



Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

Analysis of weighted networks

2. Erasmus learning mobility flows between countries
2014-2024

Vladimir Batagelj

IMFM Ljubljana, UP IAM Koper, UL FMF Ljubljana

INSNA Sunbelt 2025 workshop

June 23-29, 2025, Paris, France



Outline

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

- 1 Erasmus flow network
- 2 Skeletons
- 3 Matrix representation
- 4 Blockmodeling
- 5 Conclusions
- 6 References



Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
Current version of slides (June 23, 2025 at 17:51): [slides PDF](#)
<https://github.com/bavla/Nets>



Erasmus

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

Erasmus+ is a European Union (EU) program designed to support education, training, youth, and sport across Europe. Established in 1987, it aims to provide opportunities for individuals to study, train, gain work experience, and volunteer abroad, while also fostering cooperation and innovation in these fields.

Key features of Erasmus+ are (1) mobility opportunities, (2) cooperation projects, (3) policy development, and (4) sport initiatives.

Erasmus+ is funded by the EU, with a budget of over €26 billion for the 2021-2027 period, making it one of the largest programs of its kind. It is open to EU member states, as well as non-EU countries associated with the program. Millions of individuals and thousands of organizations participate annually.

At the bottom of the Erasmus+ page [Data visualization on learning mobility projects](#), the “Learning mobility flows since 2014” chart can be found.

Erasmus interactive chart

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

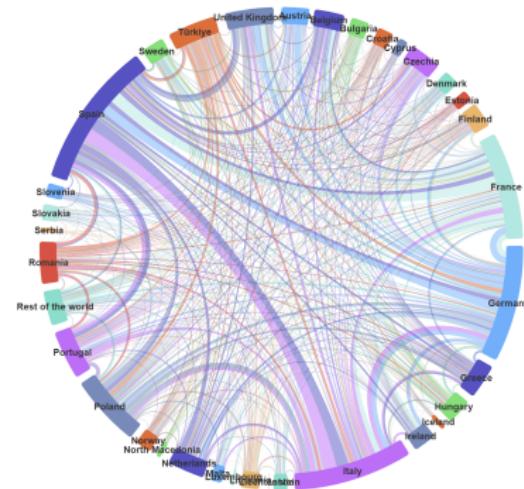
Matrix
representation

Blockmodeling

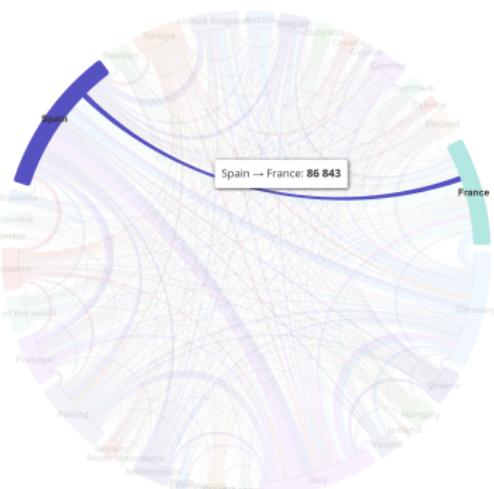
Conclusions

References

Learning mobility flows since 2014



Learning mobility flows since 2014



The interactive chart shows mobility flows between countries since 2014. The colors are related to the sending country.

Erasmus

... interactive chart

Weighted
networks 2

V. Batagelj

Erasmus flow
network

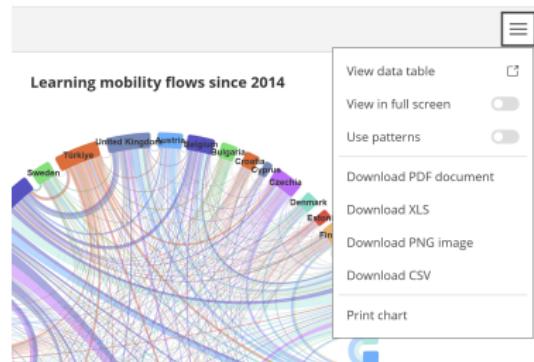
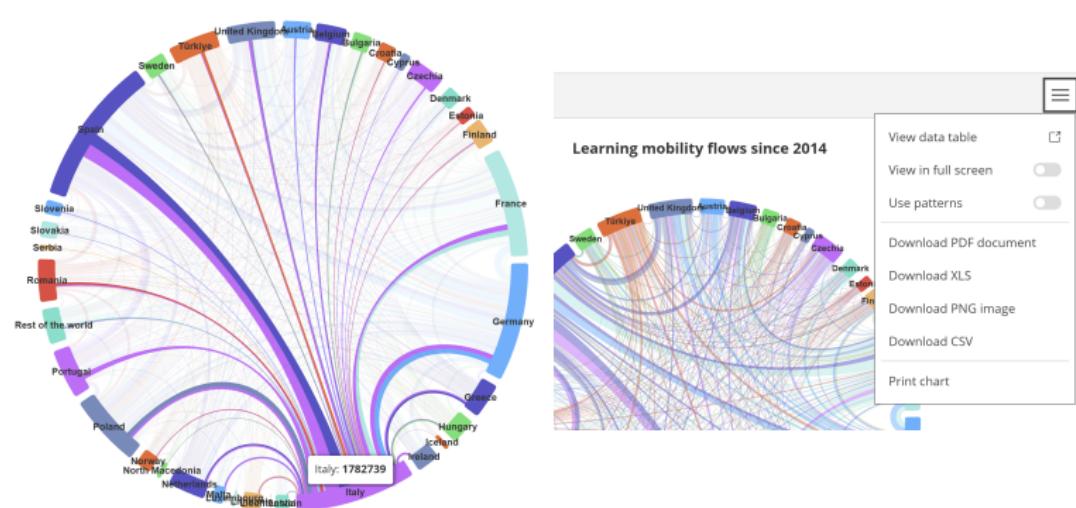
Skeletons

Matrix
representation

Blockmodeling

Conclusions

References



For example, moving the mouse over Italy will highlight all its in/outbound flows and the total count of participants. The same can be done at the flow level.

The interactive chart provides an option to download the network data.



Erasmus

creating network

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

I saved the network data on the file

Learning-mobility-flows-since-2014.csv. The dataset contains the following countries:

Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Liechtenstein, Lithuania, Luxembourg, Malta, Netherlands, North Macedonia, Norway, Poland, Portugal, Rest of the world, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Türkiye, United Kingdom.

I used the Deepseek to obtain the corresponding ISO 3166-1 alpha-2 country codes (ISO2.csv) and the total population estimate for each country (pop.csv).

I combined the collected data into an igraph network and saved it as ErasmusFlows.rds. The files are available at [GitHub/Vlado](#).



Erasmus

... creating network

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

```
> setwd("C:/Users/public/Sunbelt25")
> nWdir <- paste0("https://raw.githubusercontent.com/",
+ "bavla/Nets/refs/heads/master/netsWeight/")
> source(paste0(nWdir,"netsWeight.R"))
> F <- read.csv(file=paste0(nWdir,"data/Learning-mobility-flows-since-2000"))
> str(F)
'data.frame': 1223 obs. of 4 variables:
 $ Category : chr "highcharts-hv9y537-3" "highcharts-hv9y537-4" "highcharts-hv9y537-5" ...
 $ X..from. : chr "Austria" "Austria" "Austria" "Austria" ...
 $ X..to.   : chr "Austria" "Belgium" "Bulgaria" "Croatia" ...
 $ X..weight.: int 8394 6825 603 2389 780 3808 3962 1104 7014 15250 ...
> L <- data.frame(from=F$X..from.,to=F$X..to.,weight=F$X..weight.)
> str(L)
'data.frame': 1223 obs. of 3 variables:
 $ from : chr "Austria" "Austria" "Austria" "Austria" ...
 $ to   : chr "Austria" "Belgium" "Bulgaria" "Croatia" ...
 $ weight: int 8394 6825 603 2389 780 3808 3962 1104 7014 15250 ...
> names <- union(L$from,L$to)
> names
[1] "Austria"          "Belgium"           "Bulgaria"          "Croatia"          ...
[5] "Cyprus"            "Czechia"           "Denmark"           "Estonia"           ...
...
> C <- read.csv(file=paste0(nWdir,"data/ISO2.csv"),strip.white=TRUE)
> str(C)
'data.frame': 35 obs. of 2 variables:
 $ country: chr "Austria" "Belgium" "Bulgaria" "Croatia" ...
 $ ISO2   : chr "AT" "BE" "BG" "HR" ...
```



Erasmus

... creating network

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

```
> Pop <- read.csv(file=paste0(nWdir,"data/pop.csv"),strip.white=TRUE)
> str(Pop)
'data.frame': 35 obs. of 3 variables:
 $ n      : int 1 2 3 4 5 6 7 8 9 10 ...
 $ country: chr "Austria" "Belgium" "Bulgaria" "Croatia" ...
 $ pop    : num 9000000 11500000 6800000 4000000 1200000 ...
> V <- data.frame(name=C$country,IS02=C$IS02,pop=Pop$pop)
> N <- graph_from_data_frame(L,directed=TRUE,vertices=V)
> N$name <- "Erasmus learning mobility flows between countries 2014-2024"
> N$url <- "https://erasmus-plus.ec.europa.eu/resources-and-tools/factsheets"
> N$by <- "Vladimir Batagelj"
> N$cdate <- date()
> N
IGRAPH 51f535d DNW- 35 1223 -- Erasmus learning mobility flows between countries
+ attr: name (g/c), url (g/c), by (g/c), cdate (g/c), name (v/c), IS02 (v/c)
| pop (v/n), weight (e/n)
+ edges from 51f535d (vertex names):
 [1] Austria->Austria          Austria->Belgium          Austria->Bulgaria
+ ... omitted several edges
> saveRDS(N,file="ErasmusFlows.rds")
```



Network visualization

comments

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

- Larger, $n > 20$, dense graphs can't be presented readably with a graphical layout. For the Erasmus network, the number of nodes is $n = 35$, and the density $\gamma = 0.9984$. For dense graphs of moderate size (up to some hundreds of nodes) a better option is the matrix representation.
- What about weights? They can be represented by link thickness or level of grey of matrix cells. The problem is a very large range and the distribution of weights – most weights give almost white cells. For Erasmus $w_{\min} = 1$ and $w_{\max} = 217003$. Monotonic transformations such as $w' = a \cdot w$, $a > 0$ or $w' = \sqrt{w}$ or $w' = \log(w)$, etc. In our case, we used $w' = w^{0.1}$.



Erasmus network

monotonic transformations and weight distributions

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

```
> setwd("C:/Users/public/Sunbelt25")
> source("https://raw.githubusercontent.com/bavla/Rnet/master/R/Pajek.R")
> EF <- readRDS(file=url(paste0(nWdir,"data/ErasmusFlows.rds")))
> Z <- P <- as_adjacency_matrix(EF,attr="weight",type="both")
> n <- nrow(P); w <- sort(P[P>0])
> hist(w,col="green",border="black",breaks=50,
+       xlab="value",main="Erasmus Flow Value Distribution")
> hist(w**0.1,col="green",border="black",breaks=20,prob=TRUE,
+       xlab="value^0.1",main="Erasmus Flow Value^0.1 Distribution")
> lines(density(w**0.1,n=64),lwd=2,col="blue")
> m <- mean(w**0.1); s <- sd(w**0.1)
> m
[1] 2.183272
> s
[1] 0.3881439
> curve(dnorm(x,m,s),from=1,to=3.5,lwd=2,col="red",xaxt="n",yaxt="n",
+        add=TRUE)
> b <- rep(0,11); b[11] <- max(w); Co <- P
> for(i in 1:10) b[i] <- w[round((-1+2*i)*length(w)/20)]
> for(i in 1:n) for(j in 1:n)
+ {k <- 1; while(P[i,j]>b[k]) k <- k+1; Co[i,j] <- k}
> b
[1] 53 350 786 1327 2112 3178 4621 7323 13074 30142 217003
> hist(as.vector(Co),col="green",breaks=0:11)
```

Erasmus network

... monotonic transformations and weight distributions

Weighted networks 2

V. Batagelj

Erasmus flow
network

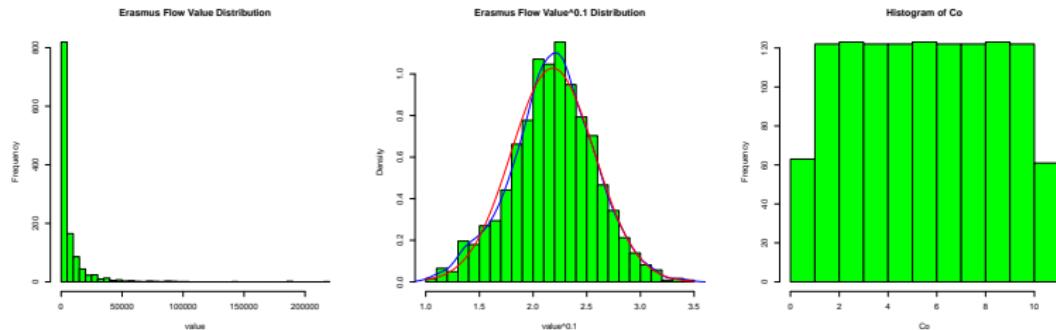
Skeletons

Matrix
representation

Blockmodeling

Conclusions

References



Monotonic (increasing) transformation $f : \mathbb{R} \rightarrow \mathbb{R}$

$$x < y \Rightarrow f(x) \leq f(y)$$

They preserve the ordering of weights.

Let $w' = f \circ w$ then $w(x) < w(y) \Rightarrow w'(x) \leq w'(y)$.



Skeletons

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

To get insight into the structure of a large (or/and) dense network, we can reduce it to its skeleton by removing less important links and/or nodes [Batagelj (2011)].

- Most often, the spanning tree, link cut, or node cut is used.
- In the closest k neighbors skeleton for each node, only the largest k incident links are preserved. Invariant for monotonic transformations.
- The Pathfinder algorithm was proposed in the 1980s by Schvaneveldt [Schvaneveldt et al.(1988), Schvaneveldt(1990), Batagelj et al.(2014)]. It removes from the network with a **dissimilarity** weight all links that do not satisfy the triangle inequality – if a shorter path exists that connects the link's end nodes, then the link is removed.
- Cores are a very efficient tool to determine the most cohesive (active) subnetworks [Batagelj and Zaveršnik(2011)]. For a node property p , the subset of nodes $\mathbf{C} \subseteq \mathcal{V}$ induces a p -core at level t if for all $v \in \mathbf{C}$ it holds $pc(v) \geq t$, and \mathbf{C} is the maximum such subset.

1 neighbor and 2 neighbors

... first and second choice

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

The package `netsWeight` contains the function

```
kNeighbors(Net,k,weight="weight",mode="out",
           strict=TRUE,loops=FALSE)
```

that determines the k -neighbors in a given network `Net`.

```
> One <- kNeighbors(N,1)
> V(One)$IS02 <- V(N)$IS02; w <- 1+log(E(CoreG)$weight)
> lo <- layout_with_dh(One)
> plot(One,layout=lo,edge.width=w,vertex.color="pink",
+       vertex.size=10,vertex.label=V(One)$IS02,vertex.label.cex=0.7)
> V(One)$x <- lo[,1]; V(One)$y <- lo[,2];
> saveRDS(One,file="EFone.rds")

> Two <- kNeighbors(N,2)
> V(Two)$IS02 <- V(N)$IS02
> lo <- layout_with_dh(Two)
> plot(Two,layout=lo,edge.width=w,vertex.color="pink",
+       vertex.size=10,vertex.label=V(Two)$IS02,vertex.label.cex=0.7)
> Pic <- tkplot(Two,800,800,layout=lo,edge.width=w,
+                  vertex.label=V(Two)$IS02,vertex.size=10,vertex.label.cex=0.7)
> # tkplot window is still active
> coor <- tk_coords(Pic,norm=FALSE)
> tk_close(Pic)
> V(Two)$x <- coor[,1]; V(Two)$y <- coor[,2]
> saveRDS(Two,file="EFtwo.rds")
```

1 neighbor and 2 neighbors
... first and second choice

Weighted networks 2

V. Batageli

Erasmus flow
network

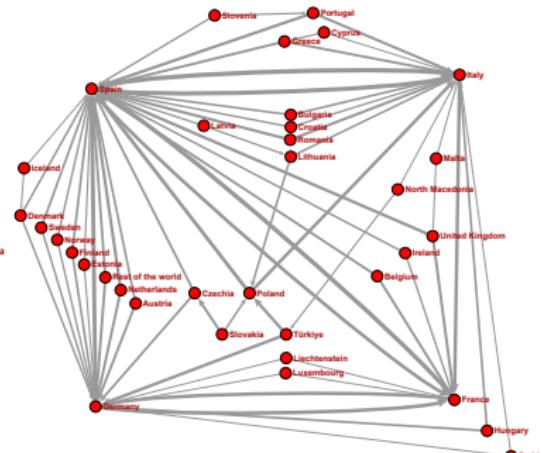
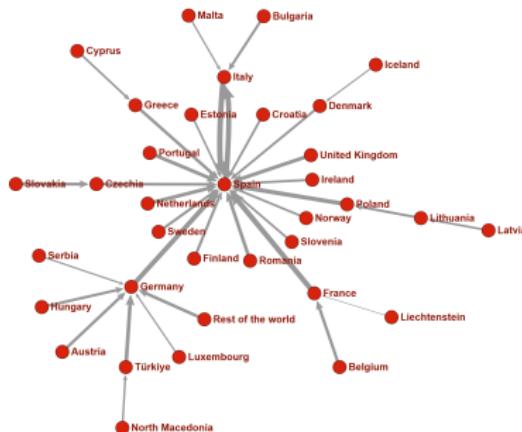
Skeletons

Matrix representation

Blockmodeling

Conclusions

References





Pathfinder skeleton

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

The *Pathfinder* algorithm was proposed in the 1980s by Schvaneveldt ([Schvaneveldt et al.(1988), Schvaneveldt(1990), Vavpetič et al.(2009)]) for simplifying weighted networks, where the weight measures a **dissimilarity** between nodes.

It is based on Minkowski operation $a \square b = \sqrt[r]{a^r + b^r}$. For $r = 1$, $r = 2$, and $r = \infty$ we get $a \square_1 b = a + b$, $a \square_2 b = \sqrt{a^2 + b^2}$, and $a \square_\infty b = \max(a, b)$.

For a path $\pi = (v_0, v_1, \dots, v_k)$ of length k we define its weight $w(\pi) = w(v_0, v_1) \square r w(v_1, v_2) \square r \dots \square r w(v_{k-1}, v_k)$.

The Pathfinder procedure removes from a given network \mathcal{N} every link (u, v) with its weight larger than the minimum weight of all u - v paths of length at most q . The resulting simplified network is denoted $\text{PFnet}(\mathcal{N}, r, q)$.

The Erasmus network weight w (number of visits) is a similarity measure. The Pathfinder procedure requires a dissimilarity measure d . A similarity w can be converted into a dissimilarity d in different ways. For example, $d_1 = w_{\max} - w$ or $d_2 = w_{\max}/w$. We will use the second option.



Pathfinder skeleton

Weighted networks 2

V. Batagelj

Erasmus flow network

Skeletons

Matrix representation

Blockmodeling

Conclusions

References

```
> PF2 <- PathFinderSim(P,r=Inf,q=Inf,s=2)
> PF2net <- graph_from_adjacency_matrix(PF2,mode="directed",
+   weighted=TRUE)
> V(PF2net)$ISO2 <- V(N)$ISO2; w <- (1+log(E(PF2net)$weight))/3
> lo <- layout_with_dh(PF2net)
> plot(PF2net,layout=lo,edge.width=w,vertex.color="pink",
+   vertex.size=12,vertex.label=V(PF2net)$ISO2,vertex.label.cex=0.7)
> Pic <- tkplot(PF2net,800,800,layout=lo,edge.width=w,
+   vertex.label=V(PF2net)$ISO2,vertex.size=12,vertex.label.cex=0.7)
> # tkplot window is still active
> coor <- tk_coords(Pic,norm=FALSE)
> tk_close(Pic)
> V(PF2net)$x <- coor[,1]; V(PF2net)$y <- coor[,2]
> PF2net$name <- "Erasmus learning mobility flows between countries 2014"
> PF2net$url <- "https://erasmus-plus.ec.europa.eu/resources-and-tools/f
> PF2net$subnet <- "Pathfinder skeleton r=2"
> PF2net$by <- "Vladimir Batagelj"
> PF2net$cdate <- date()
> saveRDS(PF2net,file="EFpf2.rds")
```

Erasmus flow Pathfinder skeleton

Weighted
networks 2

V. Batagelj

Erasmus flow
network

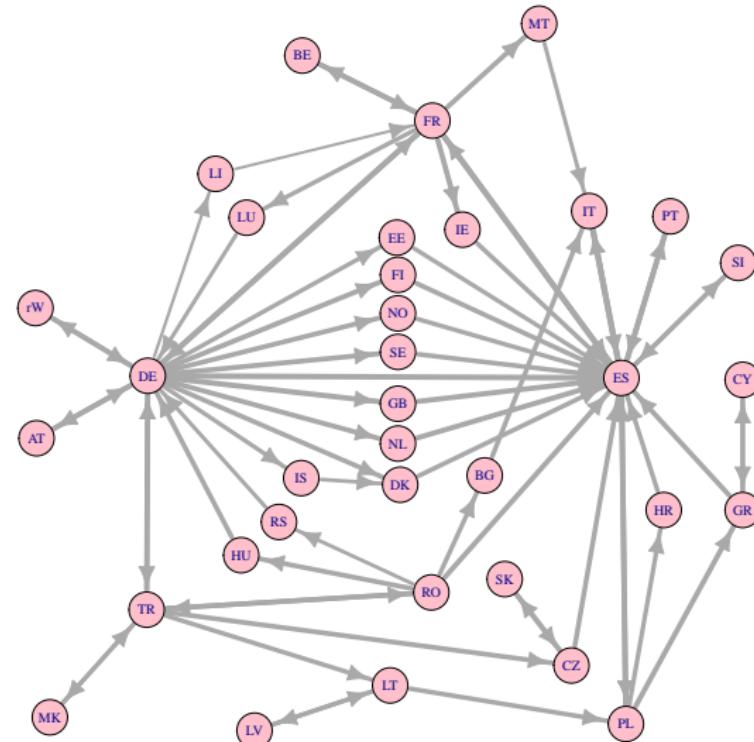
Skeletons

Matrix
representation

Blockmodeling

Conclusions

References





Observations

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

- ① The 1-neighbors skeleton highlights Spain as the main attractor in the network.
- ② The 2-neighbors skeleton shows the dominant role of Spain, Germany, France, and Italy.
- ③ The Pathfinder skeleton exposes Spain and Germany, and on the secondary level, France, Romania, Poland, and Türkiye.
- ④ These observations are confirmed by the Ps cores approach.



Generalized cores

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

The generalized core procedure can be used for any monotonic node property function $p_C(v)$

$$C_1 \subseteq C_2 \Rightarrow p_{C_1}(v) \leq p_{C_2}(v)$$

Currently, the package netsWeight provides the following properties
degree p_deg with weights=NULL

weighted degree p_wdeg

max on star links p_wmax

sum on neighbors p_nsum

max on neighbors p_nmax

neighbors diversity p_ncard

star links diversity p_lcard

all having the same parameters

(v,C,attr="deg",mode="all",loops=FALSE,weights="weight")



P_s -cores code

Weighted networks 2

V. Batagelj

Erasmus flow network

Skeletons

Matrix representation

Blockmodeling

Conclusions

References

```
> setwd("C:/Users/vlado/DL/data/erasmus/flows")
> EF <- readRDS("ErasmusFlows.rds")
> wid <- strength(EF, mode="in", weights=E(EF)$weight, loops=TRUE)
> wod <- strength(EF, mode="out", weights=E(EF)$weight, loops=TRUE)
> EFcoresA <- cores(EF, p=p_wdeg, mode="all", loops=TRUE)
> EFcoresI <- cores(EF, p=p_wdeg, mode="in", loops=TRUE)
> EFcoresO <- cores(EF, p=p_wdeg, mode="out", loops=TRUE)
> rep <- data.frame(ISO2=V(EF)$ISO2, wid=wid, wod=wod,
+   Acores=EFcoresA, Icores=EFcoresI, Ocores=EFcoresO)
> names(EFcoresA) <- V(EF)$ISO2
> top(EFcoresA, vcount(EF))

> Rnet <- "https://raw.githubusercontent.com/bavla/Rnet/master/R/"
> source(paste0(Rnet, "ClusNet.R"))
> oa <- order(EFcoresA, decreasing=TRUE)
> ca <- coreDendro(V(EF)$ISO2[oa], EFcoresA[oa])
> ha <- ca$height; ca$height <- max(unname(EFcoresA))-ca$value
> plot(ca, main="Erasmus mobility Ps cores / All", cex=0.8)
```

The function `coreDendro` in the package `netsWeight` provides dendrogram representation of the cores hierarchy.



P_s -cores

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

	> rep	ISO2	wid	wod	Acores	Icores	Ocores
Austria		AT	217234	188938	320354	175804	141526
Belgium		BE	240534	213200	345143	176038	159516
Bulgaria		BG	119420	169729	249391	103421	141731
Croatia		HR	119898	116218	207081	103421	98028
Cyprus		CY	84526	40886	123927	81600	40446
Czechia		CZ	260293	248912	340845	176038	159516
Denmark		DK	138116	119418	226716	125031	100006
Estonia		EE	77995	85311	158061	76830	80157
Finland		FI	219211	166128	320354	175804	136050
France		FR	627114	996627	652287	287693	364594
Germany		DE	791268	973914	652287	287693	364594
Greece		GR	274979	239679	376455	176038	174407
Hungary		HU	187500	208243	320354	159244	159516
Iceland		IS	43016	21817	64749	42888	21770
Ireland		IE	270104	83818	320123	200266	80157
Italy		IT	896081	886658	652287	287693	364594
Latvia		LV	86204	108230	181301	81938	96748
Liechtenstein		LI	2216	2412	4628	2216	2412
Lithuania		LT	122176	175171	242246	103421	136878
Luxembourg		LU	28702	13789	42407	28600	13761
Malta		MT	159215	24323	180545	143246	24158
Netherlands		NL	269472	305569	376455	176038	191225
North Macedonia		MK	34191	52852	84430	33208	50478
Norway		NO	137759	92329	218654	125031	86535
Poland		PL	468951	608085	498309	229822	294156
Portugal		PT	460831	293060	418598	229822	191225
Rest of the world	rW	209286	306823	376455	175804	198970	
Romania		RO	250745	388404	409279	176038	207249
Serbia		RS	27945	40593	68361	27942	40232
Slovakia		SK	103994	157337	239831	99455	136878
Slovenia		SI	106327	98170	189610	99187	88877
Spain		ES	1291788	918245	652287	287693	364594
Sweden		SE	196526	142343	300602	175804	120105
Türkiye		TR	254676	496051	409639	176038	248328
United Kingdom		GB	466488	261499	451608	274340	191225

P_s -cores

Weighted networks 2

V. Batagelj

Erasmus flow network

Skeletons

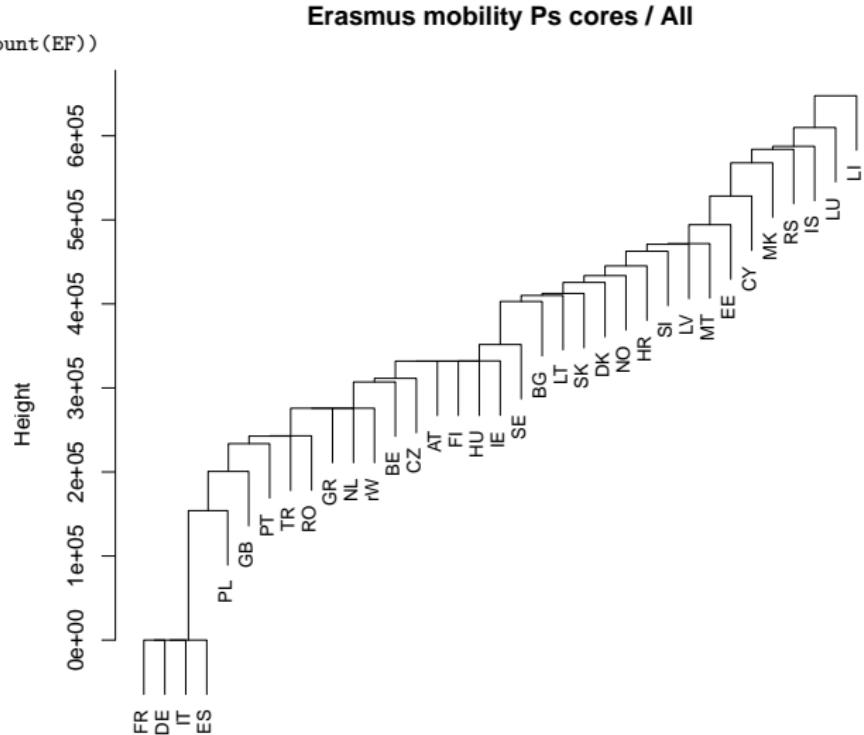
Matrix representation

Blockmodeling

Conclusions

References

```
> top(EFcoresA,vcount(EF))
   name    value
1  ES 652287
2  IT 652287
3  DE 652287
4  FR 652287
5  PL 498309
6  GB 451608
7  PT 418598
8  TR 409639
9  RO 409279
10 rW 376455
11 NL 376455
12 GR 376455
13 BE 345143
14 CZ 340845
15 HU 320354
16 FI 320354
17 AT 320354
18 IE 320123
19 SE 300602
20 BG 249391
21 LT 242246
22 SK 239831
23 DK 226716
24 NO 218654
25 HR 207081
26 SI 189610
27 LV 181301
28 MT 180545
29 EE 158061
30 CY 123927
31 MK 84430
32 RS 68361
33 IS 64749
34 LU 42407
35 LI 4628
```





Matrix representation

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

The `igraph` package uses sparse matrices based on the package `Matrix`. Some other used packages (for example `ClusNet`) are based on traditional matrices. A sparse matrix can be converted into a traditional matrix using the function `as_matrix`.

```
> library(gplots); library(Matrix)
> source("https://raw.githubusercontent.com/bavla/Rnet/
  refs/heads/master/R/ClusNet.R")
> EF <- readRDS(file=url(paste0(nWdir, "data/ErasmusFlows.rds")))
> P <- as.matrix(as_adjacency_matrix(EF, attr="weight", type="both"))
```

For visualization of matrices, we will use the function `heatmap.2` from the package `gplots`.



Matrix representation

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

The weights can be represented by the level of grey or the color of matrix cells. The problem of a very large range and the distribution of weights remains – most weights give almost white cells. Again, we use monotonic transformations.

```
> Co <- P; Co[P == 0] <- NA
> par(cex.main=1.2)
> heatmap.2(Co,Rowv=FALSE,Colv="Rowv",
+   dendrogram="none",scale="none",revC=TRUE,
+   margins=c(8,8),cexRow=0.8,cexCol=0.8,
+   col=colorpanel(30,low="grey95",high="black"),na.color="yellow",
+   trace="none", density.info="none", keyszie=0.8, symkey=FALSE,
+   main="Erasmus flows")
```

A better ordering of rows/cols in the matrix representation can be obtained by network clustering [[Batagelj et al.\(2014\)](#)].

Additional reordering of subtrees can be made manually using R by reordering nodes in the hierarchy.



Matrix representation of Erasmus network

Weighted
networks 2

V. Batagelj

Erasmus flow
network

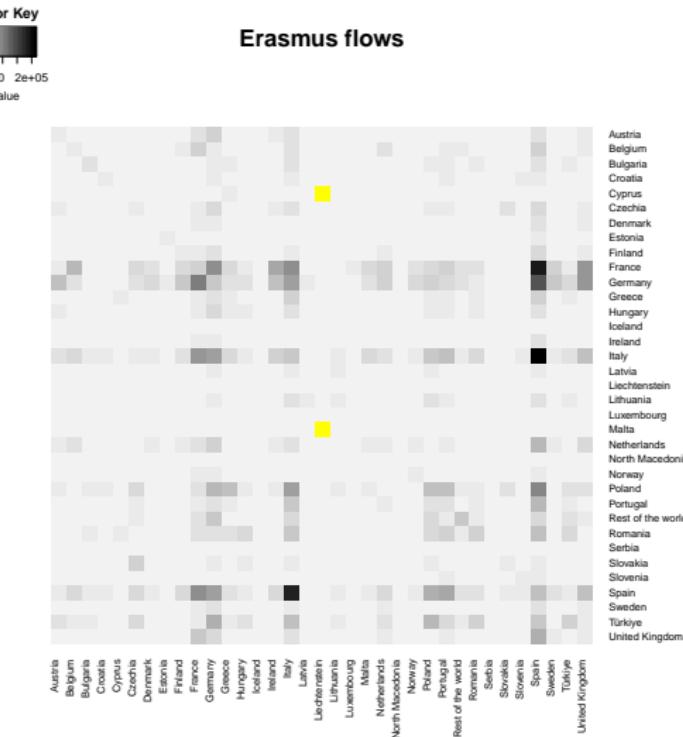
Skeletons

Matrix
representation

Blockmodeling

Conclusions

References





Matrix based (dis)similarities

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

For clustering units (nodes) we need a dissimilarity matrix D . In a square weight matrix, its weights can be sometimes considered (or transformed into) a dissimilarity.

$$D[u, v] = f(w[u, v], w[v, u]), \quad f(x, y) = f(y, x)$$

Often we use rows (and columns) as node descriptions and apply a selected dissimilarity on them

$$D[u, v] = d(w[u, .], w[v, .])$$

Typical dissimilarities are the *Euclidean distance*

$$d_e(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^2}$$

and the *Salton* or *cosine index*

$$S(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \bullet \mathbf{y}}{\sqrt{\mathbf{x}^2 \cdot \mathbf{y}^2}}, \quad d_s(\mathbf{x}, \mathbf{y}) = 1 - S(\mathbf{x}, \mathbf{y}) \text{ or } d_a(\mathbf{x}, \mathbf{y}) = \frac{\arccos S(\mathbf{x}, \mathbf{y})}{\pi}$$

where $\mathbf{x} \bullet \mathbf{y} = \sum_i x_i \cdot y_i$ and $\mathbf{x}^2 = \mathbf{x} \bullet \mathbf{x}$.



Corrected (dis)similarities

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

$$w[u, \cdot] = [w[u, 1], \dots, w[u, i], \dots, w[u, u], \dots, w[u, v], \dots, w[u, k]]$$

$$w[v, \cdot] = [w[v, 1], \dots, w[v, i], \dots, w[v, u], \dots, w[v, v], \dots, w[v, k]]$$

In traditional (dis)similarities, comparing $w[u, i]$ and $w[v, i]$ we are comparing how u relates to i with how v relates to i . The problem arises for $i = u$ and $i = v$. We would need to compare $w[u, u]$ with $w[v, v]$ and $w[u, v]$ with $w[v, u]$. This leads to **corrected** (dis)similarities.



Corrected Euclidean distance and Salton index

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

Corrected Euclidean distance

$$d'_e(u, v) = \sqrt{(w[u, v] - w[v, u])^2 + (w[u, u] - w[v, v])^2 + \sum_{t \notin \{u, v\}} (w[u, t] - w[v, t])^2}$$

Corrected Salton index of the link $(u, v) \in \mathcal{L}$

$$S'(u, v) = \frac{w[u, .] \bullet w[v, .] + (w[u, u] - w[u, v]) \cdot (w[v, v] - w[v, u])}{\sqrt{w[u, .]^2 \cdot w[v, .]^2}}$$

It also has the usual Salton index properties

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



ClusNet.R functions

Balassa, Salton, Euclid

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

```
Balassa <- function(P){  
  R <- rowSums(P); C <- rowSums(t(P)); T <- sum(R); Z <- P  
  for(u in 1:nrow(P)) for(v in 1:ncol(P)) Z[u,v] <- P[u,v]*T/R[u]/C[v]  
  Z <- log2(Z); Z[Z == -Inf] <- 0; return(Z)  
}  
  
CorSalton <- function(W){  
  S <- W; diag(S) <- 1; n = nrow(S)  
  for(u in 1:(n-1)) for(v in (u+1):n) S[v,u] <- S[u,v] <-  
    (as.vector(W[u,] %*% W[v,]) + (W[u,u]-W[v,v])*(W[v,v]-W[u,u]))/  
    sqrt(as.vector(W[u,] %*% W[u,])*as.vector(W[v,] %*% W[v,]))  
  return(S)  
}  
  
CorEuclid <- function(W){  
  D <- W; diag(D) <- 0; n = nrow(D)  
  for(u in 1:(n-1)) for(v in (u+1):n) D[v,u] <- D[u,v] <-  
    sqrt(sum((W[u,]-W[v,])**2) + 2*(W[u,u]-W[u,v])*(W[v,u]-W[v,v]))  
  return(D)  
}
```

ClusNet.R



Salton clustering

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

```
> P <- as.matrix(as adjacency_matrix(N,attr="weight",type="both"))
> n <- nrow(P); Z <- Co <- P; w <- sort(P[P>0])
> for(u in 1:n) for(v in 1:n) Z[u,v] <- P[u,v]**0.1
> t <- hclust(1-as.dist((CorSalton(Z)+CorSalton(t(Z)))/2),method="ward.D")
> t$merge <- flip(18,flip(7,flip(20,flip(30,flip(33,t$merge)))))
> t$merge <- flip(24,flip(28,flip(19,flip(4,t$merge))))
> b <- rep(0,11); b[11] <- max(w)
> for(i in 1:10) b[i] <- w[round((-1+2*i)*length(w)/20)]
> for(i in 1:n) for(j in 1:n)
+ {k <- 1; while(P[i,j]>b[k]) k <- k+1; Co[i,j] <- k}
> hist(as.vector(Co),col="green",breaks=0:11)
> Co[P==0] <- NA; par(cex.main=0.9)
> heatmap.2(Co,Rowv=as.dendrogram(t),Colv="Rowv",
+ dendrogram="column",scale="none",revC=TRUE,
+ margins=c(8,8),cexRow=0.8,cexCol=0.8,
+ col=colorpanel(30,low="grey95",high="black"),na.color="yellow",
+ trace="none", density.info="none", keyszie=0.8, symkey=FALSE,
+ main="Erasmus flows V^0.1 2014-2024 / Salton / Ward")
> Z <- Balassa(as.matrix(P))
> t <- hclust(as.dist((CorEu(Z)+CorEu(t(Z)))/2),method="ward.D")
> # pdf(file="EF14bala.pdf",width=30,height=30); par(cex.main=3)
> Z[P == 0] <- NA; par(cex.main=0.9)
> heatmap.2(Z,Rowv=as.dendrogram(t),Colv="Rowv",dendrogram="column",
+ scale="none",revC=TRUE,col=bluered(100),na.color="yellow",
+ trace="none", density.info="none", keyszie=0.8, symkey=FALSE,
+ # margins=c(15,15),cexRow=2,cexCol=2,
+ margins=c(8,8),cexRow=0.8,cexCol=0.8,
+ main="Erasmus flows 2014-2024 / Balassa / Ward")
> # dev.off()
```

Erasmus mobility flow matrix

Salton clustering

Weighted
networks 2

V. Batagelj

Erasmus flow
network

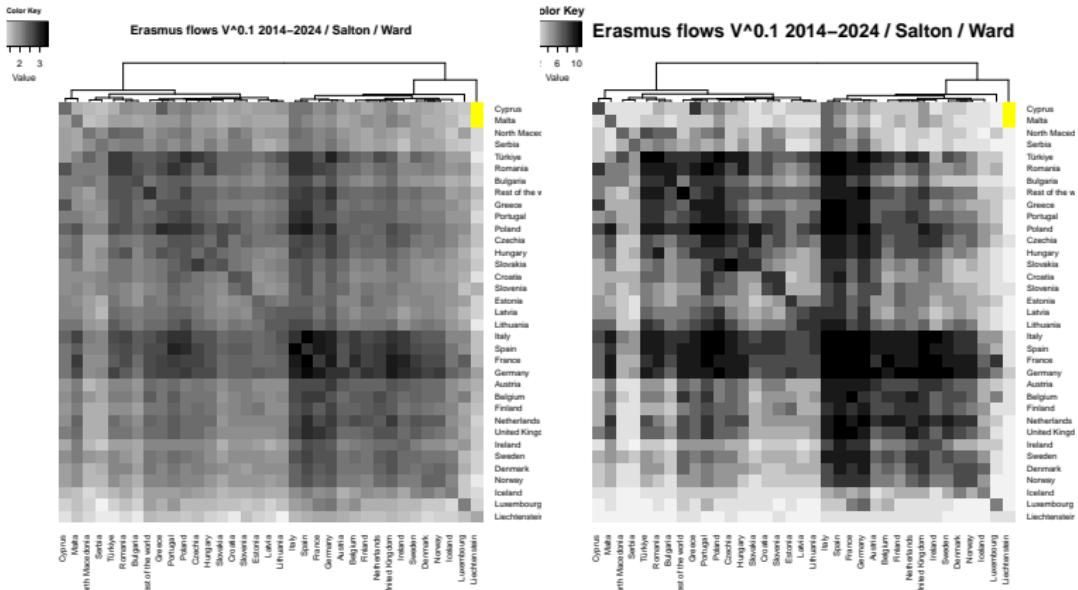
Skeletons

Matrix
representation

Blockmodeling

Conclusions

References





Observations

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

- ① The “cross” formed by $C_1 = (\text{Italy, Spain, France, Germany})$ – strong activity with almost all countries in both directions.
- ② Intense diagonal “squares” – clusters: $C_2 = (\text{Türkiye, Romania, Bulgaria, Rest of the world, Greece, Portugal, Poland, Czechia, Hungary})$, $C_3 = (\text{Poland, Czechia, Hungary, Slovakia})$, $C_4 = (\text{Croatia, Slovenia})$, $C_5 = (\text{Estonia, Latvia, Lithuania})$, $C_6 = (\text{Austria, Belgium, Finland, Netherlands, United Kingdom, Ireland, Sweden, Denmark, Norway})$, $C_1 \cup C_6$, $C_7 = (\text{Sweden, Denmark, Norway, Iceland})$
- ③ Out-diagonal “rectangles”: Luxembourg \times (France, Germany), Greece \times Cyprus, (Croatia, Slovenia) \times (North Macedonia, Serbia), etc.
- ④ In the cross, $C_4 \cup C_5 \cup \text{Slovakia}$ less often select France, etc.



ClusNet.R functions

reorder clustering

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

```
toFather <- function(tm){  
  n <- nrow(tm); T <- rep(0,2*n+1)  
  for(i in 1:n){  
    for(j in 1:2){  
      p <- tm[i,j]  
      if(p<0) T[-p] <- i+n+1 else T[n+1+p] <- i+n+1  
    }  
  }  
  return(T)  
}  
  
minCl <- function(u,v,T){  
  if(min(u,v)==0) return(T[max(u,v)])  
  # cat(u," ",v,":",T[u]," ",T[v],"\n")  
  if(u==v) return(u)  
  return( if(T[u]<T[v]) minCl(T[u],v,T) else minCl(u,T[v],T) )  
}  
  
flip <- function(k,T) {t <- T[k,1]; T[k,1] <- T[k,2];  
  T[k,2] <- t; return(T)}  
  
cl0rder <- function(M,k) if(k<0) return(-k) else  
  return(c(cl0rder(M,M[k,1]),cl0rder(M,M[k,2])))
```



Erasmus mobility flow clustering... reordering

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

```
> hm <- function(){par(cex.main=0.9)
+   heatmap.2(Z,Rowv=as.dendrogram(t),Colv="Rowv",
+   dendrogram="column",scale="none",revC=TRUE,
+   margins=c(8,8),cexRow=0.8,cexCol=0.8,
+   col=colorpanel(30,low="grey95",high="black"),na.color="yellow",
+   trace="none", density.info="none", keyszie=0.8, symkey=FALSE,
+   main="Erasmus flows V^0.1 2014-2024 / Salton / Ward")
+ }
> for(u in 1:n) for(v in 1:n) Z[u,v] <- P[u,v]**0.1
> t <- hclust(1-as.dist((CorSalton(Z)+CorSalton(t(Z)))/2),method="ward.D")
> Z[P == 0] <- NA
> s <- t; hm()
> F <- toFather(t$merge); N <- rownames(Z); n <- length(N)
> minCl(which(N=="Italy"),which(N=="Liechtenstein"),F) - n
[1] 33
> t$merge <- flip(33,t$merge); hm()
> minCl(which(N=="Italy"),which(N=="Luxembourg"),F) - n
[1] 30
> t$merge <- flip(30,t$merge); hm()
> minCl(which(N=="Iceland"),which(N=="Sweden"),F) - n
[1] 20
> t$merge <- flip(20,t$merge); hm()
.
.
.
> t <- s
> t$merge <- flip(18,flip(7,flip(20,flip(30,flip(33,t$merge)))))
> t$merge <- flip(24,flip(28,flip(19,flip(4,t$merge)))); hm()
```



Erasmus mobility flow matrix

Balassa clustering

Weighted
networks 2

V. Batagelj

Erasmus flow
network

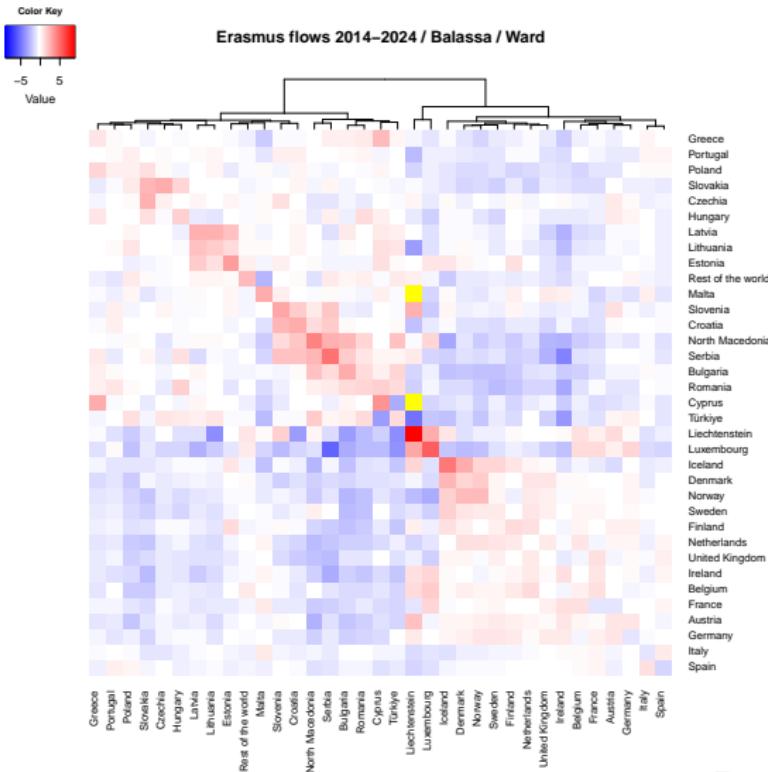
Skeletons

Matrix
representation

Blockmodeling

Conclusions

References





Observations

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

- ① Three main clusters $B_l = (\text{Greece} : \text{Türkiye})$ – less developed, $B_h = (\text{Iceland} : \text{Spain})$ – high developed. and $B_L = (\text{Liechtenstein}, \text{Luxembourg})$. Most cells inside squares are red and out-diagonal rectangles are mostly blue – exchange between countries from the same cluster is above expected, and below expected between different clusters.
- ② Red diagonal “squares” – clusters: $B_1 = (\text{Slovakia}, \text{Czechia}, \text{Hungary})$, $B_2 = (\text{Latvia}, \text{Lithuania}, \text{Estonia})$, $B_3 = (\text{Slovenia}, \text{Croatia}, \text{North Macedonia}, \text{Serbia})$, $B_4 = (\text{North Macedonia}, \text{Serbia}, \text{Bulgaria}, \text{Romania})$, B_L . The exchange between Cyprus and Türkiye is bellow expected. In the main cluster B_h we can identify a subcluster $B_5 = (\text{Iceland}, \text{Denmark}, \text{Norway}, \text{Sweden}, \text{Finland}, \text{Netherlands}, \text{United Kingdom})$. Within the clusters B_1 , B_2 , B_3 , B_4 , B_L , and ($\text{Iceland}, \text{Denmark}, \text{Norway}$) visits are much above expected.



Observations

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

- ③ Countries from the cluster B_I selected Malta below the expected times. The exchange between B_5 and B_L is below the expected.
- ④ Exchange between Cyprus and Greece is above expected.
- ⑤ Exchange of Italy, Spain, and Estonia with other countries is mostly close to as expected.

Erasmus mobility flow blockmodel

Balassa clustering

Weighted
networks 2

V. Batagelj

Erasmus flow
network

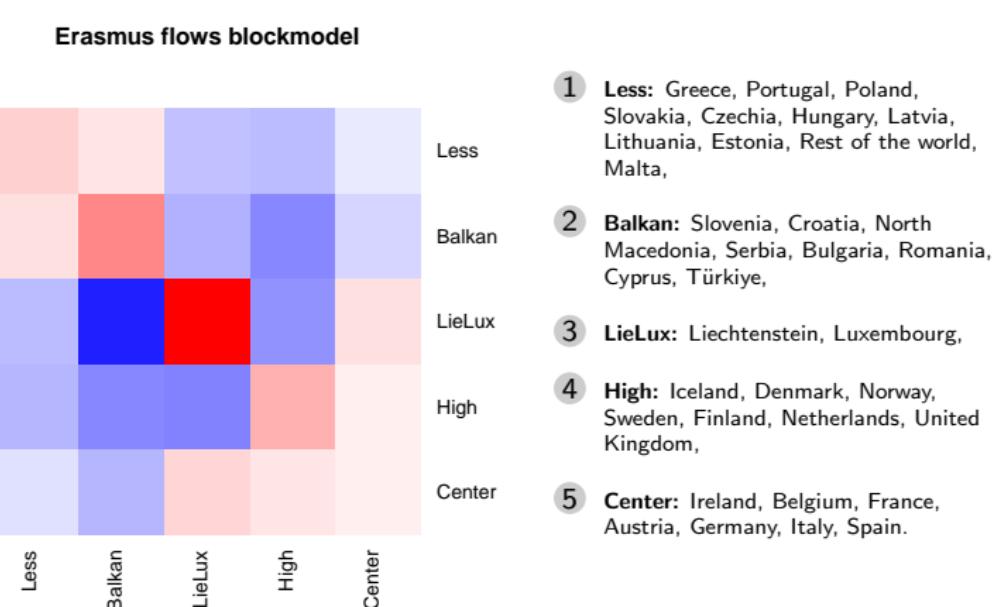
Skeletons

Matrix
representation

Blockmodeling

Conclusions

References



Blockmodeling

Weighted networks 2

V. Batageli

Erasmus flow
network

Skeletons

Matrix representation

Blockmodeling

Conclusions

References

```

> BlockModel <- function(P,q,p,lab){
+   m <- length(table(p))
+   B <- matrix(0,nrow=m,ncol=m); rownames(B) <- colnames(B) <- lab
+   for(i in 1:m){I <- q[p==i]
+     for(j in 1:m){J <- q[p==j]; B[i,j] <- sum(P[I,J])} }
+   return(B)
+ }

> q <- clOrder(t$merge,34)
> p <- c(rep(1,11),rep(2,8),rep(3,2),rep(4,7),rep(5,7))
> lab <- c("Less","Balkan","LieLux","High","Center")
> B <- BlockModel(P,q,p,lab)
> BB <- Balassa(B)
> BC <- apply(BB,1:2,function(x) min(2.5,max(-2.5,x)))
> heatmap.2(BC,Rowv=FALSE,Colv="Rowv",
+   dendrogram="none",scale="none",revC=FALSE,
+   margins=c(8,8),cexRow=1.5,cexCol=1.5,
+   col=bluered(100),na.color="yellow",
+   trace="none", density.info="none", keyszie=0.8, symkey=FALSE,
+   main=paste("Erasmus flows blockmodel",sep=""))
> t$labels[q[p==1]]
[1] "Greece"          "Portugal"         "Poland"           "Slovakia"
[5] "Czechia"          "Hungary"          "Latvia"           "Lithuania"
[9] "Estonia"          "Rest of the world" "Malta"
> t$labels[q[p==2]]
[1] "Slovenia"         "Croatia"          "North Macedonia" "Serbia"
[6] "Romania"          "Cyprus"           "Türkiye"

```



Conclusions

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

- ① Why is Spain the most attractive country?
- ② How can the blue between less and high developed countries be reduced?
- ③ This is exploratory network analysis. Collect and use additional data (neighbors relation, population size, GDP, etc.).
- ④ Temporal version of the network.
- ⑤ World trade / Compstat 2023



Acknowledgments

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

The computational work reported in this presentation was performed using R. The code and data are available at [GitHub/Vlado](#).

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



References |

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References

- Batagelj, V, Zaveršnik, M: Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, 2011. Volume 5, Number 2, 129–145.
- Batagelj, V, Doreian, P, Ferligoj, A, Kejžar, N (2014). *Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching, Visualization and Network Evolution*. Wiley. ISBN: 978-0-470-71452-2
- Batagelj, V (2011). *Large-Scale Network Analysis*. in John Scott, Peter J. Carrington eds. *The SAGE Handbook of Social Network Analysis* SAGE Publications.
- 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.
- Schvaneveldt, RW (Editor) (1990). *Pathfinder associative networks: Studies in knowledge organization*. Norwood, NJ: Ablex. [book](#)
- Schvaneveldt, RW, Dearholt, DW, Durso, FT (1988). Graph theoretic foundations of Pathfinder networks. *Computers and Mathematics with Applications*, 15, 337–345.



References II

Weighted
networks 2

V. Batagelj

Erasmus flow
network

Skeletons

Matrix
representation

Blockmodeling

Conclusions

References



Vavpetič, A, Batagelj, V, Podpečan, V (2009). "An implementation of the Pathfinder algorithm for sparse networks and its application on text network." In 12th International Multiconference Information Society, vol. A, pp. 236-239.