smacof Data Structures

Jan de Leeuw

Metric MDS

The input for a metric (ratio) smacof consists of four vectors ik, jk, sk, and wk of the same length.

Elements ik[k], jk[k], dk[k], and wk[k] give the row index, the column index, the dissimilarity value, and the weight value for observation k. There can be replications and data can be asymmetric.

We assume

- the vector dk of dissimilarities is in non-decreasing order,
- dk is non-negative,
- the vector wk of weights is strictly positive,
- for all $k = 1, \cdots, m$ index ik[k] is not equal to index jk[k].

If the dissimilarity data come in a square matrix

```
     [,1] [,2] [,3] [,4]
[1,]    0    2    2    3
[2,]    1    0   NA    5
[3,]    2    4    0    8
[4,]    3    6    8    0
```

and all weights for non-missing data are one, then then a corresponding smacof data structure is

```
      ik jk dk wk
 [1,]  2  1  1  1
 [2,]  1  2  2  1
 [3,]  1  3  2  1
 [4,]  3  1  2  1
```

```
 [5,]   1   4   3   1
 [6,]   4   1   3   1
 [7,]   3   2   4   1
 [8,]   2   4   5   1
 [9,]   4   2   5   1
[10,]   3   4   8   1
[11,]   4   3   8   1
```

Because there are ties in the dissimilarities the data structure corresponding to the matrix is not unique. From the point of view of minimizing stress an equivalent data structure is

```
     ik  jk  dk  wk
[1,] 1.0 2.0 1.5 2.0
[2,] 1.0 3.0 2.0 2.0
[3,] 1.0 4.0 3.0 2.0
[4,] 2.0 3.0 4.0 1.0
[5,] 2.0 4.0 5.0 2.0
[6,] 3.0 4.0 8.0 2.0
```

$$\sigma(X)=\frac{1}{2}\sum_{k=1}^m w_k(\delta_k-d_k(X))^2$$

$$d_k(X)=\sqrt{\text{tr }X'A_kX}$$

$$A_k:=(e_{i_k}-e_{j_k})(e_{i_k}-e_{j_k})'$$

The A_k are intended to be used in formulas, not in actual computation.

$$V=\sum_{k=1}^m w_kA_k$$

```
smacofMakeVmat <- function(dat) {
  ndat <- nrow(dat)
  nobj <- max(max(dat[, 1], max(dat[, 2])))
  vmat <- matrix(0, nobj, nobj)
  for (k in 1:ndat) {
    i <- dat[k, 1]
    j <- dat[k, 2]
    w <- dat[k, 4]
    vmat[i, j] <- vmat[i, j] - w
    vmat[j, i] <- vmat[i, j]
```

```r
  }
  diag(vmat) <- -rowSums(vmat)
  return(vmat)
}
```

```r
print(smacofMakeVmat(thedata1))
```

```
     [,1] [,2] [,3] [,4]
[1,]    6   -2   -2   -2
[2,]   -2    5   -1   -2
[3,]   -2   -1    5   -2
[4,]   -2   -2   -2    6
```

```r
print(smacofMakeVmat(thedata2))
```

```
     [,1] [,2] [,3] [,4]
[1,]    6   -2   -2   -2
[2,]   -2    5   -1   -2
[3,]   -2   -1    5   -2
[4,]   -2   -2   -2    6
```

```r
smacofDistance <- function(dat, x) {
  ndat <- nrow(dat)
  dmat <- rep(0, ndat)
  for (k in 1:ndat) {
    i <- dat[k, 1]
    j <- dat[k, 2]
    dmat[k] <- sqrt(sum((x[i, ] - x[j, ]) ^ 2))
  }
  return(dmat)
}
```

```r
dmat1 <- smacofDistance(thedata1, x)
dmat2 <- smacofDistance(thedata2, x)
print(dmat2)
```

```
[1] 1.000000 1.000000 1.414214 1.414214 1.000000 1.000000
```

```r
smacofMakeBmat <- function(dat, dmat) {
  ndat <- nrow(dat)
  nobj <- max(max(dat[, 1]), max(dat[, 2]))
  bmat <- matrix(0, nobj, nobj)
  for (k in 1:ndat) {
    i <- dat[k, 1]
    j <- dat[k, 2]
    w <- dat[k, 4]
    e <- dat[k, 3]
    d <- dmat[k]
    bmat[i, j] <- bmat[i, j] - w * (e / d)
    bmat[j, i] <- bmat[i, j]
  }
  diag(bmat) <- -rowSums(bmat)
  return(bmat)
}
```

```r
print(smacofMakeBmat(thedata2, dmat2))
```

```
          [,1]       [,2]       [,3]       [,4]
[1,] 11.242641  -3.000000  -4.000000  -4.242641
[2,] -3.000000  15.828427  -2.828427 -10.000000
[3,] -4.000000  -2.828427  22.828427 -16.000000
[4,] -4.242641 -10.000000 -16.000000  30.242641
```

```r
print(smacofMakeBmat(thedata1, dmat1))
```

```
          [,1]       [,2]       [,3]       [,4]
[1,] 11.242641  -3.000000  -4.000000  -4.242641
[2,] -3.000000  15.828427  -2.828427 -10.000000
[3,] -4.000000  -2.828427  22.828427 -16.000000
[4,] -4.242641 -10.000000 -16.000000  30.242641
```

```r
smacofStress <- function(dat, dmat) {
    return(sum(dat[, 4] * (dat[, 3] - dmat) ^ 2))
}
```

```r
print(smacofStress(thedata2, dmat2))
```

```
[1] 144.2157
```

```r
print(smacofStress(thedata1, dmat1))
```

```
[1] 144.7157
```

$$B(X)=\sum_{k=1}^m\{w_k\frac{\delta_k}{d_k(X)}A_k\mid d_k(X)>0\}$$

Semimetric

this has a numeric delta, and we add a column with tie blocks

this is for splines etc and for the shepard plot

Nonmetric

delta are rank numbers with ties getting the same rank number

Pairs and Triads

Nominal

Individual Differences