和生产实践，编写了《常用算法及应用程序》一书，供有关同志参阅和使用。

全书分三章。第一章是数理统计部份，主要是多元统计分析方法，共有六个标准过程。其中四个属于多元回归分析，二个属于聚类分析。多元统计分析方法是一种很有用的数据处理方法。在我国，多元统计分析方法已广泛应用于地质、气象、水文、医学和生物等许多方面，并取得了很好的效果。

第二章是最优化计算方法，共有十二个标准过程。其中一维搜索五个，无约束求极值五个，线性规划二个。随着国民经济的发展，最优化计算方法在科学管理，工程最优设计等方面得到了广泛的应用。

第三章是有限元法和其它应用程序，共有二个标准过程，三个程序和二个附录。其中有限元法二个，线性代数方程组的解法二个，三角高程平差的计算一个。有限元计算程序是借助计算机外存采用变带宽存贮分块求线性方程组之解，因而能在中小型计算机上求解较大型的弹性应力或应变问题。程序设计采用多次扫描方法实现了从劲度矩阵的分块到方程组分块求解过程的自动化。本章程序可应用于工程设计，野外测量等方面。

本书的标准过程和程序是用DJS-6机 ALGOL 语言编写

的，并采用统一的格式说明。在功能与方法的叙述上，力求准确，简明易懂。每个程序都在计算机上用过，大多数程序都经过实例检验，证明使用可靠。对于其它配有ALGOL语言的计算机，只要稍作改动，本书标准过程也可使用。

本书第一章由董汉忠、杨德明、文德生同志编写。第二章由姚诗源、徐明保、施要武、杨昌凤、杨新蓉同志编写。第三章由袁运祥同志、文兴贵等编写。四川大学数学系黎世烈副教授审阅了本书初稿，并提了宝贵的意见，在此表示衷心感谢。由于编著者水平有限，难免有缺点、错误，恳切希望读者批评、指正。

内 容 简 介

本书收集了一些具有实用价值的常用算法和应用程序。内容包括多元统计分析、最优化计算方法、有限元法和三角高程平差的计算等方面。每一算法均附有实例，以增强对算法的理解，掌握算法的技巧和领会算法的应用。

本书共有23个电子计算机计算程序，均用ALGOL语言编写，可供计算工作者、科技人员、工程技术人员及高等院校师生阅读、参考、使用。

目　录

第一章　数理统计

1. 逐步回归分析

（一）功　　能

本程序从与因变量y有关的诸可测因子x_1，x_2，…，x_m之中，根据各变量方差贡献的显著性检验结果，自动地建立起y与各显著变量x_i（$i=1，2，\cdots k$）之间的回归方程。这种方程可以是线性的或非线性的（通过使用者给出的函数F进行变换）。

本程序还可以根据使用者的要求，强迫选取某些变量进入回归方程。

本程序还可以分别逐步地建立各因变量y_1，y_2，…y_t的各别的回归方程，此时只要可测因子x_1，x_2，…x_m一致，取样一致。即分别建立

$$\hat{y}_1=f_1(x_1^{(1)}, x_2^{(1)}, \cdots, x_{k_1}^{(1)})$$

$$\hat{y}_2=f_2(x_1^{(2)}, x_2^{(2)}, \cdots, x_{k_2}^{(2)})$$

$$\cdots\cdots$$

$$\hat{y}_t=f_t(x_1^{(t)}, x_2^{(t)}, \cdots, x_{k_t}^{(t)})$$

其中$x_1^{(1)}$，$x_2^{(1)}\cdots x_{k_1}^{(1)}$，…，$x_1^{(t)}$，$x_2^{(t)}$，…，$x_{k_t}^{(t)}\in\{x_1, x_2, \cdots x_m\}$。在建立回归方程的同时，程序还同时给出复相关系

数，回归平方和，残差平方和，剩余标准差，各个取样的回归值及它与实测值之间的离差。

（二） 方 法 概 要

回归分析假定因变量 y 与 m 个自变量有称为回归方程的如下关系：

$$y=\beta_0+\sum_{i=1}^{m}\beta_i x_i+\varepsilon,$$

式中 β_0，β_1，…，β_m 称为回归系数，ε 是随机变量，亦称“剩余误差”。

回归分析的主要任务是根据 x_1，x_2，…，x_m，y 的 N 组观测数据得出正规方程组：

$$\begin{cases} l_{11}b_1+l_{12}b_2+\cdots+l_{1m}b_m=l_{1y} \\ l_{21}b_1+l_{22}b_2+\cdots+l_{2m}b_m=l_{2y} \\ \cdots\cdots \\ l_{m1}b_1+l_{m2}b_2+\cdots+l_{mm}b_m=l_{my}, \end{cases}$$

式中 l_{ij} 为变量 x_i 与 x_j（包括 y）的离差。然后解此方程组算出 β_0，β_1，…，β_m 的估值 b_0，b_1，…，b_m。

逐步回归不同时算出全部 b_1，b_2，…，b_m，它先在 m 个变量中选出重要性大的一个组成一元回归方程，然后按变量的贡献大小每步选一个重要变量进入回归方程，且每引入一个变量后就要考虑早先进入的变量中有无因新变量的进入而失去重要性的。若有就要剔除，且直到再无可剔除的变量时才考虑下一步引入变量问题。如此反复最终建立只包含重要

变量的回归方程。逐步回归的计算步骤如下：

1. 建立正规方程，形成离差矩阵。主要使用公式

$$l_{ij}=\sum_{k=1}^{N}(x_{ki}-\bar{x}_i)(x_{kj}-\bar{x}_j) \quad i,\ j=1,\ 2,\ \cdots,\ m$$

$$l_{im+1}=\sum_{k=1}^{N}(x_{ki}-\bar{x}_i)(y_{kd}-\bar{y}_d) \quad i=1,\ 2,\ \cdots,\ m,$$

d=因变量序号

$$l_{m+1,m+1}=\sum_{k=1}^{N}(y_{kd}-\bar{y}_d)。$$

2. 从建立一元线性回归方程开始逐步计算。当已引入了P个变量后，用同一公式$V_i^{(P)}=l_{iy}^{(P)}l_{yi}^{(P)}/l_{ii}^{(P)}$算出已引入每一变量当要剔除它时可能损失的方差贡献和未引入的每个变量一旦引入时所增加的贡献。

3. 在已引入变量中选出最小V值者，譬如是V_k，检查它的显著性，计算$F=(N-P-1)\cdot V_k^{(P)}/Q^{(P)}$。此$F$服从$F(1,N-P-1)$分布，可查表获得临界值$F_2$，也可根据问题的性质取定$F_2$，当$F<F_2$时表明在其它变量的影响下它已变得不显著，应把$x_k$从回归方程中剔除。

4. 再没有变量应该剔除时考虑是否有可以引入的变量，在未引入的变量中选取有最大V值者，仍记为x_k，计算$F=[N-(P+1)-1]\cdot V_k^{(P+1)}/Q^{(P+1)}=(N-P-2)\cdot V_k^{(P+1)}/(Q^{(P)}-V_k^{(P+1)})$，式中$Q^{(P)}$表示有$P$个变量时的残差平方和。对于入选标准$F_1(F_1\geqslant F_2)$，当$F>F_1$时表明此变量影响显著应引

入回归方程。

5. 对需要剔除或引入的x_k，用高斯一约当消去法作一次消去运算：

$$l_{ij}^{(p+1)}=\begin{cases} l_{kj}^{(p)}/l_{kk}^{(p)} & (i=k,\ j\neq k)\\ l_{ij}^{(p)}-l_{ik}^{(p)}\cdot l_{kj}^{(p)}/l_{kk}^{(p)} & (i\neq k,\ j\neq k)\\ 1/l_{kk}^{(p)} & (i=k,\ j=k)\\ -l_{ik}^{(p)}/l_{kk}^{(p)} & (i\neq k,\ j=k)。\end{cases}$$

在消去过程中可以得到一系列过渡性回归方程。

6. 在不能剔除也无法再引入的情况下建立最后的回归方程，计算有关数据。其中，复相关系数 $R=\sqrt{1-Q/L_{yy}}$，Q为残差平方和，L_{yy}为离差平方和；标准差 $S=\sqrt{Q/(N-m'-1)}$，m'—入选变量数；回归常数 $b_0=\bar{y}_d-\sum\limits_i b_i\bar{x}_i$；各样品值与回归值的残差 $e_k=y_{kd}-\hat{y}_{kd}=y_{kd}-(b_0+\sum\limits_i b_i x_{ki})$，以上求和号仅对已选变量$x_i$进行。

7. 对相关系数$r_{ij}=l_{ij}/\sqrt{l_{ii}}\cdot\sqrt{l_{jj}}$组成的矩阵按对称正定矩阵求逆$(c_{ij})$，再由$r'_{ij}=c_{ij}/\sqrt{c_{ii}}\cdot\sqrt{c_{jj}}$计算偏相关系数。

（三） 使 用 说 明

过程调用语句形式为 $MRA(M, YM, N, H, WAY, F, F1, F2, U, Y)$；

其中形参为：

M　整型量，表示自变量个数。

YM　整型量，表示随机变量个数。

N　整型量，表示样品数。

H　整型量，表示人工控制进入变量的个数，可以为0。

WAY　整型量，表示输出方式。为0时宽行输出最终结果；为1时宽行输出最终结果，窄行输出中间结果；为2时窄行输出中间结果和最终结果。其中，中间结果包括变量间的相关系数，逐步回归剔除或引入自变量的序号以及过渡性回归方程的系数；最终结果包括自变量和随机变量的均值，复相关系数R，标准差S，回归系数，随机变量的回归值和残差。偏相关系数由宽行输出。

F　过程，表示自变量的变换函数（由使用者编写）。

$F1,F2$　实型量，表示变量引入、剔除的标准（$F1 \geqslant F2$）。

U　数组，表示观测数据矩阵$\mathbf{U}=(u_{ij})_{n\times m}$。

Y　数组，表示随机变量数据矩阵$\mathbf{Y}=(y_{ij})_{n\times ym}$。

说明：本过程依次对每一列y值计算一个回归方程。算出一个回归方程后暂停，欲算下一列y再启动一次；欲跳过q列，再启动前应将控制台变量III拔为q。当$H\neq 0$时用数组$HA[0:H]$记强行进入回归方程的自变量序号，穿此数据带时应先多穿一个“0；”，例如要强行引入x_2，x_5则应穿“0；2；5；”。

(四) 过　　程

```
'PROCEDURE' MRA(M, YM, N, H, WAY, F,
F1, F2, U, Y);
'VALUE' M, YM, N, H, WAY, F1, F2;
'INTEGER' M, YM, N, H, WAY;
'REAL' F1, F2;
'ARRAY' U, Y;
'PROCEDURE' F;
'BEGIN'
'INTEGER' I, J, K, D, NMIN, NMAX, W, P;
'REAL' V, VMIN, VMAX, RN, SN, LYY;
'ARRAY' A[1:N, 1:M], XL[1:M, 1:M], AA1,
L[1:M+1, 1:M+1], AW, X[1:M+1], B[0:M];
'INTEGER''ARRAY' HA[0:H];
'SWITCH' SW:=S1, S2, S3, SY, SH;
F(U, A, M, N); P:=M+1;
'IF'WAY'GR'0'THEN' OUTAR(A);
X:=0;
'FOR'I:=1'STEP'1'UNTIL'M'DO'
'BEGIN'
'FOR'J:=1'STEP'1'UNTIL'N'DO'
X[I]:=X[I]+A[J, I];
X[I]:=X[I]/N
'END';
```

```
XL:=0;
'FOR'I:=1'STEP'1'UNTIL'M'DO'
'BEGIN'
'FOR' J:=I'STEP'1'UNTIL'M'DO'
'FOR'K:=1'STEP'1'UNTIL'N'DO'
XL[I, J]:=XL[I, J]+(A[K, I]-X[I])
*(A[K, J]-X[J]);
'FOR'J:=1'STEP'1'UNTIL'I'DO'
'BEGIN'
V:=XL[J, I]/SQRT(XL[I, I]*XL[J, J]);
AA1[I, J]:=AA1[J, I]:=V;
'IF' WAY'GR'0'THEN' OUTPUTR(V)
'END'
'END';
'IF' WAY 'GR' 0 'THEN'
'BEGIN' OUTAR(XL); DUMMY(5) 'END';
D:=0; HA:=0;
'IF' H'NQ' 0 'THEN' INPUTI(HA);
SY:D:=D+1;
'IF' WAY 'GR' 0 'THEN'
'BEGIN' OUTPUTI(D);
'FOR' I:=1'STEP' 1 'UNTIL' N 'DO'
OUTPUTR(Y[I, D])
'END';
L:=0; X[P]:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
```

```
'BEGIN'
X[P]:=X[P]+Y[I, D];
L[P, P]:=L[P, P]+Y[I, D]*Y[I, D]
'END';
L[P, P]:=L[P, P]-X[P]*X[P]/N;
X[P]:=X[P]/N;
'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO'
'BEGIN'
'FOR' J:=1 'STEP' 1 'UNTIL' M 'DO'
L[I, J]:=L[J, I]:=XL[I, J];
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
L[I, P]:=L[I, P] + (A[J, I]-X[I])*(Y[J, D]-
X[P]);
'IF' WAY'GR' 0 'THEN'
'BEGIN'
DUMMY(1); OUTPUTI(I); OUTPUTR(L[I, P])
'END'; L[P, I]:=L[I, P];
V:=L[I, P]/SQRT(L[I, I]*L[P, P]);
AA1[I, P]:=AA1[P, I]:=V;
'IF'WAY'GR' 0 'THEN'OUTPUTR(V)
'END';
'IF' WAY'GR' 0 'THEN' OUTPUTR(L[P, P]);
LYY:=L[P, P]; W:=N-1; DUMMY(3);
'PAGE'
S1: VMIN:=₁₀70; VMAX :=0;
NMIN:=NMAX:=0;
```

```
'FOR'I:=1'STEP' 1 'UNTIL' M 'DO'
'BEGIN'
B[I]:=0;
'IF'L[I, I]'LS' 10-20 'TNEN' 'GOTO' SH;
V:=L[I, P]*L[P, I]/L[I, I];
'IF' V 'LS' 0 'THEN'
'BEGIN'
B[I]:=L[I, P];
'FOR' J:=1'STEP' 1 'UNTIL' H 'DO'
'IF' I=HA[J]'THEN' 'GOTO' SH;
'IF' -V 'LS' VMIN 'THEN'
'BEGIN' VMIN:=-V; NMIN:=I'END'
'END' 'ELSE'
'IF' V 'GR' VMAX 'THEN'
'BEGIN' VMAX:=V; NMAX:=I 'END';
SH:
'END';
'IF' WAY 'GR' 0 'THEN'
'BEGIN' DUMMY(1); OUTAR(B) 'END';
'IF' VMIN*W/L[P, P]'GQ' F2 'THEN'
'BEGIN'
'IF' VMAX*(W-1)/(L[P, P]-VMAX) 'LS' F1
'THEN'
'BEGIN'
'FOR' I:=1 'STEP' 1 'UNTIL' H 'DO'
'BEGIN'
```