



Multiway
networks

V. Batagelj

Introduction to the analysis of multiway networks

Vladimir Batagelj
IMFM Ljubljana and IAM UP Koper

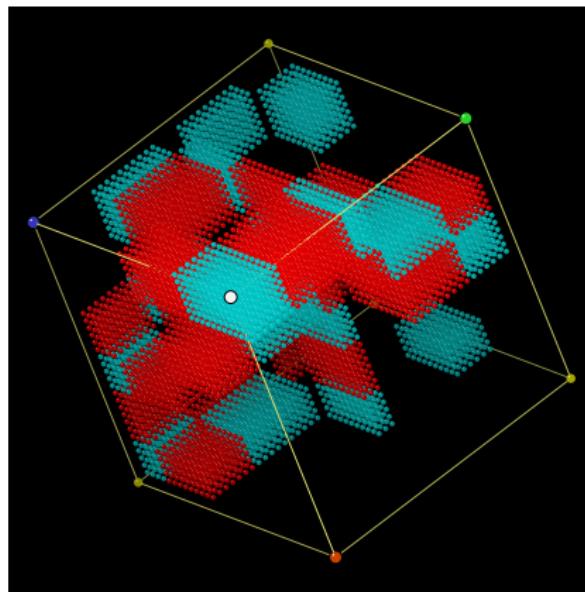
Lecture at University of Trieste, Zoom, 30 May 2023, based on
HiTEc Spring Course, 3-5 April 2023, Limassol, Cyprus



Outline

Multiway
networks

V. Batagelj



Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Current version of slides (May 30, 2023 at 10:08): [slides PDF](#)

<https://github.com/bavla/ibm3m/>

Multiway
networks

V. Batagelj



My renewed interest in the analysis of multiway networks was motivated by my Italian colleagues from Napoli and Salerno. At the IFCS 2022 conference in July in Porto, they were presenting their research on Italian students' mobility. They developed their own approach and asked me for comments and alternatives.



Italian students' data

Multiway
networks

V. Batagelj

Their data consists of quadruples of *ways* (province, university, study program, year) with the corresponding value (weight) counting how many students from a given province selected in a given year to study a given program at a given university. Such data are called *weighted multiway networks*.

Raw data

	e	prov	univ	prog	year	w
1	AG	Bicocca	BAL	2008	4	
2	AG	Bicocca	Edu	2008	1	
3	AG	Bicocca	NsMS	2008	1	
4	AG	Bicocca	SsJI	2008	1	
5	AG	Bocconi	BAL	2008	11	
6	AG	Foscari	AH	2008	1	
⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮
37200	VV	uTorino	AFFV	2017	3	
37201	VV	uTorino	AH	2017	11	
37202	VV	uTorino	BAL	2017	3	
37203	VV	uTorino	Edu	2017	1	
37204	VV	uTorino	ICT	2017	1	
37205	VV	uTorino	SsJI	2017	3	

Factors

	e	prov	univ	prog	year	w
1		1	1	1	4	1
2		2	1	1	5	1
3		3	1	1	9	1
4		4	1	1	11	1
5		5	1	2	4	1
6		6	1	3	3	1
⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮	⋮⋮
37200		107	72	2	4	3
37201		107	72	3	4	11
37202		107	72	4	4	3
37203		107	72	5	4	1
37204		107	72	8	4	1
37205		107	72	11	4	3



Italian students' data

Multiway
networks

V. Batagelj

The approach proposed by my Italian colleagues was to analyze the data for each year separately by joining the *ways* university and study program into a single way [university, study program] thus obtaining a weighted two-mode network that can be analyzed using available methods.

I first transformed in R their data into Pajek format and applied some additional methods to them.

Afterward, I created an R representation of multiway networks and saved the data in JSON format.

I extended the obtained data with some additional data about **Provinces of Italy** and long/short names.



Multiway networks

A *weighted multiway network* $\mathcal{N} = (\mathcal{V}, \mathcal{L}, w)$ is based on *nodes* from k finite sets (ways or dimensions) $\mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_k)$, the set of *links* \mathcal{L} , and the *weight* $w : \mathcal{L} \rightarrow \mathbb{R}$. The incidence function

$I : \mathcal{L} \rightarrow \mathcal{V}_1 \times \mathcal{V}_2 \times \dots \times \mathcal{V}_k$ assigns to each link $e \in \mathcal{L}$ a k -tuple of its nodes $I(e) = (e(1), e(2), \dots, e(i), \dots, e(k))$, $e(i) \in \mathcal{V}_i$. If for $i \neq j$, $\mathcal{V}_i = \mathcal{V}_j$, we say that \mathcal{V}_i and \mathcal{V}_j are of the same *mode*.

In a general multiway network, different additional data can be known for nodes and/or links $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$, where \mathcal{P} is a set of node properties $p : \mathcal{V}_i \rightarrow S_p$, and \mathcal{W} is a set of link weights $w : \mathcal{L} \rightarrow S_w$.

For a subset of links $\mathcal{L}' \subseteq \mathcal{L}$ we denote

$$\mathcal{V}_i(\mathcal{L}') = \{e(i) : e \in \mathcal{L}'\}$$

the set of nodes from the way \mathcal{V}_i that are an end-node of a link from \mathcal{L}' .



Simple networks

A multiway network \mathcal{N} is *simple* if and only if all its link tuples are mutually different – for all $e, f \in \mathcal{L}$ it holds $e \neq f \Rightarrow I(e) \neq I(f)$.

A multiway network $\mathcal{N}' = ((\mathcal{V}'_1, \mathcal{V}'_2, \dots, \mathcal{V}'_k), \mathcal{L}', w')$ is a *subnetwork* of the multiway network $\mathcal{N} = ((\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_k), \mathcal{L}, w)$ if and only if $\mathcal{V}'_i \subseteq \mathcal{V}_i$ for all $i \in 1..k$, $\mathcal{L}' \subseteq \mathcal{L}$, $I(\mathcal{L}') \subseteq \mathcal{V}'_1 \times \mathcal{V}'_2 \times \dots \times \mathcal{V}'_k$, and for all $e \in \mathcal{L}'$ holds $w'(e) = w(e)$.

The notion of a subnetwork could be extended to the omission of some ways. We could also introduce the notion of homomorphism/isomorphism of multiway networks.

For a node $u \in \mathcal{V}_i$ we call a *star* in the node u , the set of links

$$S(u) = \{e \in \mathcal{L} : e(i) = u\}.$$



Multiway network representation in R

Multiway
networks

V. Batagelj

I decided to use for multiway networks the following representation in R

```
MW <- list(  
  format="MWNets",  
  info= metadata,  
  ways= list of ways' names,  
  nodes= list of ways' dataframes,  
  links= dataframe of links and weights,  
  data= list of additional data  
)
```



Italian students mobility

Multiway
networks

V. Batagelj

```
> str(MN)
List of 6
$ format: chr "MWNets"
$ info :List of 5
..$ network: chr "students"
..$ title : chr "Student mobility data in Italian universities 2008, 2011, 2014, 2017"
..$ by    : chr [1:4] "VG Genova" "G Giordano" "G Ragozini" "MP Vitale"
..$ creator: chr "V. Batagelj"
..$ date   : chr "Sun Nov 6 22:31:11 2022"
$ ways :List of 4
..$ prov: chr "province"
..$ univ: chr "university"
..$ prog: chr "programme"
..$ year: chr "year"
$ nodes :List of 4
..$ prov:'data.frame': 107 obs. of 9 variables:
... ..$ ID      : chr [1:107] "AG" "AL" "AN" "AO" ...
... ..$ type    : chr [1:107] "F" "O" "D" ...
... ..$ province: chr [1:107] "Agrigento" "Alessandria" "Ancona" "Aosta" ...
... ..$ capital : chr [1:107] "Agrigento" "Alessandria" "Ancona" "Aosta" ...
... ..$ region  : chr [1:107] "Sicily" "Piedmont" "Marche" "Aosta Valley" ...
... ..$ IDreg   : chr [1:107] "SIC" "PIE" "MAR" "VAL" ...
... ..$ macreg  : chr [1:107] "Insular" "North-West" "Centre" "North-West" ...
... ..$ population: int [1:107] 416181 409392 464419 124089 203425 290811 336501 209390 402929
... ..$ area    : num [1:107] 3053 3559 1963 3261 1228 ...
..$ univ:'data.frame': 79 obs. of 2 variables:
... ..$ ID      : chr [1:79] "Bicocca" "Bocconi" "Foscarini" "Biomedico" ...
... ..$ long    : chr [1:79] "Bicocca" "Bocconi" "Cà Foscari" "Campus Biomedico" ...
..$ prog:'data.frame': 11 obs. of 2 variables:
... ..$ ID      : chr [1:11] "Q" "AFFV" "AH" "BAL" ...
... ..$ long    : chr [1:11] "?" "Agriculture, forestry, fisheries and veterinary" "Arts and humani...
..$ year:'data.frame': 4 obs. of 1 variable:
... ..$ ID      : chr [1:4] "2008" "2011" "2014" "2017"
```



Italian students mobility

Multiway
networks

V. Batagelj

```
$ links :'data.frame': 37205 obs. of 5 variables:  
..$ prov: int [1:37205] 1 1 1 1 1 1 1 1 1 ...  
..$ univ: int [1:37205] 1 1 1 1 2 3 4 5 5 5 ...  
..$ prog: int [1:37205] 4 5 9 11 4 3 9 3 5 9 ...  
..$ year: int [1:37205] 1 1 1 1 1 1 1 1 1 1 ...  
..$ w : int [1:37205] 4 1 1 1 11 1 1 1 1 1 ...  
$ data :List of 1  
..$ regs:'data.frame': 20 obs. of 2 variables:  
... ..$ ID : chr [1:20] "ABR" "BAS" "CAL" "CAM" ...  
... ..$ long: chr [1:20] "Abruzzo" "Basilicata" "Calabria" "Campania" ...
```

- the names of ways in ways should be the same as in nodes and links
- each data frame in nodes should contain the ID column
- the data frame links should contain at least one weight column



Multi-relational networks

A special kind of multiway networks are multi-relational (multiplex) networks. **Manlio De Domenico Datasets**

In a *multi-relational* network

$$\mathcal{N} = (\mathcal{V}, (\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_k), \mathcal{P}, \mathcal{W})$$

the set of links \mathcal{L} is partitioned into different relations (sets of links) $(\mathcal{L}_i)_{i \in I}$ over the same set of nodes. Also the weights from \mathcal{W} are defined on different relations or their union. A default weight is $w(e) = 1$ for all $e \in \mathcal{L}$.

The corresponding multiway network $\mathcal{M} = (\mathcal{V}_M, \mathcal{L}_M, \mathcal{W}_M)$ is constructed as follows:

- $\mathcal{V}_M = (\mathcal{V}, \mathcal{V}, (\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_k))$, \mathcal{L}_i are labels of relations
- the link $(u, v) \in \mathcal{L}_i$ with a weight $w_i(u, v)$ produces a link $e = (u, v, i) \in \mathcal{V}_M$ and $w_M(e) = w_i(u, v)$



Sources of multiway networks

Multiway
networks

V. Batagelj

- Manlio De Domenico: [Datasets Released for Reproducibility](#)
- [Alephsys NETWORK DATASETS](#) HIGGS TWITTER, CS-AARHUS, VICKERS CHAN 7TH GRADERS
- Multiplex networks [StarWars](#)
- [zalmquist/networkdata](#) Krackhardt
- [sciMAG2015](#) data access???
- [ICON - The Colorado Index of Complex Networks](#)
- [UCI](#): Cars, Mushrooms
- [PEW](#): GitHub, Data lab, Tools and resources, Data, Belief, Global religion
- European social survey

Multiway analysis

Multiway
networks

V. Batagelj

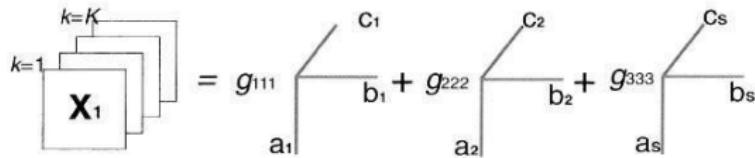
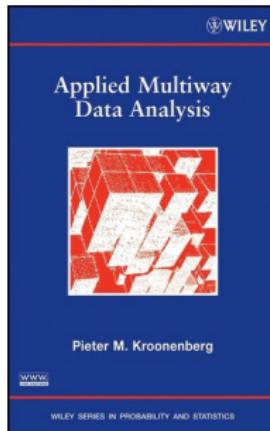


Figure 4.9 The Parafac model.

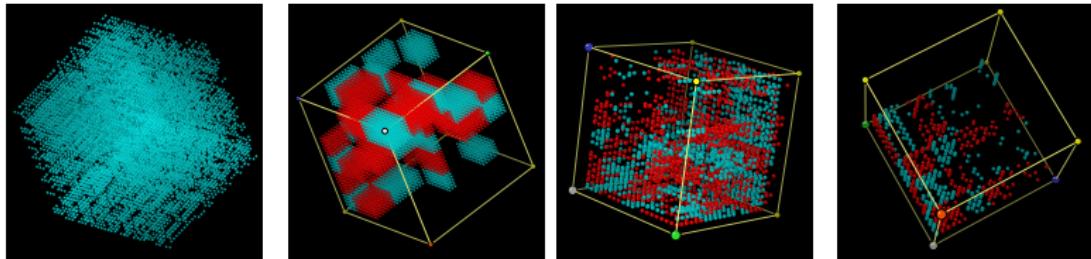
In data analysis already exists an approach called *multiway analysis* which deals with generalizations of ideas from the multivariate analysis to multiway arrays. They are known as Tucker, Parafac, SimPCA, ... models [?, ?].

We will continue in another, "combinatorial" direction.

Blockmodeling

Multiway
networks

V. Batagelj



I already worked (with Anuška and Pat, in 2006) on the indirect approach to blockmodeling of binary 3-way networks ([Github bavla/ibm3m](#)) [?]. [Lazega, Krackhardt](#) [?]

We were dealing with 3-way networks also at the INSNA 2009 Viszards session analyzing the [Bibsonomy](#) data.

Drilling into the data of my Italian friends I got some ideas for possible approaches.



MWnets

Multiway
networks

V. Batagelj

To support the analysis of multiway networks I started to develop an R package MWnets. It is available at GitHub/Bavla

<https://github.com/bavla/ibm3m>



```
> source("https://raw.githubusercontent.com/bavla/ibm3m/master/  
multiway/MWnets.R")
```

Multiway network data sets

<https://github.com/bavla/ibm3m/tree/master/data>

For user-friendly inspection of a JSON file, you can use the Firefox browser.



Some functions

Multiway
networks

V. Batagelj

`reorderways(MN,ord)` – Reorder ways of the multiway network MN in the order determined in the vector ord.

`slice(MN,P)` – Extract from the multiway network MN a subnetwork of links satisfying a predicate P on links columns.

`extract(MN,ways,clus)` – Extract from the multiway network MN a subnetwork determined by selected ways and corresponding clusters (vectors containing list of nodes).

`flatten(MN,col,by,FUN="sum")` – Removes ways that are not listed in the vector by and applies FUN on the weights col for so obtained duplicates.

A selected way \mathcal{V}_i is removed from the network.

$$\mathcal{V}' = (\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_{i-1}, \mathcal{V}_{i+1}, \dots, \mathcal{V}_k)$$

$$w'(v_1, v_2, \dots, v_{i-1}, v_{i+1}, \dots, v_k) = \sum_{v \in \mathcal{V}_i} w(v_1, v_2, \dots, v_{i-1}, v, v_{i+1}, \dots, v_k)$$



Some functions

Multiway
networks

V. Batagelj

It can be used, by contains a complete list of ways, to aggregate duplicate links in the original data – simplification.

`reorderlinks(MN, per=NULL)` – Reorders the links (rows) by the given permutation `per`.

`per = NULL` - random, `per = permutation vector`, `per = expression`

`joinways(MN, way1, way2, way3, sep="÷")` – Replaces a given pair of ways `way1` and `way2` with a new way `way3`:

$$way_3 = \{a \div b : \exists e \in \mathcal{L} : e(way_1) = a \wedge e(way_2) = b\}$$

It can be used for combining ways of small size into a new way with a richer range.



Robins Bank

Multiway
networks

V. Batagelj

Data from the paper Pattison, Wasserman, Robins, Kanfer:
Statistical Evaluation of Algebraic Constraints for Social Networks. J
of Math Psychology, 44(2000)4, 536-568.
I copied the matrices from the paper, put them into Pajek format,
and saved them as a list of arcs to "bankArc.csv".

```
> wdir <- "C:/Users/vlado/docs/papers/2022/ifcs2022/genova/MWnets/Bank"
> setwd(wdir)
> library(jsonlite)
> V <- read.table("https://raw.githubusercontent.com/bavla/ibm3m/master/data/bankArc.csv")
> names(V) <- c("P1","P2","R")
> V$w <- rep(1,nrow(V))
> S <- c("BM","DM","A1","A2","A3","T1","T2","T3","T4","T5","T6")
> P <- c("branch manager", "deputy manager", "service adviser 1",
+ "service adviser 2", "service adviser 3", "teller 1", "teller 2",
+ "teller 3", "teller 4", "teller 5", "teller 6")
> R <- c("Advice seeking", "Close friendship", "Satisfying interaction", "Confiding")
> Q <- c("advice", "friend", "satisfy", "confide")
> info <- list(network="Robins bank",
+ title="Robins: organizational structure in a bank branch",
+ by="Garry Robins",
+ ref="Pattison, et al: Statistical Evaluation of Algebraic Constraints for Social Networks. J
+ href="https://www.sciencedirect.com/science/article/pii/S0022249699912610",
+ creator="V. Batagelj",
+ date=date() )
> ways=list(P1="first person",P2="second person",R="relation")
> nodes=list(P1=data.frame(ID=S,long=P),P2=data.frame(ID=S),R=data.frame(ID=Q,long=R))
> MN <- list(format="MWnets",info=info,ways=ways,nodes=nodes,links=V,data=list())
> write(toJSON(MN),"RobinsBank.json")
```



Data from the paper

Multiway
networks

V. Batagelj

	Label	Relation	Label	Relation
	<i>Advice-seeking</i>	0 1 1 1 0 0 0 0 0 0 0	<i>Close</i>	0 1 1 1 1 0 0 0 0 0 0
A		1 0 0 0 0 0 0 0 0 0 0	<i>friendship</i>	0 0 0 0 0 0 0 0 0 0 0
		0 1 0 0 0 0 0 0 0 0 0	F	0 0 0 1 0 1 0 0 0 0 0
		1 0 0 0 0 0 0 0 0 0 0		0 0 1 0 0 1 0 0 0 0 1
		0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0
		0 0 1 1 1 0 0 0 0 0 0		0 0 1 1 0 0 0 0 0 0 1
		1 1 1 1 1 1 0 0 0 0 0		0 0 1 1 0 1 0 0 1 0 0
		1 1 1 1 0 1 1 0 1 0 0		0 0 0 1 0 0 0 0 1 0 0
		1 1 1 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 1 0 0 0
		1 0 0 0 0 0 0 0 0 0 0		0 0 0 1 0 0 0 0 0 0 0
		0 0 1 1 0 1 1 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0
	<i>Satisfying</i>	0 0 0 0 0 0 0 0 0 0 0	<i>Confiding</i>	0 1 1 1 0 0 0 0 0 0 0
	<i>interaction</i>	1 0 1 1 1 1 1 1 1 1	C	1 0 0 0 0 0 0 0 0 0 0
S		1 1 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0
		1 1 1 0 1 1 1 1 1 1 1		1 0 0 0 0 0 0 0 0 0 0
		1 1 1 1 0 1 1 1 1 1 1		1 1 1 1 0 1 1 0 0 0
		0 0 1 1 0 0 0 0 0 0 0		0 0 1 1 0 0 0 0 0 0 0
		1 0 1 1 1 1 0 1 0 0 0		0 0 1 1 1 0 0 0 0 0 0
		0 0 1 1 1 0 1 0 1 0 1		1 0 0 1 0 0 0 0 1 0 0
		0 0 0 0 0 0 0 0 0 0 0		1 0 0 0 0 0 0 1 0 0 0
		0 0 0 1 0 0 0 0 0 0 0		1 0 0 1 0 0 0 0 0 0 0
		0 0 0 1 0 1 1 0 0 1 0		0 0 0 1 0 1 1 0 0 1 0



RobinsBank.json structure

Multiway
networks

V. Batagelj

```
> MN <- fromJSON("https://raw.githubusercontent.com/bavla/ibm3m/master/data/RobinsBank.json")
> str(MN)
List of 6
$ format: chr "MWNets"
$ info :List of 7
..$ network: chr "Robins bank"
..$ title : chr "Robins: organizational structure in a bank branch"
..$ by : chr "Garry Robins"
..$ ref : chr "Pattison, et al: Statistical Evaluation of Algebraic Constraints for Social Networks"
..$ href : chr "https://www.sciencedirect.com/science/article/pii/S0022249699912610"
..$ creator: chr "V. Batagelj"
..$ date : chr "Wed Mar 29 03:29:28 2023"
$ ways :List of 3
..$ P1: chr "first person"
..$ P2: chr "second person"
..$ R : chr "relation"
$ nodes :List of 3
..$ P1:'data.frame': 11 obs. of 2 variables:
...$ ID : chr [1:11] "BM" "DM" "A1" "A2" ...
...$ long: chr [1:11] "branch manager" "deputy manager" "service adviser 1" "service adviser 2" ...
..$ P2:'data.frame': 11 obs. of 1 variable:
...$ ID: chr [1:11] "BM" "DM" "A1" "A2" ...
..$ R:'data.frame': 4 obs. of 2 variables:
...$ ID : chr [1:4] "advice" "friend" "satisfy" "confide"
...$ long: chr [1:4] "Advice seeking" "Close friendship" "Satisfying interaction" "Confiding in"
$ links :'data.frame': 128 obs. of 4 variables:
..$ P1: int [1:128] 1 1 1 2 3 4 6 6 7 ...
..$ P2: int [1:128] 2 3 4 1 2 1 3 4 5 1 ...
..$ R : int [1:128] 1 1 1 1 1 1 1 1 1 ...
..$ w : num [1:128] 1 1 1 1 1 1 1 1 1 ...
$ data : list()
```



Converting a data frame into a multiway network

Olympics

Multiway
networks

V. Batagelj

`DF2MWN(DF,vars,w=weights, network=name, title=tit)` – builds the basic multiway network structure from a given data frame *DF* for selected variables *vars* and weights *weights*.

An example is the construction of the Olympics network. An interesting data set *120 years of Olympic history: athletes and results* in the form of a data frame is available at [Kaggle](#).

From it I built the multiway network of summer medals.

```
> MT <- DF2MWN(OS,c("Name","Games","Team","NOC","Year","Sport","Event","Sex","Medal"),  
+   w=c("Age","Height","Weight"),network="Olympic16S0",title="Summer Olympic medals till 2016")
```

For details see [Summer Olympic medals till 2016](#).



3D layout

To get some insight into the structure of the network and obtained cores we can inspect their 3D layouts in X3D format. They are created using the MWnets function `mwnX3D`.

```
mwnX3D(MN,u,v,z,w,lu="ID",lv="ID",lz="ID",
pu=NULL,pv=NULL,pz=NULL,shape="Box",col=c(1,0,0),
bg=c(0.8,0.8,0.8),maxsize=1,file="MWnets.x3d") – Exports
a 3D layout description of the 3-way (sub)network of network MN on
ways u, v, z with the weight w in X3D format to the selected file.
```

The parameters `lu`, `lv`, `lz` are the names of node properties containing node labels. The vectors `pu`, `pv`, `pz` contain the permutations of nodes in the corresponding ways. The parameter `shape` can be either "Box" or "Sphere". The parameter `col` specifies the color of each link, and `bg` the background color. The parameter `maxsize` controls the maximum size of the shape.



Inspecting a 3D layout

Multiway
networks

V. Batagelj

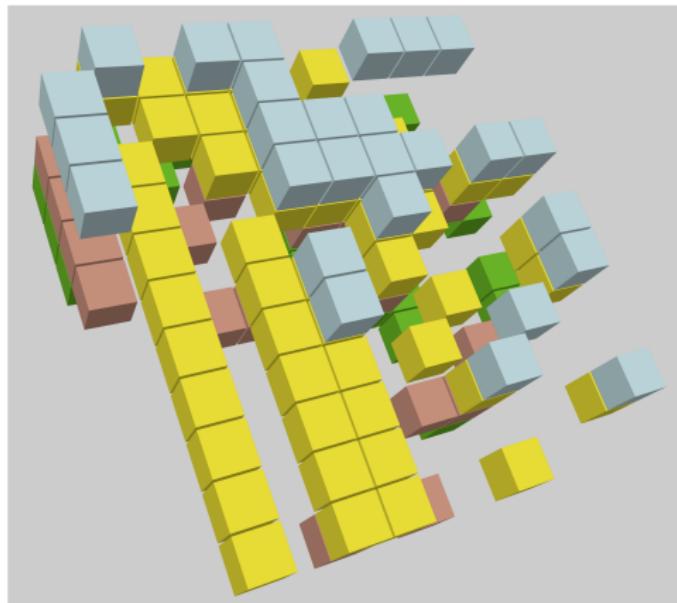
The 3D layouts can be interactively inspected using special viewers such as **view3dscene** [?]. They can be also included in HTML web pages and inspected (rotate, move, zoom in/out) using a Javascript-based library X3DOM [?]. Inspecting a 3D layout the user can also dynamically get information about the selected link (shape) by clicking on it.

The current implementation of the function `mwnX3D` provides only the basic functionalities – for a given simple/simplified multiway network displaying three ways by given orderings, coloring each shape, and the volume of the shape representing a link is proportional to its weight. It will be improved.

In producing the layout it is very important to provide "clever" orderings of the ways. For ordinal ways we usually stick to natural ordering; for nominal (categorical) ways the ordering is often determined using clustering or by sorting some important quantity defined on a selected way – EDA: property \sim structure.

<http://vladowiki.fmf.uni-lj.si/doku.php?id=vlado:work:2m:mwn:x3d:inline>

Clicked link description: link 104 : branch manager , service adviser 2 , confide , 1



3D layout



Projections of two-mode networks

Multiway
networks

V. Batagelj

Let $\mathcal{N} = ((\mathcal{U}, \mathcal{V}), \mathcal{L}, w)$ be a weighted two-mode network with a matrix $\mathbf{W} = [w[u, v]]$. A usual approach to its analysis is to *project* it to the set \mathcal{U} or to the set \mathcal{V} and analyze the so obtained ordinary (one-mode) network.

The projection to the set \mathcal{U} is determined by the matrix $\mathbf{C} = [c[u, t]] = \mathbf{W}\mathbf{W}^T$; and to the set \mathcal{V} by the matrix $\mathbf{W}^T\mathbf{W}$.

$\mathcal{N}_C = (\mathcal{U}, \mathcal{L}_C, c)$, where $\mathcal{L}_C = \{(u, t) : c[u, t] \neq 0\}$ and for $(u, t) \in \mathcal{L}_C : c(u, t) = c[u, t]$.

$$c[u, t] = \sum_{v \in \mathcal{V}} w[u, v] \cdot w^T[v, t] = \sum_{v \in \mathcal{V}} w[u, v] \cdot w^T[t, v]$$



Projection to a selected way

Because of the reordering option, we can assume that we selected the way \mathcal{V}_1 . A projection to a selected way is a generalization of the projection of two-mode networks. The projection creates an ordinary weighted network $(\mathcal{V}_1, \mathcal{A}, p)$, $\mathcal{A} \subseteq \mathcal{V}_1 \times \mathcal{V}_1$ and $p : \mathcal{A} \rightarrow \mathbb{R}$. Let $u, t \in \mathcal{V}_1$ then

$$p(u, t) = \sum_{(v_2, \dots, v_k) \in \mathcal{V}_2 \times \dots \times \mathcal{V}_k} w(u, v_2, \dots, v_k) \cdot w(t, v_2, \dots, v_k)$$

This network can be analyzed using traditional methods for the analysis of weighted networks. Sometimes it is more appropriate to apply projection(s) to a normalized version of the original multi-way network.

The operation can be generalized to the product of compatible multiway networks.



Salton index has the following properties

Multiway
networks

V. Batagelj

From the projection p we can get the corresponding measure of similarity – *Salton index* $S(u, t)$ [?]

$$S(u, t) = \frac{p(u, t)}{\sqrt{p(u, u) \cdot p(t, t)}}$$

that can be used for clustering the set \mathcal{V}_1 .

The Salton index has the following properties

- ① $S(u, t) \in [-1, 1]$
- ② $S(u, t) = S(t, u)$
- ③ $S(u, u) = 1$
- ④ $w : L \rightarrow \mathbb{R}_0^+ \Rightarrow S(u, t) \in [0, 1]$
- ⑤ $S(\alpha u, \beta t) = S(u, t), \quad \alpha, \beta > 0$
- ⑥ $S(\alpha u, u) = 1, \quad \alpha > 0$



Robins Bank analysis

Multiway
networks

V. Batagelj

```
> source("https://raw.githubusercontent.com/bavla/ibm3m/master/
        multiway/MWnets.R")
> Cou <- projection(MN,"P1","w")
> Sau <- salton(Cou); Du <- as.dist(1-Sau); Du[is.na(Du)] <- 1
> tu <- hclust(Du,method="ward")
> plot(tu,hang=-1,cex=1,main="Robins Bank - P1 / Ward")

> Cov <- projection(MN,"P2","w")
> Sav <- salton(Cov); Dv <- as.dist(1-Sav); Dv[is.na(Dv)] <- 1
> tv <- hclust(Dv,method="ward")
> plot(tv,hang=-1,cex=1,main="Robins Bank - P2 / Ward")

> clrs <- c("green4","tomato3","gold1","cadetblue3")
> pie(rep(1,4), col=clrs)
> CC <- col2rgb(clrs)/255
> Col <- cbind(CC[1,MN$links$R],CC[2,MN$links$R],CC[3,MN$links$R])
> MN$nodes$P2$long <- MN$nodes$P1$long
> I <- inv(tu$order); J <- inv(tv$order); K <- c(3,4,1,2)
> mwnX3D(MN,"P1","P2","R","w",maxsize=0.95,pu=I,pv=I,pz=K,
+   lu="long",lv="long",col=Col,file="RbankClu1.x3d")
> mwnX3D(MN,"P1","P2","R","w",maxsize=0.95,pu=J,pv=J,pz=K,
+   lu="long",lv="long",col=Col,file="RbankClu2.x3d")
> mwnX3D(MN,"P1","P2","R","w",maxsize=0.95,pu=I,pv=J,pz=K,
+   lu="long",lv="long",col=Col,file="RbankClu3.x3d")
```

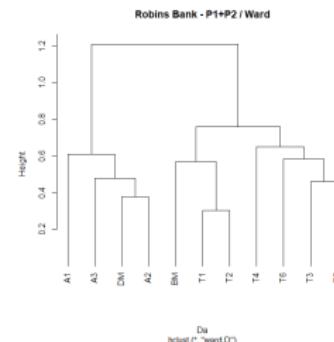
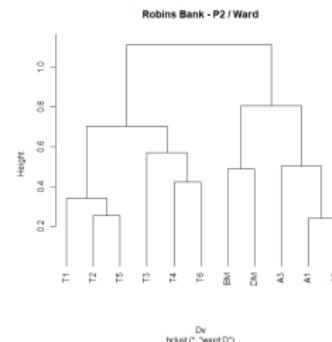
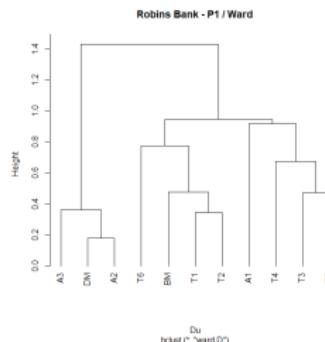
wiki

Clusterings

Multiway
networks

V. Batagelj

```
> Da <- (Du+Dv)/2; ta <- hclust(Da,method="ward")
> plot(ta,hang=-1,cex=1,main="Robins Bank - P1+P2 / Ward")
> L <- inv(ta$order)
> mwnX3D(MN,"P1","P2","R","w",maxsize=0.95,pu=L,pv=L,pz=K,
+   lu="long",lv="long",col=Col,file="RbankClu4.x3d")
```



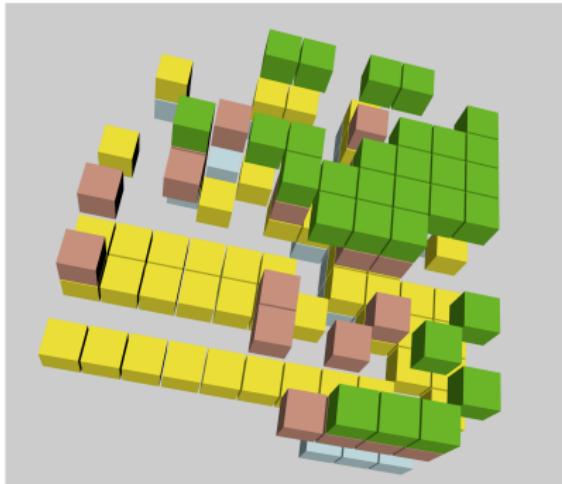


Robins Bank 3D layouts

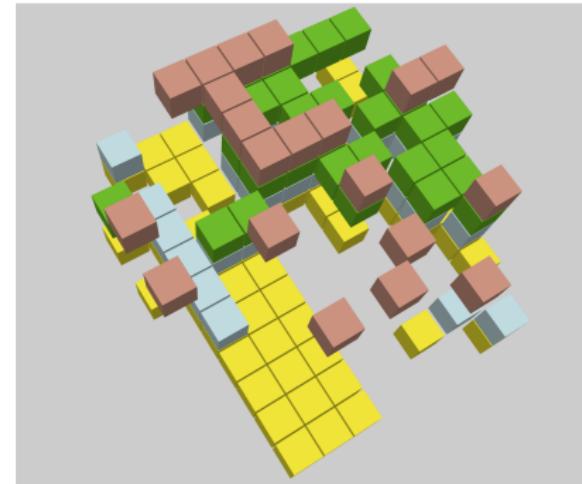
Multiway
networks

V. Batagelj

Clicked link description: link 34 : branch manager , service adviser 3 , friend , 1



Clicked link description: link 33 : branch manager , service adviser 2 , friend , 1





Recoding (binning)

Let $\mathcal{B} = \{B_1, B_2, \dots, B_s\}$ be a partition of \mathbb{R} . The *recoding* transformation transforms the weight function w into a new weight w' determined for a link $e \in L$ as

$$w'(e) = i \Leftrightarrow w(e) \in P_i$$

It could be used also for transforming weights to ways.

Code 0 corresponds to the case $e = (v_1, v_2, \dots, v_k) \notin L$ which is usually equivalent to $w(v_1, v_2, \dots, v_k) = 0$ in the array representation. If 0 is also a legal weight value we have to introduce another zero, \square , that indicates the absence of a link.

A special case is a *binarization* for which $B_0 = \{0\}$ and $B_1 = \mathbb{R} \setminus B_0$.

Recoding is often used to get more readable matrix visualizations of a given network. [wiki](#)

```
Mc <- recodecol2bins(MN,"w","code",bins=c(1,5,10,20,Inf))
Mr <- recodeway2part(MN,"prov","IDreg","regs","region")
Mre <- flatten(Mr,"w",c("regs","univ","prog","year"))
```



European airports and flight companies dataset

Multiway
networks

V. Batagelj

The *European airports and flight companies* dataset was collected by A. Cardillo and his collaborators for the paper

A. Cardillo, J. Gómez-Gardeñes, M. Zanin, M. Romance, D. Papo, F. del Pozo, S. Boccaletti: *Emergence of Network Features from Multiplexity*, *Scientific Reports* 3, 1344 (2013).

It is an example of a multiway network. It contains 450 airports making two ways `airA` (departure airports) and `airB` (arrival airports), the third way is `line` (37 flight companies). There are 7176 links (*from, to, line*) $\in \mathcal{L}$ telling that there is a flight provided by the company *line* from the airport *from* to the airport *to*. Each link has the weight $w = 1$.

The multiway network `AirEu2013` represented in JSON format and an R package `MWNets` for analysis of multiway networks are available at GitHub/Bavla.



Dataset

Multiway
networks

V. Batagelj

```
List of 6
$ format: chr "MWhnets"
$ info :List of 7
..$ network: chr "AirEu2013"
..$ title : chr "Air Transportation Multiplex"
..$ ref : chr "Cardillo A. et al. Emergence of network features from multiplexity, Scientific"
..$ href : chr "http://complex.unizar.es/~atnmultiplex/"
$ ways :List of 3
..$ airA: chr "first airport"
..$ airB: chr "second airport"
..$ line: chr "airline"
$ nodes :List of 3
..$ airA:'data.frame': 450 obs. of 7 variables:
...$ ID : chr [1:450] "LCLK" "EDDF" "EDDK" "EGNX" ...
...$ lon : num [1:450] 33.63 8.57 7.14 -1.33 -3.41 ...
...$ lat : num [1:450] 34.9 50 50.9 52.8 50.7 ...
...$ iata : chr [1:450] "LCA" "FRA" "CGN" "EMA" ...
...$ long : chr [1:450] "Larnaca International Airport" "Frankfurt Airport" "Cologne Bonn"
...$ region : chr [1:450] "Larnaka" "Hessen" "Nordrhein-Westfalen" "England" ...
...$ country: chr [1:450] "CY" "DE" "DE" "GB" ...
..$ airB:'data.frame': 450 obs. of 2 variables:
...$ ID : chr [1:450] "LCLK" "EDDF" "EDDK" "EGNX" ...
...$ long: chr [1:450] "Larnaca International Airport" "Frankfurt Airport" "Cologne Bonn Air
..$ line:'data.frame': 37 obs. of 1 variable:
...$ ID: chr [1:37] "Lufthansa" "Ryanair" "Easyjet" "British A" ...
$ links :'data.frame': 7176 obs. of 4 variables:
..$ airA: int [1:7176] 1 1 2 2 2 2 2 2 2 ...
..$ airB: int [1:7176] 2 38 1 7 8 10 14 15 17 18 ...
..$ line: int [1:7176] 1 1 1 1 1 1 1 1 1 ...
..$ w : int [1:7176] 1 1 1 1 1 1 1 1 1 ...
$ data :List of 1
..$ Eu:'data.frame': 40 obs. of 6 variables:
...$ alpha2 : chr [1:40] "AT" "BA" "BE" "BG" ...
...$ alpha3 : chr [1:40] "AUT" "BIH" "BEL" "BGR" ...
...$ Ccode : int [1:40] 40 70 56 100 756 196 203 276 208 233 ...
...$ long : chr [1:40] "Austria" "Bosnia and Herzegovina" "Belgium" "Bulgaria" ...
...$ region : chr [1:40] "Europe" "Europe" "Europe" "Europe" ...
...$ subregion: chr [1:40] "Western Europe" "Southern Europe" "Western Europe" "Eastern Europe"
```



Derived networks

Multiway
networks

V. Batagelj

Using transformations of multiway networks we can create ordinary networks and analyze them using network analysis tools.

```
> wdir <- "C:/Users/vlado/DL/data/multi/cores/air"; setwd(wdir)
> library(jsonlite)
> source("https://raw.githubusercontent.com/bavla/Rnet/master/R/Pajek.R")
> source("https://raw.githubusercontent.com/bavla/ibm3m/master/multiway/MWnets.R")
> MN <- fromJSON("https://raw.githubusercontent.com/bavla/ibm3m/master/data/AirEu2013Ext.json")
> str(MN)
> Net <- flatten(MN,"w",c("airA","airB"))
> head(Net$links)
> mwn2net(Net,"airA","airB",ID1="long",ID2="long",w="w",twomode=FALSE,Net="lines.net")
```

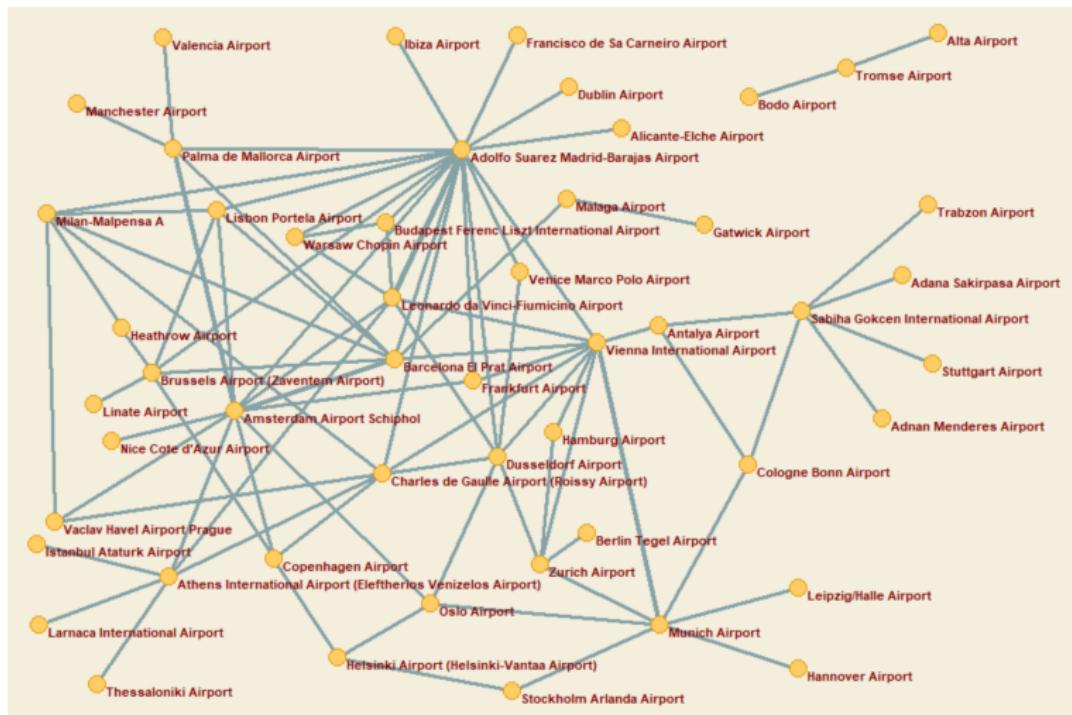
We obtained a network of flights between airports with the weight $w(a, b)$ counting the number of companies flying from the airport a to the airport b .

We analyze it in **Pajek**: draw, transform to undirected (max), info weights, link-cut at level 3, draw; skeletons, cores, islands.

Link-cut at level 3

Multiway
networks

V. Batagelj





Clustering

Multiway
networks

V. Batagelj

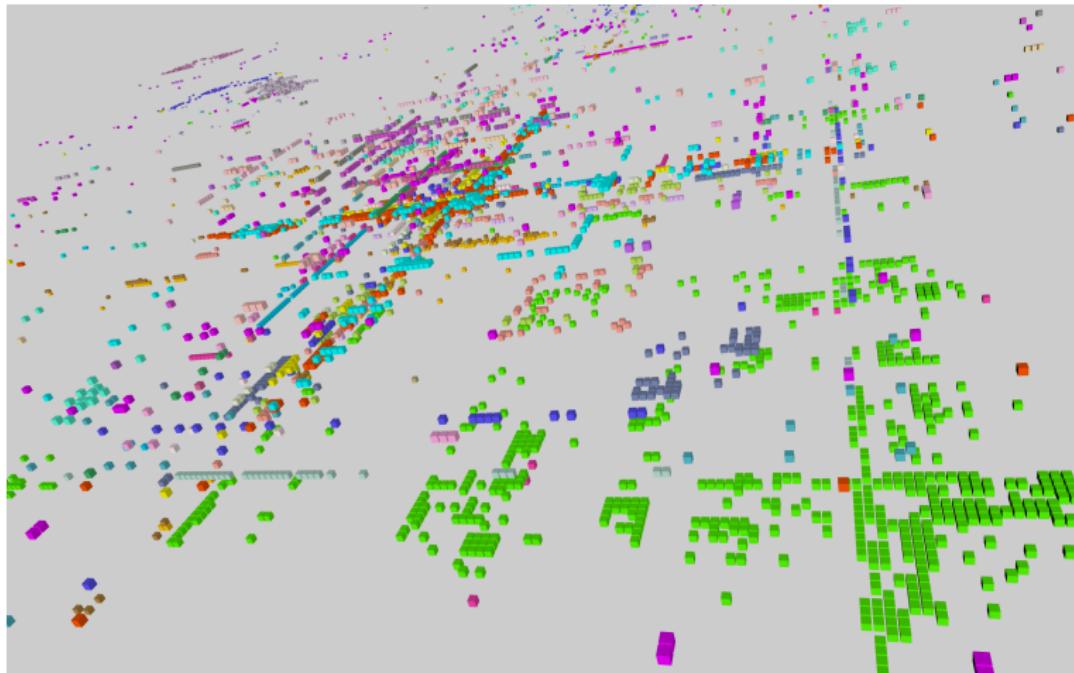
```
> setwd("C:/test/cores")
> library(jsonlite)
> library(Polychrome)
> source("https://raw.githubusercontent.com/bavla/ibm3m/master/multiway/MWnets.R")
> MN <- fromJSON("https://raw.githubusercontent.com/bavla/ibm3m/master/data/AirEu2013Ext.json")
> str(MN)
> MN$nodes$line$short <- substr(MN$nodes$line$ID,1,4)
> Cou <- projection(MN,"airA","w")
> Sau <- salton(Cou); Du <- as.dist(1-Sau); Du[is.na(Du)] <- 1
> tu <- hclust(Du,method="ward")
> plot(tu,hang=-1,cex=0.2,main="EU airports - airports / Ward")
> Coz <- projection(MN,"line","w")
> Saz <- salton(Coz); Dz <- as.dist(1-Saz); Dz[is.na(Dz)] <- 1
> tz <- hclust(Dz,method="ward")
> plot(tz,hang=-1,cex=0.8,main="EU airports - companies / Ward")
> I <- inv(tu$order); K <- inv(tz$order)
> CC <- col2rgb(createPalette(37,c("#ff0000","#00ff00","#0000ff")))/255
> Col <- cbind(CC[1,MN$links$line],CC[2,MN$links$line],CC[3,MN$links$line])
> mwnX3D(MN,"airA","airB","line","w",maxsize=0.85,pu=I,pv=I,pz=K,lu="long",lv="long",
+ col=Col,file="EUair.x3d")
> row.names(Cou) <- MN$nodes$airA$long
> svg("airports.svg",width=18,height=8)
> plot(tu,hang=-1,cex=0.2,main="EU airports - airports / Ward")
> dev.off()
```

Airports dendrogram, 3D layout.

Screenshot of a part of the 3D layout of AirEu2013

Multiway
networks

V. Batagelj

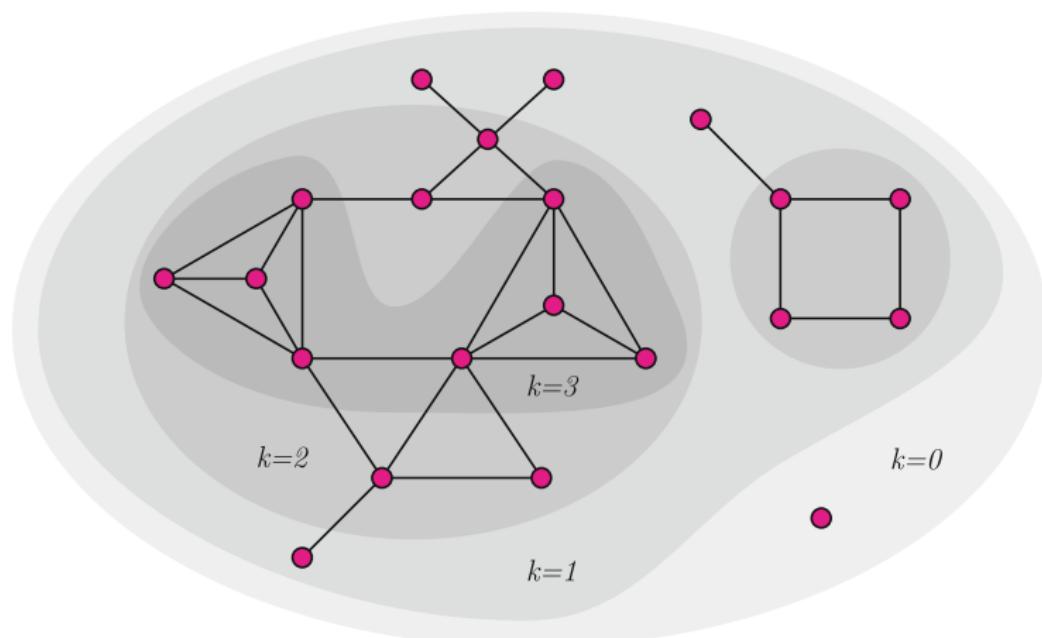


The green links in the bottom right corner are provided by Ryan air – it is mostly linking airports that are not linked by other companies.

Cores in networks

Multiway
networks

V. Batagelj



Extended to other node properties, two-mode networks, and temporal networks [?, ?].



Node properties

A list of subways $\mathcal{C} = (C_1, C_2, \dots, C_r)$, $C_i \subseteq \mathcal{V}_{h(i)}$ we will call a *selection*. Note that the selection \mathcal{C} defines a function $h : 1..r \rightarrow 1..k$. For a link $e \in \mathcal{L}$ we introduce abbreviations

$$h(e) \equiv (e(h(1)), e(h(2)), \dots, e(h(r)))$$

for a *selected part* of the link e ; and

$$h(e) \in \mathcal{C} \equiv \forall i \in 1..r : e(h(i)) \in C_i$$

for a statement that all nodes of the selected part of the link e belong to selected subways.

For a node $u \in \mathcal{V}_i$ and a selection \mathcal{C} , we call a *star* in the node u for the selection \mathcal{C} , the set

$$S(u, \mathcal{C}) = \{e \in \mathcal{L} : e(i) = u \wedge h(e) \in \mathcal{C}\}.$$



... Node properties

Multiway
networks

V. Batagelj

For a node $u \in \mathcal{V}_i$ we call the set of *neighbors* of the node u in the way \mathcal{V}_j for the selection \mathcal{C} , the set of nodes $N(u, \mathcal{C}, \mathcal{V}_j) = \mathcal{V}_j(S(u, \mathcal{C}))$.

Using stars and sets of neighbors we can define some node property functions on multiway networks. For a node $u \in \mathcal{V}_i$ and a selection \mathcal{C}

degree: $p_d(u, \mathcal{C}) = \text{card}(S(u, \mathcal{C}))$

weighted degree for the weight w : $p_S(u, \mathcal{C}; w) = \sum_{e \in S(u, \mathcal{C})} w(e)$

diversity of the way \mathcal{V}_j : $p_\delta(u, \mathcal{C}, \mathcal{V}_j) = \text{card}(N(u, \mathcal{C}, \mathcal{V}_j))$

contribution of a (measured) node property/attribute a to the way \mathcal{V}_j : $p_c(u, \mathcal{C}, \mathcal{V}_j; a) = \sum_{v \in N(u, \mathcal{C}, \mathcal{V}_j)} a(v)$

We say that the node property function $p(v, \mathcal{C})$:

- is *local* iff: $p(v, \mathcal{C}) = p(v, N(v, \mathcal{C}))$ for all $v \in \mathcal{V}$.
- is *monotonic* iff: $\mathcal{C}_1 \subset \mathcal{C}_2 \Rightarrow \forall v \in \mathcal{V}: p(v, \mathcal{C}_1) \leq p(v, \mathcal{C}_2)$.



Node properties in MWnets

Multiway
networks

V. Batagelj

Currently in MWnets the node property function(s) can be selected among

- p_d : pDeg(MN,u,cip,C)
- p_s : pWsum(MN,u,cip,C,weight=))
- p_M : pWmax(MN,u,cip,C,weight=))
- p_δ : pDiv(MN,u,cip,C,way=)
- p_c : pAttr(MN,u,cip,C,way=,attr=,FUN=sum)

The user can implement and use in MWcore his/her own node property functions.



Important nodes

Multiway
networks

V. Batagelj

Using node properties we can identify important nodes (with respect to the selected property). In network analysis, such properties are degree, weighted degree, closeness, betweenness, etc. For multiway networks, similar concepts are still to be developed.

```
> wdir <- "C:/Users/vlado/DL/data/multi/cores/air"
> setwd(wdir)
> library(jsonlite)
> source("https://raw.githubusercontent.com/bavla/ibm3m/master/multiway/MWnets.R")
> MN <- fromJSON("https://raw.githubusercontent.com/bavla/ibm3m/master/data/AirEu2013Ext.json")
> str(MN)
> cw <- c("airA","airB","line")
> # complete selection
> C <- lapply(cw, \x) 1:nrow(MN$nodes[[x]])); names(C) <- cw
> for(line in 1:37) cat(line,MN$nodes$line$ID[line],pWsum(MN,line,c(3,1,2),C,"w"),"\n")
> Dep <- rep(0,450)
> for(a in 1:450) Dep[a] <- pWsum(MN,a,c(1,2,3),C,"w")
> I <- order(Dep,decreasing=TRUE)
> top20=Dep[I[1:20]]
> names(top20) <- MN$nodes$airA$long[I[1:20]]
> top20
> Lin <- rep(0,37)
> for(line in 1:37) Lin[line] <- pWsum(MN,line,c(3,1,2),C,"w")
> I <- order(Lin,decreasing=TRUE)
> top10 <- Lin[I[1:10]]
> names(top10) <- MN$nodes$line$ID[I[1:10]]
> top10
```



... Important nodes

Multiway
networks

V. Batagelj

> top20

Amsterdam Airport Schiphol	156
Adolfo Suarez Madrid-Barajas Airport	152
Munich Airport	139
Barcelona El Prat Airport	137
Frankfurt Airport	127
Gatwick Airport	124
Vienna International Airport	122
Leonardo da Vinci-Fiumicino Airport	119
Charles de Gaulle Airport (Roissy Airport)	118
London Stansted Airport	118
Dusseldorf Airport	115
Dublin Airport	110
Palma de Mallorca Airport	106
Brussels Airport (Zaventem Airport)	104

> top10

Ryanair	1202
Easyjet	614
Lufthansa	488
Air Berlin	368
Netjets	360
Turkish A	236
SAS	220
Flybe	198
Alitalia	186
Wizz Air	184



Cores in multiway networks

Let $\mathcal{C} = (C_1, C_2, \dots, C_k) \subseteq \mathcal{V}$ be a selection, $\mathbf{p} = (p_i)$ be a list of monotonic node property functions over the selected ways, and $\mathbf{t} = (t_i)$ a list of the corresponding thresholds.

The multiway subnetwork $\text{Core}(\mathbf{p}, \mathbf{t}) = (\mathcal{C}, \mathcal{L}(\mathcal{C}))$ induced by the selection \mathcal{C} in a multiway network \mathcal{N} is a *generalized multiway core* for node property functions \mathbf{p} at levels \mathbf{t} if and only if it holds that for all $v \in C_i : p_i(v, \mathcal{C}) \geq t_i$ and \mathcal{C} is the maximal such selection.

Then for monotonic node property functions \mathbf{p} the generalized core can be determined by the following algorithm

```
1: function COREMW( $\mathcal{N}, \mathbf{p}, \mathbf{t}$ )
2:    $\mathcal{C} \leftarrow (\mathcal{V}_{h(1)}, \mathcal{V}_{h(2)}, \dots, \mathcal{V}_{h(r)})$ 
3:   while  $\exists u \in C_i : p_i(u, \mathcal{C}) < t_i$  do  $C_i \leftarrow C_i \setminus \{u\}$ 
4:   return  $\mathcal{C}$ 
5: end function
```



Cores in multiway networks

Multiway cores are partially nested

$$\mathbf{t}' < \mathbf{t} \Rightarrow \text{Core}(\mathbf{p}, \mathbf{t}) \subseteq \text{Core}(\mathbf{p}, \mathbf{t}')$$

In an elaboration of this algorithm, we have many options. Again the result does not depend on the node elimination order. Here we present a simple algorithm that is eliminating nodes way-wise.

The core conditions are given in a list P . Each entry of the list is a triple (p, t, args) where p is a node property function, t is the corresponding level, and args is a list of ways on which the function p is defined. It is assumed that the node u belongs to the way $\text{args}[1]$.

In an elaboration of this algorithm, we have many options. Again the result does not depend on the node elimination order. Here we present a simple algorithm that is eliminating nodes way-wise.

It is implemented as the function `MWcore(MN,P)` where `MN` is a multiway network and `P` is a list of core conditions. The user can implement and use in `MWcore` his/her own node property functions.



Cores algorithm

```
1: function MWCORE( $\mathcal{N}, h, P$ )
2:    $\mathcal{C} \leftarrow (\mathcal{V}_{h(1)}, \mathcal{V}_{h(2)}, \dots, \mathcal{V}_{h(r)})$ 
3:   repeat
4:      $exit \leftarrow true$ 
5:     for  $(p, t, args) \in P$  do
6:        $R \leftarrow \emptyset; i \leftarrow args[1]$ 
7:       for  $u \in C_i$  do
8:         if  $p(\mathcal{N}, u, \mathcal{C}) < t$  then
9:            $R \leftarrow R \cup \{u\}; exit \leftarrow false$ 
10:          end if
11:        end for
12:         $C_i \leftarrow C_i \setminus R$ 
13:      end for
14:    until  $exit$ 
15:    return  $\mathcal{C}$ 
16: end function
```



Cores

Multiway
networks

V. Batagelj

```
> source("https://raw.githubusercontent.com/bavla/Rnet/master/R/Pajek.R")
> source("https://raw.githubusercontent.com/bavla/ibm3m/master/multiway/MWnets.R")
> cw <- c("airA","airB")
> ci <- unname(sapply(cw,\(x) which(names(MN$ways)==x)))
> pDIV <- function(MN,v,cip,C,...) pDiv(MN,v,cip,C,way="line")
> P <- list(
+   p1 = list(p = pDIV, t = 10, cwp = c("airA","airB"), cip = NULL),
+   p2 = list(p = pDIV, t = 10, cwp = c("airB","airA"), cip = NULL),
+   cways = list(cw=cw,ci=ci) )
> for(i in 1:(length(P)-1)) P[[i]]$cip <- unname(sapply(P[[i]]$cwp,\(x) which(cw==x)))
> str(P)
> P[[1]]$t <- 13; P[[2]]$t <- 13
> CC <- MWcore(MN,P)
> listCore(MN,CC,P)
> Ap <- MN$nodes$airA$long[CC[[1]]]

> w1 <- CC$airA; w2 <- CC$airB
> Score <- extract(MN,c("airA","airB"),c("w1","w2"))
> act <- as.integer(names(table(Score$links$line)))
> Rcore <- extract(Score,"line","act")
> str(Rcore)
> c27 <- glasbey.colors(27); CC <- col2rgb(c27)/255
> Col <- cbind(CC[1,Rcore$links$line],CC[2,Rcore$links$line],CC[3,Rcore$links$line])
> ts <- c(1,20,10,26,27,19,24,9,7,8,21,13,18,4,2,15,11,23,6,22,5,16,3,12,14,17,25,28)
> t <- inv(ts)
> qs <- c(1,5,10,19,3,7,12,4,16,13,21,8,20,6,11,18,17,2,23,9,15,24,25,14,22,27,26)
> qq <- inv(qs)
> mwnX3D(Rcore,"airA","airB","line","w",pu=t,pv=t,pz=qq,lu="long",lv="long",maxsize=0.85,
+   col=Col,file="EuAirCore13.x3d")

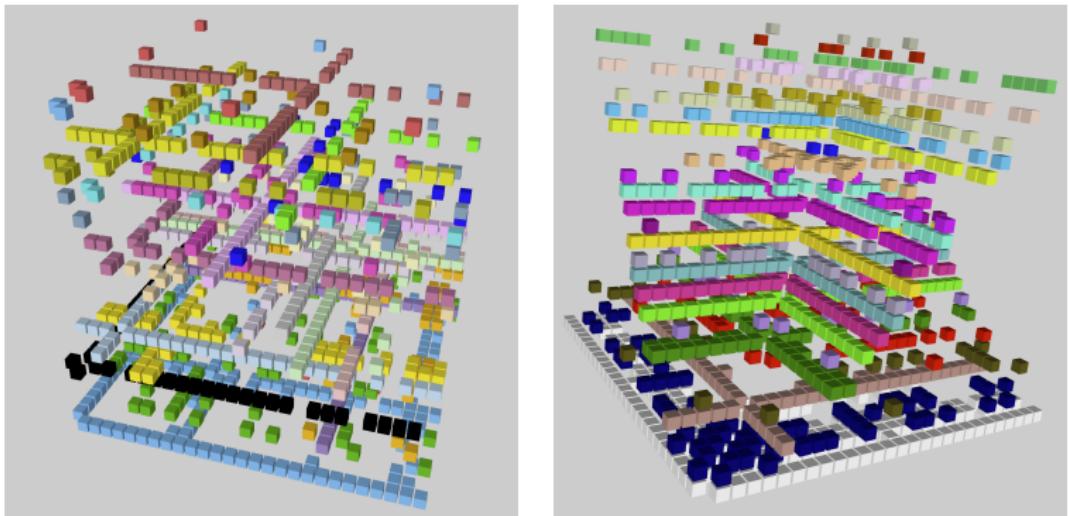
> percents(MN,Rcore,"airA","airB","line","w")
```

Core 3D layout, arXiv

3D layout of EUair diversity core at level 13 initial (left) and reordered by crosses (right)

Multiway
networks

V. Batagelj



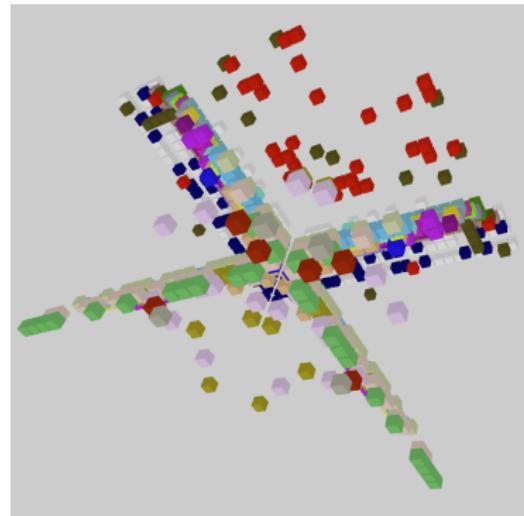
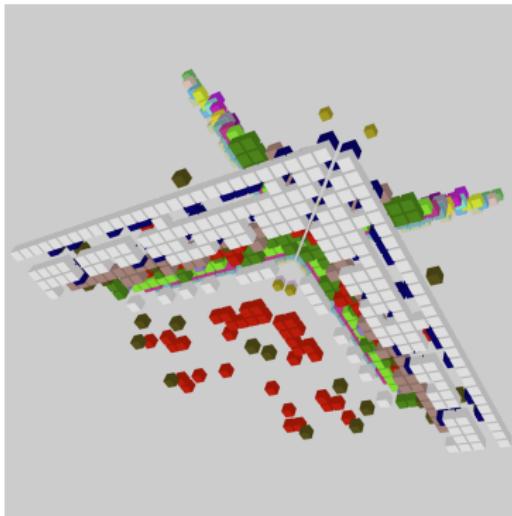
- (1) Frankfurt Ap, (2) Berlin Tegel Ap, (3) Munich Ap, (4) Dusseldorf Ap, (5) Hamburg Ap, (6) Zurich Ap, (7) Geneva Ap, (8) Milan-Malpensa Ap, (9) Copenhagen Ap, (10) Stockholm Arlanda Ap, (11) Heathrow Ap, (12) Warsaw Chopin Ap, (13) Vienna IAp, (14) Amsterdam Ap Schiphol, (15) Charles de Gaulle Ap (Roissy Ap), (16) Adolfo Suarez Madrid-Barajas Ap, (17) Barcelona El Prat Ap, (18) Malaga Ap, (19) Vaclav Havel Ap Prague, (20) Leonardo da Vinci-Fiumicino Ap, (21) Brussels Ap (Zaventem Ap), (22) Budapest Ferenc Liszt IAp, (23) Athens IAp (Eleftherios Venizelos Ap), (24) Ben Gurion Ap, (25) Henri Coanda IAp, (26) Venice Marco Polo Ap, (27) Sofia Ap, (28) Nice Cote d'Azur Ap.

3D layout of EUair diversity core at level 13

two views exposing companies based on multiple airports

Multiway
networks

V. Batagelj



(1) Lufthansa, (2) Air Berlin, (3) Swiss IAL, (4) Netjets, (5) Easyjet, (6) SAS, (7) Norwegian AS, (8) British A, (9) LOT Polish A, (10) Austrian A, (11) Niki, (12) KLM, (13) Transavia H, (14) Air France, (15) Iberia, (16) Air Nostrum, (17) Vueling A, (18) Ryanair, (19) Czech A, (20) Alitalia, (21) Brussels A, (22) European AT, (23) Malev HA, (24) Wizz Air, (25) Aegean A (26) Olympic Air, (27) TNT Airways.

The companies that were not providing any line between core airports are Turkish A, Flybe, TAP Portugal, Finnair, Air Lingus, Germanwings, Pegasus A, SunExpress, Air Baltic, and Wideroe.



Some observations

Multiway
networks

V. Batagelj

- Lufthansa and Air Berlin are serving lines from German airports Frankfurt, Berlin, Munich, Dusseldorf, Hamburg, and Milan in Italy
- Swiss IAL and Netjets are serving lines from Swiss airports Zurich and Geneva
- SAS and Norwegian AS are serving lines from Copenhagen and Stockholm
- Spanish airports Madrid, Barcelona, and Malaga are served by companies Iberia, Air Nostrum, Vueling, and Ryanair
- companies serving lines from their base airport: (British A, Heathrow), (LOT, Warsaw), (Austrian A, Niki; Vienna), (KLM, Transavia; Amsterdam), (Air France, CDG), (Czech A, Prague), (Alitalia, Fiumicino), (Brussels A, European AT; Brussels), (Malev, Budapest), (Aegean, Olympic Air; Athens); the only “irregular” link is by Iberia between Barcelona and Budapest



... Some observations

Multiway
networks

V. Batagelj

- Companies with dispersed services inside the core are Easyjet, Netjets, Wizz Air, European AT, and TNT Airways
 - Easyjet is serving lines from Milan and also from CDG, Amsterdam, Fiumicino, Madrid
 - Netjets is serving lines also from Barcelona, Venice, Vienna, and Nice
 - Wizz Air is serving lines from Budapest and Fiumicino
 - European AT has also some lines from Heathrow, Milan, Venice, Barcelona, and Madrid
 - TNT Airways is linking airports CDG and Athens, and Sofia and Henri Coanda



Olympics

Multiway
networks

V. Batagelj

The data set *120 years of Olympic history – athletes and results* with basic bio data on athletes and medal results from Athens 1896 to Rio 2016 is available at Kaggle [?]. The original data describe 134732 participants of the Summer and Winter Olympics. We transformed it into the multiway network `Olympics16S` about the medalists of the Summer Olympics till 2016 (available at GitHub/Bavla [?]). It contains data about 34088 medals. Omitting the athlete Name info and simplifying the network we obtain a multiway network with 10429 links on five ways (Games (29), NOC (147), Sport (52), Sex (2), Medal (3)), the weight w counting the medals, and additional weights Age, Height, and Weight.

We decided to search for a core with a large number of medals on the ways Games, NOC, and Sport.



Olympics core

Multiway
networks

V. Batagelj

After examining the distributions of values of the node property function p_S on all three ways we decided to use the core conditions list P that contains three triples

$(p = pWsum, t = 100, \text{args} = (\text{Games}, \text{NOC}, \text{Sport}))$,

$(p = pWsum, t = 100, \text{args} = (\text{NOC}, \text{Games}, \text{Sport}))$, and

$(p = pWsum, t = 100, \text{args} = (\text{Sport}, \text{Games}, \text{NOC}))$.

The obtained core was too large. We increased the thresholds to (500, 300, 350). The corresponding p_S -core

$\mathcal{C} = (C_{\text{Games}}, C_{\text{NOC}}, C_{\text{Sport}})$ is of order (577, 308, 390) – inside the core, each Olympics accounts for at least 577 medals, each country (NOC) for at least 308 medals, and each sport discipline for at least 390 medals.



... Olympics core

Multiway
networks

V. Batagelj

C_{Games} (25): 1988 Seoul (1405), 1984 Los Angeles (1354), 2008 Beijing (1352), 2012 London (1338), 2000 Sydney (1324), 2016 Rio de Janeiro (1320), 2004 Athens (1305), 1980 Moscow (1270), 1996 Atlanta (1251), 1976 Montreal (1220), 1992 Barcelona (1112), 1972 Munich (1080), 1920 Antwerp (1054), 1964 Tokyo (952), 1968 Mexico City (949), 1912 Stockholm (853), 1960 Rome (826), 1952 Helsinki (797), 1956 Melbourne + Stockholm (795), 1936 Berlin (754), 1948 London (731), 1924 Paris (681), 1908 London (656), 1928 Amsterdam (610), 1932 Los Angeles (577).

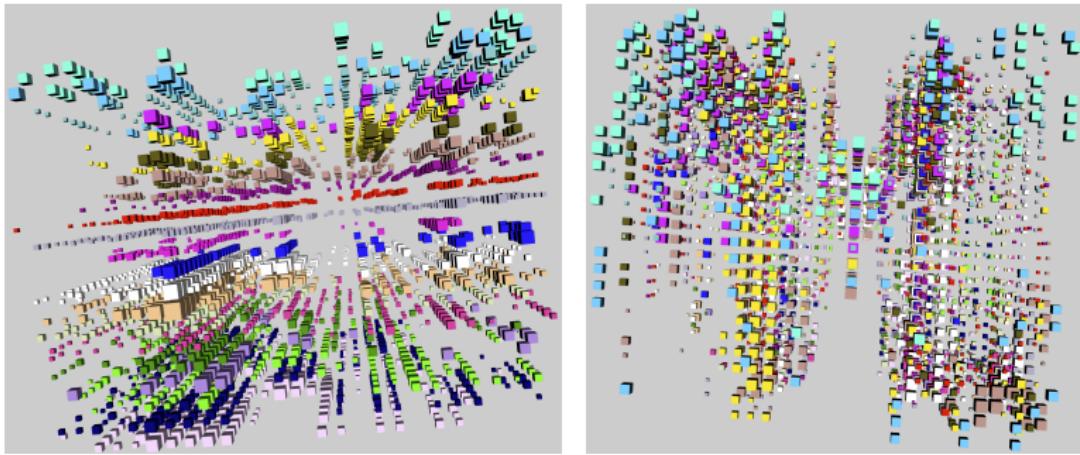
C_{NOC} (29): USA (4153), URS (2027), GBR (1594), GER (1573), ITA (1325), FRA (1184), AUS (1165), HUN (1052), SWE (1019), NED (854), GDR (843), RUS (776), CHN (662), JPN (661), ROU (632), CAN (613), NOR (546), POL (520), DEN (519), FRG (500), FIN (446), BRA (446), ESP (416), YUG (379), SUI (352), KOR (347), BEL (337), BUL (317), TCH (308).

C_{Sport} (21): Athletics (2911), Swimming (2815), Rowing (2535), Gymnastics (2043), Fencing (1584), Football (1216), Cycling (1129), Hockey (1126), Canoeing (1010), Shooting (995), Sailing (995), Wrestling (941), Handball (924), Basketball (914), Equestrianism (883), Water Polo (876), Volleyball (839), Boxing (596), Judo (425), Weightlifting (419), Diving (390).

Screenshot of a 3D layout of Olympics weighted degree core at levels (577, 308,390)

Multiway
networks

V. Batagelj



It turns out that the core contains $25566/34088 = 75\%$ of all medals, $3607/10429 = 34.6\%$ of all links, and occupies 6.9% of the total space. Note also that the set \mathcal{C}_{NOC} contains the Soviet Union and Russia; and Germany, West Germany, and East Germany.

The links belonging to the same sport discipline are of the same color. The Olympics are ordered chronologically. The order of NOCs (countries) and sport disciplines was determined using clustering.



Additional datasets

Multiway
networks

V. Batagelj

Vlado wiki: Multiway networks



Conclusions

Multiway
networks

V. Batagelj

① Work in progress

- ① more testing,
- ② improving some code (efficiency, programming solutions – objects?, dplyr?),
- ③ unification,
- ④ robustness.
- ② Additional data sets. New problems. Apply the methods to different data sets (test, experience, ideas)
- ③ Normalizations
- ④ Converting numerical properties (weights) into ways by recoding. Combining small ways into a large way by joining.
- ⑤ Julia and/or Python version
- ⑥ Additional Javascript support for X3DOM (slicing, ...) and SVG (additional options) visualization.



Acknowledgments

Multiway
networks

V. Batagelj

The computational work reported in this presentation was performed using R library MWnets. The code and data are available at Github/Bavla [?].

We would like to thank Andreas Plesch for extending the library **X3DOM** with support for Anchor's attribute description.

This work is supported in part by the Slovenian Research Agency (research program P1-0294 and research projects J5-2557, J1-2481 and J5-4596), and prepared within the framework of the COST action CA21163 (HiTEc).



References |

Multiway
networks

V. Batagelj

-  Batagelj, V (2023) GitHub/Bavla, Multiway network analysis in R. <https://github.com/bavla/ibm3m>
-  Batagelj, V, Cerinšek, M.: On bibliographic networks. *Scientometrics* 96 (2013), pages 845–864. [Springer](#)
-  Batagelj, V.: On fractional approach to the analysis of linked networks. *Scientometrics* 123 (2020), pages 621–633. [Springer](#)
-  Batagelj, V, Ferligoj, A, Doreian, P: Indirect Blockmodeling of 3-Way Networks. In: Brito, P, Cucumel, G, Bertrand, P, de Carvalho, F (eds) Selected Contributions in Data Analysis and Classification pp 151–159 [Springer](#), 2007; [preprint](#).
-  Batagelj, V, Zaveršnik, M (2011) Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, Volume 5, Number 2, 129–145.
-  Borgatti, SP, Everett, MG: Regular blockmodels of multiway, multimode matrices. *Social Networks* 14(1992)1-2, 91–120.



References II

Multiway
networks

V. Batagelj

- Cerinšek, M, Batagelj, V (2015) Generalized two-mode cores. *Social Networks* 42, 80–87.
- Coleman, J, Katz, E, Menzel, H (1957) The Diffusion of an Innovation Among Physicians. *Sociometry* 20:253–270.
- de Domenico, M (2023) Datasets Released for Reproducibility. <https://manliodedomenico.com/data.php>
- Everett, MG, Borgatti, SP: Chapter 9, Partitioning Multimode Networks. In: Doreian, P, Batagelj, V, Ferligoj, A (Eds) *Advances in Network Clustering and Blockmodeling*, Wiley, 2019.
- Genova, VG, Giordano, G, Ragozini, G, Vitale, MP (2022) Clustering student mobility data in 3-way networks. IFCS 2022, Porto. Slides.
- Kapferer B (1972) *Strategy and transaction in an African factory*. Manchester University Press.
- Krackhardt, D (1987). Cognitive social structures. *Social Networks*, 9, 104-134.
- Kroonenberg, PM: *Applied multi-way data analysis*. Wiley, 2008.



References III

- Lazega, E (2001) The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership. Oxford University Press.
- rgriffin (2018) 120 years of Olympic history: athletes and results; basic bio data on athletes and medal results from Athens 1896 to Rio 2016
<https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results>
- Schoonees, PC, Groenen, PJF, van de Velden, M: Least-squares bilinear clustering of three-way data. *Advances in Data Analysis and Classification* (2022) 16:1001–1037. [paper](#).
- Seidman, SB (1983).
Network structure and minimum degree.
Social Networks, 5:269–287.
- x3dom – instant 3D the HTML way! (2023) <https://www.x3dom.org/>
- view3dscene** a 3D viewer from Castle Game Engine (2023)
<https://castle-engine.io/view3dscene.php>