Les méthodes d'appariement avec TDA et SAS

Laurent Lesnard
Centre de données socio-politiques et
Observatoire sociologique du changement (CNRS - Sciences Po)
Laboratoire de sociologie quantitative du CREST (Insee)

Organisation du fichier de données

- sous forme séquentielle avec une ligne = une séquence
et autant de colonnes (variables) que d'épisodes
- chaque état est codé de 1 à n (n = nombre d'états)

TDA

Appariement optimal

SAS

Classification ascendante hiérarchique

I re étape : TDA

- Importe les données au format Spss Portable ou Tda (v6)
- Deux utilisations possibles
  - interactif : `tda i`
  - batch : `tda cf=name_of_a_command_file`

Chaîne de traitement typique

Lecture des données

```
rspss()=d:\data\seqtrav.por;
rstata()=d:\data\seqtrav.v7.dta;
```

Déclaration des séquences

```
seqdef()=trav1,,trav144;
```

Optimal Matching

```
seqm(scost=2)=d:\data\distlist.dat;
```

seqm

- `sm =`
  - 1: saute les valeurs manquantes (et décale les séquences)
  - 2 : chaque suite d'états identiques devient un état unique (ex : 1111222 3 -> 123)
  - 1,2 : cumule les deux options
- `icost =`
  - α : coût indel unique
  - M : matrice qui contient la série de coûts indel
  - α,β : fonction de coûts indel linéaire $g(k) = \alpha + \beta (k-1)$
- `scost =`
  - 1: coûts de substitution entre a_i et b_j = $| a_i - b_j |$
  - 2 : inversement proportionnels aux fréquences de transition
  - M : matrice qui contient les coûts de substitution
- `df =` fichier.txt : donne des informations sur les coûts

Déclaration d'une matrice

- Déclaration d'une matrice

```
mdef(X,m,n) = x11, . . . , x1n, . . . , xm1, . . . , xmn;
```

- Exemple

```
mdef(I,3,3) = 1,0,0,
              0,1,0,
              0,0,1;
```

```
seqdef = Y1,,Y8;

mdef(SCOST,10,10) =
  0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9,
  0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,
  0.2, 0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,
  0.3, 0.2, 0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6,
  0.4, 0.3, 0.2, 0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5,
  0.5, 0.4, 0.3, 0.2, 0.1, 0.0, 0.1, 0.2, 0.3, 0.4,
  0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0, 0.1, 0.2, 0.3,
  0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0, 0.1, 0.2,
  0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0, 0.1,
  0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0;

seqm(
    icost = 1,
    scost = SCOST,
    df = seqm3.tst,
    tst = 2,3,
    dtda = seqm3.tda,
) = seqm3.d;
```

```
Output file: seqm1.d
-----------------------------
 2       1       5       5  2.00
 3       1       5       5  2.00
 3       2       5       5  2.00
 4       1       5       5  4.00
 4       2       5       5  2.00
 4       3       5       5  2.00

TDA description file: seqm1.tda
-------------------------------
nvar(
  dfile = seqm1.d,
  noc = 6,
  SEQ1N <5>[6.0] = c1  , # sequence A case number
  SEQ2N <5>[6.0] = c2  , # sequence B case number
  SEQ1L <2>[6.0] = c3  , # sequence A length
  SEQ2L <2>[6.0] = c4  , # sequence B length
  DIST  <4>[5.2] = c5  , # distance
);
```

2e étape : SAS

- Importer la liste de distances
- Transformer cette liste en matrice de distance
- Réaliser une classification ascendante hiérarchique