Aplikace matematiky

Pavel Bureš

Algoritmy. 15. ANAREG. Mnohonásobná lineární regrese

Aplikace matematiky, Vol. 13 (1968), No. 4, 361--365

Persistent URL: http://dml.cz/dmlcz/103180

ALGORITMY

15. ANAREG

MNOHONÁSOBNÁ LINEÁRNÍ REGRESE

PAVEL BUREŠ, prom. mat., Ústav výpočtové techniky ČSAV a ČVUT Horská 3, Praha 2

Používá se metody nejmenších čtverců, která vede na soustavu lineárních rovnic a ta se řeší upravenou Gauss-Jordanovou metodou.

```
procedure ANAREG (n, m, f, data); value n, m, f;
  integer n, m; real f; array data;
comment n je počet náhodných veličin, m je počet pozorování v každé veličině,
f je číslo z intervalu (0,3) a udává, zda veličina má být přijata do regresní závislosti,
data je identifikátor pole s rozměry n + 2 a m, které obsahuje n + 2 m-rozměrných
vektorů jednotlivých pozorování. Na prvních n místech jsou vektory nezávisle pro-
měnných, na n + 1 místě je vektor závisle proměnné veličiny a na n + 2 místě
je vektor koeficientů vah (obvykle jsou to jednotky);
begin
real array mat [1 : n + 1, 1 : n + 1], prum, odch, b [1 : n + 1];
real a, b0, sy, vmin, vmax, v, fi, c1, c2, c3, c4, c5, yn, roz, pom;
integer i, j, k, nmin, nmax;
boolean p;
a := 0;
for i := 1 step 1 until m do a := a + data [n + 2, i];
for i := 1 step 1 until n + 1 do
begin prum [i] := 0;
    for j := 1 step 1 until m do prum [i] := prum [i] +
        data [n + 2, j] × data [i, j];
    prum [i] := prum [i]/a
end i;
for i := 1 step 1 until n + 1 do
for j := 1 step 1 until i do
begin mat [i, j] := 0;
    for k := 1 step 1 until m do mat [i, j] := mat [i, j] + data [i, k]
        × data [j, k] × data [n + 2, k];
        mat [i, j] := mat [i, j] − prum [i] × prum [j] × a;
end j;
```

```
for i := 1 step 1 until n + 1 do odch [i] := sqrt (mat [i, i]);
comment lze tisknout součet vah a, průměry prum [i] a odchylky odch [i] jednotli-
vých proměnných;
for i := 1 step 1 until n + 1 do
    for j := 1 step 1 until i do mat [j, i] := mat [i, j] := mat [i, j]/odch[i]
    /odch [j];
comment lze tisknout korelační matici mat [i, j];
fi := a − 1;
l1 : for i := 1 step 1 until n do b [i] := 0;
   sy := odch [n + 1] × sqrt (mat [n + 1, n + 1]/fi); vmin = 10↑4;
     vmax := nmin := nmax := 0;
     for i := step 1 until n do
         begin
         if mat [i, i] < 10↑−6 then go to l2;
           v := mat [i, n + 1] × mat [n + 1, i]/mat [i, i];
         if v = 0 then go to l2; if v > 0 then go to l3;
           b[i] := mat [i, n + 1] × odch [n + 1]/odch [i];
         if abs (v) ≧ abs (vmin) then go to l2; vmin := v; nmin := i;
         go to l2;
l3:      if v ≦ vmax then go to l2; vmax := v; nmax := i; b[i] := 0;
         l2 : end i;
           b0 := 0; for i := 1 step 1 until n do b0 := b0 + b[i] × prum [i];
           b0 := prum [n + 1] − b0;
      if abs (vmin) × fi/mat [n + 1, n + 1] < f then
         begin k := nmin; fi := fi + 1; go to l4 end;
if vmax × fi/(mat [n + 1, n + 1] − vmax) < f then go to l5;
  k := nmax; fi := fi − 1;
l4 : for i := 1 step 1 until n + 1 do begin
    for j := 1 step 1 until n + 1 do begin
       p := true; if j = k then begin c1 := 0; c2 := 1; p := false end
       else begin c1 := mat [i, j]; c2 := mat [k, j] end;
    if i = k then begin if p then b [j] := c2/mat [k, k] else
    begin c4 := c2/mat [k, k]; c3 := mat [k, k] end end
    else if p then mat [i, j] := c1 − c2 × mat [i, k]/mat [k, k]
    else c3 := c1 − c2 × mat [i, k]/mat [k, k];
    end j; mat [i, k] := c3;
    end i;
    for i := 1 step 1 until n + 1 do mat [k, i] := b[i];
      mat [k, k] := c4; go to l1;
l5 : comment zde tiskneme rozptyl regresní proměnné sy o absolutní koeficient
     regrese b0;
     for i := 1 step 1 until n do
```

```
    begin if b [i] = 0 then go to l6;
        c1 = sy × sqrt (mat [i, i])/odch [i];
        comment zde tiskneme podle pořadí i regresní koeficienty b [i] a jejich roz-
        ptyly sb_i jež jsou postupně počítány v c1;
l6 : end i;
c1 := c2 := c3 := 0;
for i := 1 step 1 until m do begin
yn := 0;
for j := 1 step 1 until n do yn := yn + b [j] × data [j, i];
yn := b0 + yn; c1 := c1 + yn × data [n + 1, i] × data [n + 2, i];
roz := yn − data [n + 1, i];
comment zde lze tisknout spočtenou regresní proměnnou, jejíž hodnoty nabývá
postupně yn a rozdíly e_i mezi yn_i a veličinou y_i roz;
c2 := c2 + roz ↑2; if i = 1 then pom := roz;
c3 := c3 + (roz − pom) ↑2; pom := roz
end i;
c4 := odch [n + 1]↑2 + prum [n + 1]↑2 × a;
c5 := odch [n + 1]↑2;
c2 := c3/c2; c3 := (c4 − c1)/(m − n); c5 := 1 + (c1 − c4)/c5;
c4 := c1/c4;
comment lze tisknout Durban-Watsonovu statistiku d v c2, míru variance S_Y^2 v c3,
        míru korelace R1 a R2 tj. c4 a c5;
end proc anereg;
```

Algoritmus je určen hlavně pro nalezení lineárních závislostí mezi náhodnými veličinami. Při použití transformací ho lze též využít pro aproximaci hledané funkce, známe-li předem její tvar.

Přesnost výsledků závisí na počtu pozorování m a na přesnosti vnitřního zobrazení čísla v počítači.

Vektory označujeme velkými písmeny, jejich složky odpovídajícími malými písmeny s indexy.

Mějme m pozorování od každé náhodné veličiny $X_1, X_2, \ldots, X_n$, Y a dále m rozměrný vektor koeficientů vah ω (obvykle $\omega_j = 1$ pro $j = 1, 2, \ldots, m$). Tyto údaje jsou zadány v dvojrozměrném poli *data*. Hledá se lineární závislost mezi veličinami Y a $X_1, \ldots, X_n$ tak, aby výraz

$$\sum_{j=1}^{m} (y_j \omega_j - \sum_{i=1}^{n} b_i x_{ij} \omega_j)^2$$

byl minimální. Tato metoda nejmenších čtverců vede na řešení soustavy n lineárních rovnic, které se řeší upravenou Gauss-Jordanovou metodou.

Algoritmus počítá průměry μ_{X_i} resp. μ_Y a směrodatné odchylky σ_{X_i} resp. σ_Y náhodných veličin $X_1, \ldots, X_n$ resp. Y;

dále matici korelačních koeficientů mezi všemi náhodnými veličinami

$$\varrho_{X_i X_k} = \frac{\sqrt{\left(\frac{1}{m}\sum_{j=1}^{m}(x_{ij} - \mu_{X_i})(x_{kj} - \mu_{X_k})\right)}}{\sigma_{X_i}\sigma_{X_k}}$$

a

$$\varrho_{X_i Y} = \frac{\sqrt{\left(\frac{1}{m}\sum_{j=1}^{m}(x_{ij} - \mu_{X_i})(y_j - \mu_Y)\right)}}{\sigma_{X_i}\sigma_Y} .$$

Regresní proměnná je pak dána vztahem

$$Yn = b0 + \sum_{i=1}^{n} b_i X_i .$$

Algoritmus dále počítá její rozptyl sy,

koeficienty regrese b_i, jejich rozptyly sb_i a koeficient $b0$, dále rozdíly mezi veličinou Y a vypočtenou regresní proměnnou Yn, $e_j = yn_j - y_j$, kde $j = 1, 2, \ldots, m$.

Durban-Watsonovu statistiku

$$d = \sum_{j=2}^{m}(e_j - e_{j-1})^2 / \sum_{j=1}^{m} e_j^2$$

je možno použít k testování hypotezy o normálním rozložení veličiny Y. V případě, že $d \sim 2$ nelze tuto hypotezu zamítnout. Naposledy se počítá

míra variance

$$S_Y^2 = \frac{\sum_{j=1}^{m}(\omega_j y_j^2 - \omega_j yn_j y_i)}{m - n}$$

a míry korelace

$$R_1 = \frac{\sum_{j=1}^{m} yn_j y_j \omega_j}{\sum_{j=1}^{m} \omega_j y_j^2} ,$$

$$R_2 = 1 + \frac{\sum_{j=1}^{m} yn_j y_j \omega_j - \sum_{j=1}^{m} \omega_j y_j^2}{\sum_{j=1}^{m} \omega_j y_j^2 - \left(\sum_{j=1}^{m} \omega_j y_j\right)^2 / \sum_{j=1}^{m} \omega_j} .$$

Z technických důvodů byla v algoritmu použita jiná označení, než se ve statistice běžně užívají. Přehled těchto odchylek je následující:

$prum\,[i]$ ... μ_{X_i}	$mat\,[i, k]$ ... $\varrho_{X_i X_k}$
$prum\,[n+1]$ ... μ_Y	$mat\,[i, n+1]$ ... $\varrho_{X_i Y}$
$odch\,[i]$ ... σ_{X_i}	$b[i]$ ... b_i
$odch\,[i]$ ... σ_Y	pro $i = 1,2, \ldots n$; $k = 1,2, \ldots n$

Kontrolní příklad

Vstupní údaje:

$n = 1$;
$m = 11$;
$f = 0.5$;

prvky pole *data*:

$X = 2.063, 1.721, 1.403, 1.125, 0.898, 1.586, 1.376, 1.142, 1.268, 1.103, 1.019$;
$Y = 3.601, 3.192, 2.892, 2.581, 2.326, 3.117, 2.886, 2.616, 2.805, 2.627, 2.585$;
$\omega = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1$;

Vypočtené hodnoty:

$a = 11$; $prum\,[1] = 1.3367$; $prum\,[2] = 2.8389$; $odch\,[1] = 1.0857$;
$odch\,[2] = 1.1303$;
$mat\,[1,2] = 0.9937$;
$sy = 0.0423$;

Regresní rovnice:

$Yn = 1.456 + 1.0345X$; tj. $b0 = 1.456$; $b = 1.0345$;

rozptyl regresního koef. $sb_1 = 0.038965$;

$Yn = 3.5903, 3.2364, 2.9075, 2.6199, 2.385, 3.0968, 2.8795, 2.6375, 2.7678, 2.5971, 2.5102$;

Vektor rozdílů $(Yn - Y) = E$

$E = -0.0107, +0.0444, -0.0155, +0.0389, +0.059, -0.0202, -0.0065, +0.0215, -0.0372, -0.0299, -0.0748$;
$d = 1{,}0926$; $S_Y^2 = 0.0018$;
$R_1 = 0.9998$;
$R_2 = 0.9874$;

Algoritmus byl naprogramován ve strojním kódu počítače URAL 2 jako standardní program a ověřen v jazyku Algol.

[1] *Ralston - Wilf:* Numerical Methods of Digital Computers.
[2] *A. Hald:* Statistical Theory with Engineering Applications.
[3] *Lucy J. Slater:* Regression Analysis (The Computer Journal Vol. IV., No. 4).