STATISTICA MULTIVARIATA

Corso di laurea in matematica | Anno accademico 2004/05 | Numero 4

La matrice dei dati

Situazione 15.1. Sia $X \in \mathbb{R}^n_m$ con $n \geq 2$. Questa matrice sarà la nostra *matrice dei dati*. Scriviamo X nella forma

$$X = \begin{pmatrix} x_{11} & \dots & x_{1m} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{nm} \end{pmatrix}$$

La j-esima colonna di X è denotata con X_j. Abbiamo quindi

$$X_j = \begin{pmatrix} x_{1j} \\ \vdots \\ x_{nj} \end{pmatrix}$$

La i-esima riga di X è invece

$$x_i := (x_{i1}, \dots, x_{im})$$

Nota 15.2. Nel caso $m = 2$, per confrontare la nuova notazione con quella usata finora, scriveremo talvolta $x = X_1$, $y = X_2$. In questo caso la matrice dei dati è

$$X = (x, y)$$

Definizione 15.3. $1^n_n \in \mathbb{R}^n_n$ sia la matrice quadratica $n \times n$ i cui coefficienti sono tutti uguali ad 1:

$$1^n_n := \begin{pmatrix} 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{pmatrix}$$

Definiamo

$$M := \frac{1}{n} 1^n_n$$

Anche M è naturalmente una matrice quadratica $n \times n$.

Osservazione 15.4. *Per $v \in \mathbb{R}^n$ si ha*

$$Mv = \overline{v} 1^n$$

Dimostrazione. Infatti

$$\begin{aligned} Mv &= \frac{1}{n} \begin{pmatrix} 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \\ &= \frac{1}{n} \begin{pmatrix} v_1 + \dots + v_n \\ \vdots \\ v_1 + \dots + v_n \end{pmatrix} \\ &= \begin{pmatrix} \overline{v} \\ \vdots \\ \overline{v} \end{pmatrix} = \overline{v} 1^n \end{aligned}$$

Corollario 15.5. $M 1^n = 1^n$.

Dimostrazione. Per l'osservazione 15.4 e usando l'osservazione 6.6 abbiamo

$$M 1^n = \overline{1^n} 1^n = 1^n$$

Corollario 15.6. $MX = (\overline{X_1} 1^n, \dots, \overline{X_m} 1^n)$.

Dimostrazione. Per definizione

$$X = (X_1, \dots, X_m)$$

per cui

$$MX = (MX_1, \dots, MX_m)$$

L'enunciato segue dall'osservazione 15.4.

Nota 15.7. Nel caso $m = 2$ abbiamo perciò

$$MX = M(x, y) = (\overline{x} 1^n, \overline{y} 1^n)$$

Definizione 15.8. $\overline{X} := (\overline{X_1}, \dots, \overline{X_m})$ è il baricentro delle righe di X. Si noti che $\overline{X} \in \mathbb{R}_m$, mentre $MX \in \mathbb{R}^n_m$.

Definizione 15.9. La matrice

$$X^{CE} := X - MX$$

si chiama la matrice dei dati *centralizzata*. Se con δ denotiamo la matrice identica in $\mathbb{R}^n_n$, possiame anche scrivere

$$X^{CE} = (\delta - M) X$$

Per il corollario 15.6

$$\begin{aligned} X^{CE} &= (X_1 - \overline{X_1} 1^n, \dots, X_m - \overline{X_m} 1^n) \\ &= (X_1^{CE}, \dots, X_m^{CE}) \end{aligned}$$

Nel caso $m = 2$ abbiamo

$$(x, y)^{CE} = (x^{CE}, y^{CE})$$

Nella letteratura la matrice $X - MX$ viene talvolta anche chiamata la *matrice delle deviazioni* (dalle medie).

Osservazione 15.10.

$$1^n_n 1^n_n = n 1^n_n = \begin{pmatrix} n & \dots & n \\ \vdots & & \vdots \\ n & \dots & n \end{pmatrix}$$

Dimostrazione. Chiaro da

$$\begin{pmatrix} 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{pmatrix} = \begin{pmatrix} n & \dots & n \\ \vdots & & \vdots \\ n & \dots & n \end{pmatrix}$$

Corollario 15.11. $M^2 = M$.

Dimostrazione. Per l'osservazione 15.10 abbiamo

$$\begin{aligned} M^2 &= \frac{1}{n} 1^n_n \cdot \frac{1}{n} 1^n \\ &= \frac{1}{n^2} 1^n_n 1^n_n = \frac{n}{n^2} 1^n_n \\ &= \frac{1}{n} 1^n_n = M \end{aligned}$$

Corollario 15.12. $(\delta - M)^2 = \delta - M$.

Dimostrazione. Infatti dal corollario 15.11 abbiamo

$$\begin{aligned} (\delta - M)^2 &= \delta - 2M + M^2 \\ &= \delta - 2M + M = \delta - M \end{aligned}$$

In questo numero

apply

Introduciamo qui una funzione di R fondamentale per la trasformazione di righe o colonne di una matrice. Siano `f` una funzione definita per vettori (a valori non necessariamente numerici) che per ogni argomento restituisca un vettore della stessa lunghezza ≥ 1 ed `A` una matrice (a valori non necessariamente numerici). Allora

```
t(apply(A,1,f,***))
```

è la matrice che si ottiene da `A` eseguendo `f` su ogni riga di `A`, ed

```
apply(A,2,f,***)
```

è la matrice che si ottiene eseguendo `f` su ogni colonna di `A`. In entrambi i casi `***` indica eventuali ulteriori argomenti di `f`.

Esempio:

```
A = matrix (c(1:4,2:9,3:7,16:9,1:3),
  ncol=4)
print(A)

# 1 5  5 12
# 2 6  6 11
# 3 7  7 10
# 4 8 16  9
# 2 9 15  1
# 3 3 14  2
# 4 4 13  3

B = apply(A,2,sort)
print(B)

# 1 3  5  1
# 2 4  6  2
# 2 5  7  3
# 3 6 13  9
# 3 7 14 10
# 4 8 15 11
# 4 9 16 12
```

Definiamo le seguenti due funzioni: `Smg.M(X)` calcola MX, `Smg.cen(X)` corrisponde a X^{CE}. Si noti comunque che la seconda è più semplice e, a causa della vettorialità delle operazioni in R, non richiede la prima.

```
Smg.M = function (X)
{n=nrow(X)
apply(X,2,
  function (x) rep(mean(x),n))}

Smg.cen = function (X)
apply(X,2,function (x) x-mean(x))
```

Proiezione su $[0, 1]$

In statistica conviene spesso trasformare i valori contenuti in un vettore `v` di dati numerici in valori compresi tra `0` e `1`. Ciò può essere ottenuto con l'operazione

$$\bigcirc_x \frac{x - m}{M - m}$$

applicata agli elementi di `v`, dove m è il minimo in `v`, M il massimo. Denotiamo il vettore così ottenuto con v^{01}.

In R programmiamo (come abbiamo fatto a pagina 7 del corso di Fondamenti)

```
S.tra01 = function (v)
{m=min(v); (v-m)/(max(v)-m)}
```

Questa funzione fa parte della sezione S (statistica) della nostra libreria. Esempio:

```
x=1:5
print(S.tra01(x))
# 0.00 0.25 0.50 0.75 1.00
```

Per applicare `S.tra01` a tutte le colonne di una matrice, ottenendo così la matrice

$$X^{01} := (X_1^{01}, \ldots, X_m^{01})$$

combiniamo questa funzione con `apply`:

```
Sm.tra01 = function (X)
apply(X,2,S.tra01)
```

Per la matrice X dei 15 comuni su questa stessa pagina con `Sm.tra01(X)` otteniamo allora, dopo arrotondamento,

$$\begin{pmatrix}
0.00 & 1.00 & 0.54 & 0.16 \\
0.27 & 0.14 & 0.50 & 0.14 \\
0.05 & 0.68 & 1.00 & 0.00 \\
0.08 & 0.02 & 0.32 & 0.58 \\
0.27 & 0.13 & 0.54 & 0.08 \\
0.47 & 0.05 & 0.01 & 0.30 \\
1.00 & 0.32 & 0.77 & 0.21 \\
0.14 & 0.03 & 0.18 & 0.07 \\
0.11 & 0.14 & 0.64 & 0.34 \\
0.04 & 0.01 & 0.07 & 0.22 \\
0.08 & 0.01 & 0.06 & 1.00 \\
0.68 & 0.62 & 0.75 & 0.13 \\
0.06 & 0.51 & 0.79 & 0.17 \\
0.19 & 0.00 & 0.00 & 0.67 \\
0.06 & 0.10 & 0.39 & 0.05
\end{pmatrix}$$

Questa tecnica è utile molto spesso tranne nei casi in cui, per la presenza di uno o più valori eccezionali in una colonna, la colonna trasformata diventa troppo concentrata su una piccola porzione dell'intervallo $[0, 1]$:

```
x=c(2,3,5,6,7,11,13,100)
x1=S.tra01(x)
print(round(x1,2))
# 0 0.01 0.03 0.04 0.05 0.09 0.11 1
```

Proprio nella statistica esploratoria lo studio di X^{01} è in genere da preferire all'uso delle normalizzazione statistica

$$X^{NS} := (X_1^{NS}, \ldots, X_m^{NS})$$

che appartiene piuttosto alla statistica parametrico-inferenziale.

Quindici comuni

Lavoreremo negli esempi spesso con la seguente tabella di quindici comuni italiani, di cui abbiamo quattro dati: numero degli abitanti, altezza sul mare, distanza dal mare, superficie del territorio comunale. Per avere numeri di grandezza confrontabili, indichiamo gli abitanti in migliaia, l'altezza in metri, la distanza dal mare in chilometri, la superficie in chilometri quadrati.

comune	ab.	alt.	mare	sup.
Belluno	35	383	75	148
Bologna	380	54	70	141
Bolzano	97	262	140	53
Ferrara	132	9	45	405
Firenze	375	50	75	103
Genova	632	19	2	236
Milano	1302	122	108	182
Padova	210	12	25	93
Parma	170	55	90	261
Pisa	92	4	10	188
Ravenna	140	4	8	660
Torino	901	239	105	131
Trento	106	194	110	158
Venezia	275	1	0	458
Vicenza	110	39	55	81

I nomi naturalmente non fanno parte della matrice dei dati che in questo esempio è uguale a

$$X = \begin{pmatrix}
35 & 383 & 75 & 148 \\
380 & 54 & 70 & 141 \\
97 & 262 & 140 & 53 \\
132 & 9 & 45 & 405 \\
375 & 50 & 75 & 103 \\
632 & 19 & 2 & 236 \\
1302 & 122 & 108 & 182 \\
210 & 12 & 25 & 93 \\
170 & 55 & 90 & 261 \\
92 & 4 & 10 & 188 \\
140 & 4 & 8 & 660 \\
901 & 239 & 105 & 131 \\
106 & 194 & 110 & 158 \\
275 & 1 & 0 & 458 \\
110 & 39 & 55 & 81
\end{pmatrix}$$

Se come prima denotiamo le colonne con X_1, X_2, X_3, X_4, abbiamo (arrotondando)

$$\overline{X_1} = 330.5$$
$$\overline{X_2} = 96.5$$
$$\overline{X_3} = 61.2$$
$$\overline{X_4} = 219.9$$

Possiamo così calcolare

$$X^{CE} = M - MX$$

$$= \begin{pmatrix}
-295.5 & 286.5 & 13.8 & -71.9 \\
49.5 & -42.5 & 8.8 & -78.9 \\
-233.5 & 165.5 & 78.8 & -166.9 \\
-198.5 & -87.5 & -16.2 & 185.1 \\
44.5 & -46.5 & 13.8 & -116.9 \\
301.5 & -77.5 & -59.2 & 16.1 \\
971.5 & 25.5 & 46.8 & -37.9 \\
-120.5 & -84.5 & -36.2 & -126.9 \\
-160.5 & -41.5 & 28.8 & 41.1 \\
-238.5 & -92.5 & -51.2 & -31.9 \\
-190.5 & -92.5 & -53.2 & 440.1 \\
570.5 & 142.5 & 43.8 & -88.9 \\
-224.5 & 97.5 & 48.8 & -61.9 \\
-55.5 & -95.5 & -61.2 & 238.1 \\
-220.5 & -57.5 & -6.2 & -138.9
\end{pmatrix}$$

Il comune di Ferrara ha un territorio molto grande, corrispondente a un quadrato di 20 km di lato, praticamente uguale a quello di Vienna (415 km^2), di poco inferiore a quello di Venezia e più del doppio di quello di Milano. Oltre a Ravenna e Venezia abbiamo trovato solo questi comuni italiani più grandi di Ferrara (superficie in km^2): Roma 1508, Foggia 506, Grosseto 475, L'Aquila 467, Perugia 450, Altamura 428, Caltanissetta 416, Viterbo 406.

Ranghi

Molto utile in una prima fase dell'analisi è anche l'informazione sui *ranghi* dei valori nelle colonne della matrice X. Ciò in R avviene tramite la funzione `rank` di R che calcola per ogni elemento di un vettore il suo rango, cioè la posizione di quell'elemento nel vettore ordinato (che si ottiene con `sort`).

Esempio:

```
x=c(3,5,1,10,9,2,8,6)
v=sort(x); print(v)
# 1 2 3 5 6 8 9 10

u=rank(x)
print(u)
#  3 4 1 8 7 2 6 5

x=c(2,5,6,2,1,3,5)
u=rank(x)
print(u)
# 2.5 5.5 7 2.5 1 4 5.5
```

Come si vede nel secondo esempio, nell'impostazione iniziale, quando il vettore contiene valori uguali, `rank` assegna a questi elementi la media dei ranghi. Ciò nella visualizzazione grafica crea il problema che questi elementi (almeno in una dimensione) non sono più distinguibili. Con l'impostazione `ties.method='first'` in `rank` i ranghi diventano di nuovo unici, assegnando un rango minore a quelli tra due o più elementi uguali che nel vettore appaiono per primi:

```
x=c(2,5,6,2,1,3,5)
u=rank(x,ties.method='first')
print(u)
# 2 5 7 3 1 4 6
```

Creiamo quindi due funzioni, di cui la seconda va applicata colonna per colonna a matrici di dati numerici, che calcolano i ranghi riportati a una scala che può essere impostata a seconda delle esigenze:

```
S.rango = function (x, scala=1)
{u=rank(x,ties.method='first')
(u-1)*scala/(length(x)-1)}

Sm.rango = function (X,scala=1)
apply(X,2,S.rango,scala=scala)
```

Nell'impostazione iniziale (`scala=1`) i ranghi vengono riportati all'intervallo $[0, 1]$, quindi per un vettore di 5 elementi otteniamo i punti 0, 0.25, 0.5, 0.75, 1:

```
x=c(2,8,1,3,2)
u=S.rango(x)
print(u)
# 0.25 1 0 0.75 0.5
```

Visualizzazione di ranghi

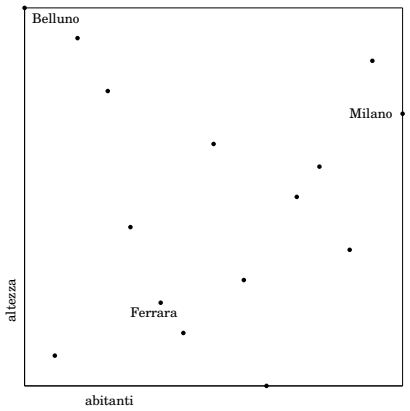
Se, per una figura su uno spazio di 50 mm, desideriamo in `S.rango` o `Sm.rango` una scala di 50, la possiamo reimpostare:

```
x=c(7,2,3,5,8,20,1,8,9,17,2)
u=S.rango(x,scala=50)
print(u)
# 25 5 15 20 30 50 0 35 40 45 10
```

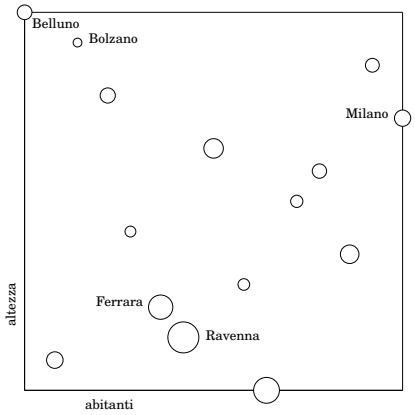
Queste funzioni si prestano particolarmente per la visualizzazione di matrici a due colonne. Applichiamo la funzione `Sm.rango` con `scala=50` alla matrice Y che consiste delle prime due colonne (relative al numero degli abitanti ed all'altezza sul mare) della matrice X che contiene i dati sui nostri 15 comuni. Otteniamo allora la matrice dei ranghi trasformati

$$\begin{pmatrix} 0 & 50 \\ 39 & 29 \\ 7 & 46 \\ 18 & 11 \\ 36 & 25 \\ 43 & 18 \\ 50 & 36 \\ 29 & 14 \\ 25 & 32 \\ 4 & 4 \\ 21 & 7 \\ 46 & 43 \\ 11 & 39 \\ 32 & 0 \\ 14 & 21 \end{pmatrix}$$

Riportiamo questi valori graficamente:



Se rappresentiamo ogni comune come cerchietto la cui area è proporzionale alla superficie, otteniamo



Esercizio:

(1) Aggiungere i nomi mancanti.
(2) Contrassegnare con una tilde i comuni distanti meno di 50 km dal mare. Non a sorpresa essi si trovano nella parte bassa della figura.
(3) Usare R per calcolare il coefficiente di correlazione tra altezza e distanza dal mare, cioè tra la seconda e la terza colonna della matrice X.

Lo studio dei ranghi è molto utile per una prima valutazione qualitativa delle relazioni tra le variabili; è però difficile tradurre questa visione qualitativa in criteri numerici che possano essere applicati successivamente ad altri insiemi di dati.

Correlazione di rango

La correlazione tra i vettori di rango si chiama la *correlazione di rango* di Spearman. Benché talvolta intuitiva e convincente, è difficile da interpretare numericamente.

Per il calcolo della correlazione di rango in genere si usano ranghi medi per elementi uguali la cui presenza crea qualche problema più teorico che pratico. In R la correlazione di rango tra due vettori x ed y la si ottiene quindi con

```
cor(rank(x),rank(y))
```

Usando `apply` possiamo anche calcolare la matrice delle correlazioni di rango per la matrice dei dati, ad esempio per i 15 comuni. Con

```
R=cor(apply(X,2,rank))
print(R)
```

otteniamo così

```
 1     -0.06  -0.06   0.08
-0.06   1      0.88  -0.54
-0.06   0.88   1     -0.51
 0.08  -0.54  -0.51   1
```

La matrice è simmetrica perché lo è il coefficiente di correlazione ed è chiaro che la diagonale principale è occupata da 1. Vediamo tra l'altro che la correlazione (dei ranghi) tra il numero degli abitanti e l'altezza sul mare è quasi zero (e negativa), ma anche quella tra numero degli abitanti e superficie, mentre la correlazione tra altezza e distanza dal mare è piuttosto alta (0.88).

Trasformazione affine dei dati

Nota 17.1. Vogliamo adesso studiare il comportamento della matrice dei dati quando sottoponiamo le variabili a una trasformazione affine. Consideriamo prima una matrice con 2 colonne e 3 righe:

$$X = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{pmatrix}$$

Se sostituiamo ogni riga (x_{i1}, x_{i2}) con

$$(x'_{i1}, x'_{i2}) = (4x_{i1}+3x_{i2}+9,\ 6x_{i1}+5x_{i2}+1)$$

otteniamo una matrice

$$X' = \begin{pmatrix} 4x_{11}+3x_{12}+9 & 6x_{11}+5x_{12}+1 \\ 4x_{21}+3x_{22}+9 & 6x_{21}+5x_{22}+1 \\ 4x_{31}+3x_{32}+9 & 6x_{31}+5x_{32}+1 \end{pmatrix}$$

Per capire come questa operazione possa essere scritta in forma matriciale, introduciamo la matrice

$$A = \begin{pmatrix} 4 & 6 \\ 3 & 5 \end{pmatrix}$$

Allora

$$(x'_{i1}, x'_{i2}) = (x_{i1}\ \ x_{i2}) \begin{pmatrix} 4 & 6 \\ 3 & 5 \end{pmatrix} + (9\ \ 1)$$

e, come adesso si vede facilmente,

$$X' = XA + \begin{pmatrix} 9 & 1 \\ 9 & 1 \\ 9 & 1 \end{pmatrix}$$

Infine

$$\begin{pmatrix} 9 & 1 \\ 9 & 1 \\ 9 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} (9\ \ 1) = 1^3 (9\ \ 1)$$

e quindi, con $b := (9\ \ 1) \in \mathbb{R}_2$,

$$X' = XA + 1^3 b$$

Che la matrice di trasformazione A appare alla destra di X è naturalmente dovuto al fatto che essa opera sui vettori riga della matrice dei dati.

È chiaro che questo ragionamento rimane valido in tutte le dimensioni. Una *trasformazione affine* dei dati è quindi un'operazione della forma

$$X \mapsto X' = XA + 1^n b$$

con $A \in \mathbb{R}^m_m$ e $b \in \mathbb{R}_m$.

A causa delle operazioni vettoriali in R possiamo ottenere X' semplicemente con

```
X%*%A+b
```

Proposizione 17.2. *Come nella nota 17.1 siano $A \in \mathbb{R}^m_m$, $b \in \mathbb{R}_m$ e*

$$X' = XA + 1^n b$$

Allora

$$MX' = MXA + 1^n b$$
$$X'^{CE} = X^{CE} A$$

Dimostrazione. La prima equazione segue dalla relazione

$$MX' = M(XA + M1^n b) = MXA + M1^n b$$

perché dal corollario 15.5 sappiamo che $M1^n = 1^n$.

Per la seconda equazione abbiamo, usando la prima,

$$\begin{aligned} X'^{CE} &= X' - MX' \\ &= XA + 1^n b - (MXA - 1^n b) \\ &= XA - MXA \\ &= (X - MX)A = X^{CE} A \end{aligned}$$

Bibliografia

3784 **A. Rizzi:** Analisi dei dati. NIS 1985.