Starting TraMineR

Matthias Studer

Alexis Gabadinho, Gilbert Ritschard

LIVES NCCR, Université de Genève
Institut de Démographie et Socioéconomie, Université de Genève
http://mephisto.unige.ch/traminer

Déjeuner R, 6 décembre 2016

mvad variables

1	id	unique individual identifier
2	weight	sample weights
3	male	binary dummy for gender, 1=male
4	catholic	binary dummy for community, 1=Catholic
5	Belfast	binary dummies for location of school, one of five Education and Library Board areas in Northern Ireland
6	N.Eastern	"
7	Southern	"
8	S.Eastern	"
9	Western	"
10	Grammar	binary dummy indicating type of secondary education, 1=grammar school
11	funemp	binary dummy indicating father's employment status at time of survey, 1=father unemployed
12	gcse5eq	binary dummy indicating qualifications gained by the end of compulsory education, 1=5+ GCSEs at grades A-C, or equivalent
13	fmpr	binary dummy indicating SOC code of father's current or most recent job,1=SOC1 (professional, managerial or related)
14	livboth	binary dummy indicating living arrangements at time of first sweep of survey (June 1995), 1=living with both parents
15	jul93	Monthly Activity Variables are coded 1-6, 1=school, 2=FE, 3=employment, 4=training, 5=joblessness, 6=HE
	⋮	"
86	jun99	"

The 'mvad' data set

- McVicar and Anyadike-Danes (2002)'s study of school to work transition in Northern Ireland.
- dataset distributed with the TraMineR library.
- 712 cases (survey data).
- 72 monthly activity statuses (July 1993-June 1999)
- States are:
  - EM Employment
  - FE Further education
  - HE Higher education
  - JL Joblessness
  - SC School
  - TR Training.
- 14 additional (binary) variables
- The follow-up starts when respondents finished compulsory school (16 years old).

The mvad sequences are in STS form

- The mvad sequences are organized in STS form, i.e., each sequence is given as a (row) vector of consecutive states.

```
head(mvad[, 17:22])
```

```
##        Sep.93     Oct.93     Nov.93     Dec.93   Jan.94   Feb.94
## 1  employment employment employment employment training training
## 2          FE         FE         FE         FE       FE       FE
## 3    training   training   training   training training training
## 4    training   training   training   training training training
## 5          FE         FE         FE         FE       FE       FE
## 6 joblessness   training   training   training training training
```

Creating the state sequence object

- Most TraMineR functions for state sequences require a state sequence object as input argument.
- The state sequence object contains
  - the sequences
  - and their attributes (alphabet, labels, colors, weights, ...)
- Hence, we first have to create this object

Creating the state sequence object

- Loading the data set and creating the 'state sequence' object
  (from Sept 93 to June 99; i.e., positions 17 to 86: We skip July-August 93)

```
data(mvad)
mvad.lab <- c("employment", "further education", "higher education",
    "joblessness", "school", "training")
mvad.shortlab <- c("EM", "FE", "HE", "JL", "SC", "TR")
mvad.seq <- seqdef(mvad[, 17:86], states = mvad.shortlab,
    labels = mvad.lab, weights = mvad$weight, xtstep = 6)
```


State sequence object

- The state sequence object is at the core of TraMineR's features for state sequences.
- Alongside the set of sequences, it includes among others
  - the alphabet
  - short state labels for printed output (`states`)
  - long state labels for color legends in graphics (`labels`)
  - position labels (`cnames`)
  - color palette for representing the states (`cpal`)
  - weights (`weights`)
  - specifications about missing values (`left, gaps, right`)
  - ...
- To ensure homogeneity of produced output and figures, all these characteristics are defined once with `seqdef()` ... and retrieved when necessary by other TraMineR's functions.

Main sequence object attributes and seqdef arguments

Attribute name	Description	Argument	Default	Retrieve/Set
	input format	`informat=`	`"STS"`	
alphabet	list of states	`states=`	from input data	`alphabet()`
cpal	color palette	`cpal=`	from RColorBrewer	`cpal()`
labels	long state labels	`labels=`	from input data	`stlab()`
cnames	position names	`cnames=`	from input data	`names()`
xtstep	jumps between tick marks	`xtstep=`	`1`	
row.names	row (sequence) labels	`id=`	from input data	`rownames()`
weights	optional case weights	`weights=`	`NULL`	
	missing handling	`left=`	`NA`	
	"	`gaps=`	`NA`	
	"	`right=`	`"DEL"`	

Warning: Check the order of the states!

- Short and long labels should conform with state order!
- By default the order of the states is the alphabetical order.
- This is installation dependent and may vary across Windows, Linux and Mac OS.
- You can check the states and their order with

```
seqstatl(mvad[, 17:86])
```

```
## [1] "employment"  "FE"          "HE"          "joblessness" "school"
## [6] "training"
```

- and enforce the order with the `alphabet=` argument

```
mvad.alph <- c("employment", "FE", "HE", "joblessness",
    "school", "training")
mvad.seq <- seqdef(mvad[, 17:86], states = mvad.shortlab,
    labels = mvad.lab, weights = mvad$weight, xtstep = 6,
    alphabet = mvad.alph)
```


References I

McVicar, D. and M. Anyadike-Danes (2002). Predicting successful and unsuccessful transitions from school to work using sequence methods. *Journal of the Royal Statistical Society A 165*(2), 317–334.