計量 MDS

慶應義塾大学大学院 理工学研究科 開放環境科学専攻
修士 1 年 竹内研究室　横 山　暁*

0　序

最近，統計解析の手法の一つである多次元尺度構成法（Multi-Dimensional Scaling　以下 MDS と略す）に興味を持っている．

MDS とは，対象 i と j の親近性 s_{ij} がデータとして与えられたときに，ユークリッド空間にサンプルを布置し，類似したものを近くに，そうでないものを遠くに配置する方法であり，距離をデータとする**計量 MDS**，順序尺度で測定された親近性データを解析する**非計量 MDS** がある．

今回は，計算過程が割りと簡単である計量 MDS のプログラムを J によって実現した．

1　計量 MDS（Torgerson の方法）

地点 i の座標を $(x_{i1}, x_{i2}, \cdots, x_{iP})$，地点 j の座標を $(x_{j1}, x_{j2}, \cdots, x_{jP})$ とすると，2 点間のユークリッド距離 d_{ij} は，

$$d_{ij} = \sqrt{\sum_{m=1}^{P} (x_{im} - x_{jm})^2}$$

である．地点 k を原点にとり，任意の 2 点 i, j がつくる原点からの内積 z_{ij} を，

$$z_{ij} = \sum_{m=1}^{P} x_{im} x_{jm} \tag{1}$$

と定義すると，三角形の余弦定理より，

$$d_{ij}^2 = d_{ik}^2 + d_{jk}^2 - 2 d_{ik} d_{jk} \cos\theta$$

となり，内積 z_{ij} は，

$$z_{ij} = d_{ik} d_{jk} \cos\theta$$

となるので，

$$z_{ij} = \frac{1}{2} \left(d_{ik}^2 + d_{jk}^2 - d_{ij}^2 \right) \tag{2}$$

となる．

各地点の距離データから z_{ij} を求め，それを要素とする $(n-1) \times (n-1)$ 行列（元データのうち，k を原点としているため $(n-1) \times (n-1)$ となる）を内積行列 Z，そして求めるべき座標を並べた $(n-1) \times P$ の行列を X とすると，

$$Z = X X^t$$

と表される．Z から X を求める方法は，因子分析における相関行列から因子負荷行列を求める因子分解に相当し，Z の固有値，固有ベクトルを求め，固有値を対角要素とする行列を Λ，対応する固有ベクトルを列に並べたものを Y とすると，

$$Z = Y \Lambda Y^t$$

という分解（エッカート－ヤング分解）を行う．これより X は

$$X = Y \Lambda^{1/2} \tag{3}$$

によって与えられる．

この解法は距離に誤差を含まない場合を前提としていたため，どの点を原点としても問題なかったが，実際のデータでは誤差を含まないことは稀なため，どの点を原点とするかによって結果が異なってしまう．そこで原点に重心 $(\bar{x}_{\cdot 1}, \bar{x}_{\cdot 2}, \cdots, \bar{x}_{\cdot P})$ をとって解を求めると良い．原点を重心とした場合 (2) 式は，

$$z_{ij} = \frac{1}{2} \left(\sum_{i=1}^{n} \frac{d_{ij}^2}{n} + \sum_{j=1}^{n} \frac{d_{ij}^2}{n} - \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{d_{ij}^2}{n^2} - d_{ij}^2 \right) \tag{4}$$

* satoru_y@ae.keio.ac.jp

となる（このときの Z は $n \times n$ 行列になる）．この z_{ij} を要素とする Z をエッカート－ヤング分解して，X を求めれば良い．

2 プログラム

2.1 基本的な使い方

J のプログラムは最後に添付しておくが，プログラムは大きく分けて下のような 3 つの部分からなっている．

1. 座標データから距離 d_{ij} を計算する部分
2. d_{ij} から内積行列 Z を計算する部分
3. $X = Y\Lambda^{1/2}$ を求める部分

である．実際のプログラム (torgerson) では, それぞれ

1. distance
2. makemat
3. mds

に対応している．

実行方法は，座標データのファイルを用意しておき（例として，d:¥data.txt とする）

```
torgerson 'd:\data.txt'
```

とすればよい．

また，内積行列が用意されている場合には，

```
mds 3 3$48 _38 _36 _38 11 _1 _36 _1 7
```

とすればよい．

2.2 distance について

この部分では，data に入力されたマトリックスデータから 1 行ずつ取り出し，ユークリッド距離を

```
dist=:%:@(+/@:*:)
```

で繰り返し計算させ，matdata0 という距離行列 d_{ij} を作成している．

2.3 makemat について

この部分では，(4) 式の計算を行っている．具体的には，前段階で求めた matdata0 の 2 乗をとり，その列方向の和を個数で割ったもの $\sum_{i=j}^{n} \frac{d_{ij}^2}{n}$ を各行に並べたもの (aa) とそれを転置させたもの，$\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{d_{ij}^2}{n^2}$ を計算してすべての個数で割ったもの (cc) を各行各列に並べたものを求め，

```
matdata=:(aa+(|:aa)-(cc+matdata2))%2
```

により，(4) 式の計算を行っている．

2.4 mds について

メインとなる mds の部分では，前段階で求めた matdata（つまり $Z = \{z_{ij}\}$）の固有値，固有ベクトルを eigen によって求め，(3) 式の計算を行っている．ここで，出力される X は，Z の次元数だけ出力されるようになっている．

3 実例

簡単な例を実行した結果を示す．

このような 2 次元データを用意し，

$$(1,-3),(-3,4),(4,1),(-2,-2)$$

プログラムを実行した．その結果，

```
  _2.82843 _1.41421  8.26812e_7 _1.1493e_7
   4.94975 0.707107  5.34472e_7 1.74466e_7
  _2.12132  3.53553  1.44685e_7 5.60327e_7
6.08276e_12 _2.82843 _9.89312e_8  8.0149e_7
```

と出力され，

```
plot (0{"1 kekka);(1{"1 kekka)
```

としたところ図 1 のようになった．

また，データをそのままプロットすると図 2 のようになった．これらを比較すると，実行結果のプロットは，元のデータのプロットから，傾けた形にほぼなっていて，元のデータを 2 次元空間に上手く布置できていることがわかる．

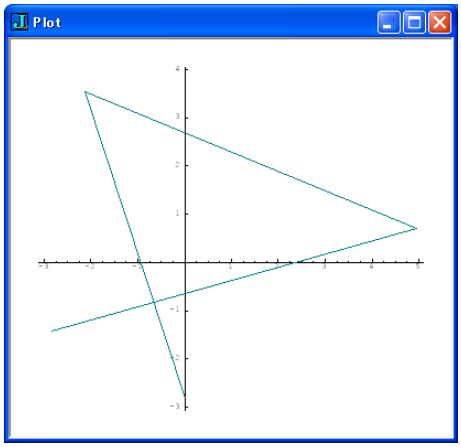

図 1　実行結果のプロット

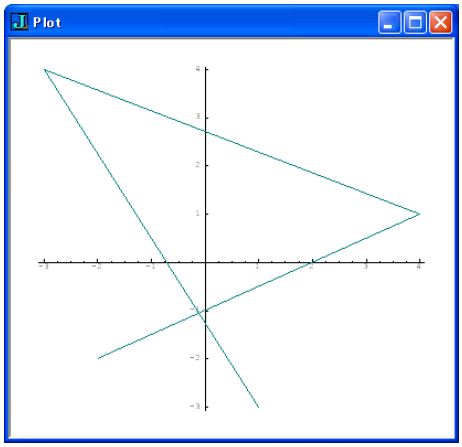

図 2　元のデータのプロット

参考文献

[1] 永田靖・棟近正彦共著：多変量解析入門，サイエンス社，164〜173，2001.
[2] 高根芳雄著：多次元尺度法，朝倉書店，1980.
[3] 高根芳雄著：多次元尺度法，東京大学出版会，1980.
[4] R.N. シェパード他編，岡太・渡邊訳，多次元尺度構成法　—理論編—，共立出版，1976.
[5] 林知己夫・飽戸弘著，多次元尺度解析法，サイエンス社，1976.
[6] 柳井晴夫・高根芳雄著：新版 多変量解析法，朝倉書店，1985.

付録 A　スクリプト

```
NB. Multi-Dimensional Scaling by Torgerson
NB. This is Designed for J5.01a
NB. Modified by Satoru Yokoyama 2003/10/28
NB. Last Modifid by 2003/12/04

load 'c:\j501a\temp\profile.ijs'
load 'c:\j501a\temp\jacobi.ijs'

  torgerson=: 3 : 0
data=:".'m' fread y.
distance''
makemat''
mds matdata
)

  mds=: 3 : 0
matdata=:y.
la=:eigen matdata
t=:%:(*/$matdata) NB.%:#d
lambda=:%: t{.{.la
lavec=:t{."1 }.la
kekka=:lavec * "1 lambda
)

   distance=: 3 : 0
dist=:%:@(+/@:*:)
d=:i.0
i=:0
n=:#data
for. i.n do.
    temp1=:i{data
    temp2=:data
    d=:d,dist"1(temp1-"1 temp2)
    i=:i+1
end.
matdata0=:((%:#d),%:#d)$d
)

   makemat=: 3 : 0
matdata2=:matdata0^2
aaa=:((1,#matdata2)$+/matdata2)%#matdata2
bbb=:(((#matdata2),1)$+/matdata2)%#matdata2
ccc=:+/+/"1 matdata2
aa=:i.0
for. i.#matdata2 do.
aa=:aa,aaa
end.
aa=:}.aa
cc=:(((#matdata2),#matdata2)$ccc)%(#matdata2)^2
matdata=:(aa+(|:aa)-(cc+matdata2))%2
)
```