Temporal density of complex networks and ego-community dynamics

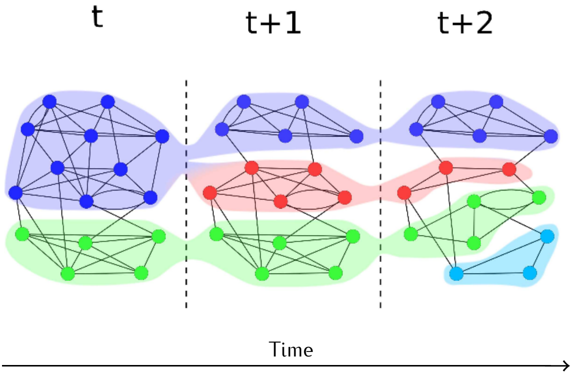
Sergey Kirgizov Éric Leclercq

UBFC — Université Bourgogne Franche-Comté, Le2i, UMR CNRS 6306

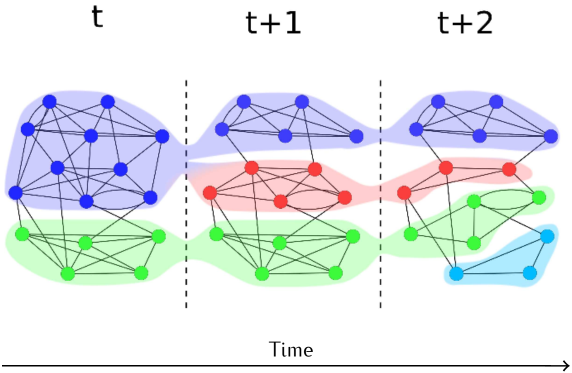
19th September 2016
Conference on Complex Systems
Amsterdam

1. Classical community dynamics
2. Our method
3. Illustrations of our method
    - School contacts network
    - Twitter temporal network

Snapshots based detection/visualisation.



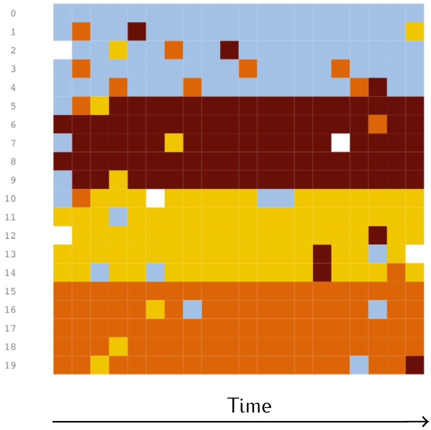
Snapshots based detection/visualisation.



This visualisation is not suitable for large graphs or large number snapshots

Another method of visualisation

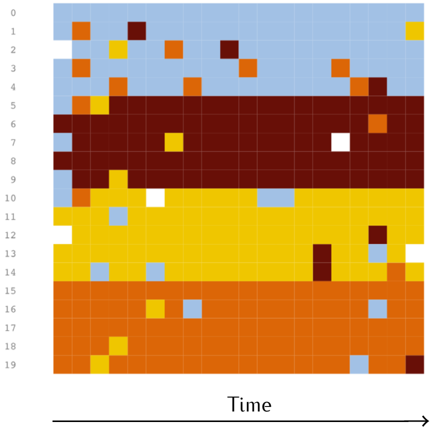
Lines corresponds to nodes, colors to different communities.



A visualisation from **Intrinsically Dynamic Network Communities** by Mitra, Tabourier, Roth 2011[3]

Another method of visualisation

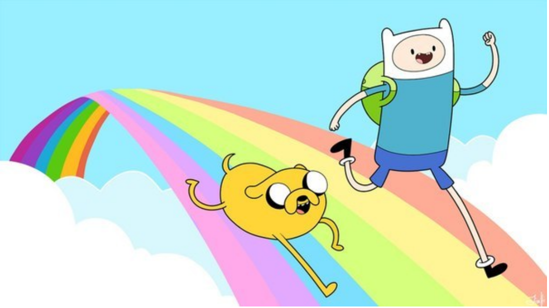
Lines corresponds to nodes, colors to different communities.



Usually, real networks changes more smoothly...

1. Classical community dynamics
2. Our method
3. Illustrations of our method
    3.1 School contacts network
    3.2 Twitter temporal network

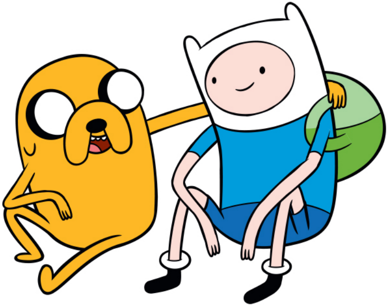
The ultimate goal is hard



Detect communities in dynamic networks and visualise their evolution!

The ultimate goal is hard

It is not so easy to detect communities in dynamic networks and visualise their evolution



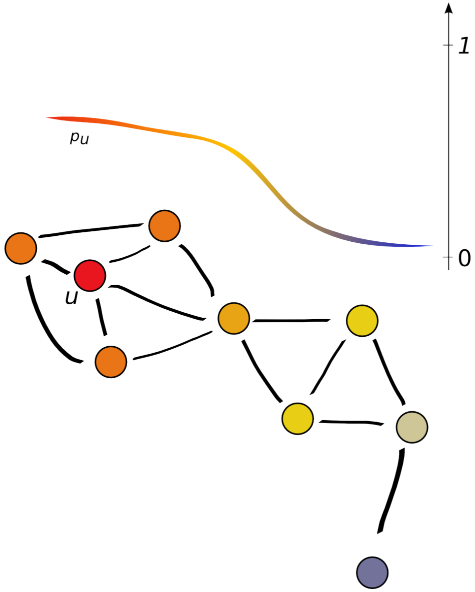
Instead we consider ego-community evolution.

Ego-community structure

Node u is a centre of the ego-community.

Function $p_u : V \to [0, 1]$ assigns a probability for a node v be in the community of node u.

Red color means large probability, blue means small.



Algorithms

Many algorithms can provide such community structure

1. Eigenvector centrality
2. **Personalised pagerank**
3. Shi–Malik's normalised relaxed mincut
4. Pons–Latapy's Walktrap
5. Danisch–Guillaume–Le Grand's Carryover opinion
6. Heat propagation based methods
7. Kleinberg's HITS
8. …

Complex networks from real world

Personalised pagerank can find good communities (even if they overlap) in real-world networks (DBLP, Youtube, Amazon)

Community membership identification from small seed sets

Kloumann et Kleinberg, SIGKDD, 2014

Overlapping Community Detection Using Neighborhood-Inflated Seed Expansion

Whang, Gleich, Dhillon, 2015

Personalised pagerank

A, D : adjacency matrix, degree matrix

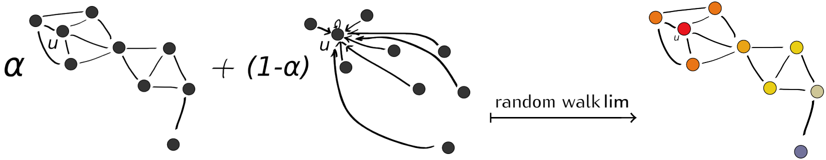
$M = (D^{-1}A)^T$: transition matrix

x_0 : initial probability distribution over nodes (seed selection)

α : teleportation parameter

$\Upsilon = x_0 \mathbf{1}^T$: teleportation matrix

$\widehat{M} = \alpha M + (1-\alpha)\Upsilon$: google matrix



Personalised pagerank is a vector r satisfies $r = \widehat{M} r$

The anatomy of a large-scale hypertextual Web search engine

Brin and Page, 1998

By pagerank we find instantaneous ego-communities

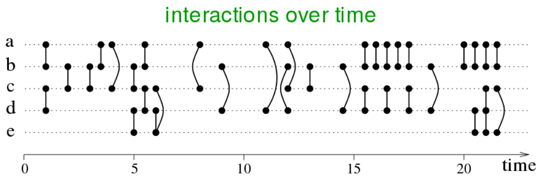
(or ego-communities in static networks)

What about the dynamics?

Discrete dynamic graph models

Dynamic graphs

- Snapshots [Hopcroft et al., 2004, Leskovec et al., 2005],
- Time-varying graphs [Casteigts et al., 2012, Wehmuth et al., 2013]
- Link Streams [Viard, Latapy, and Magnien, 2016] **illustrated below**



Changes in these models are discrete.

Our **“Temporal Density”** approach

1. Smooth the discrete input data (link stream)
2. Cut smoothed data into timeslices
3. Perform pagerank for all timeslices
4. Visualise

Typical dynamic dataset: stream of links

A node interacts with another node at time t.

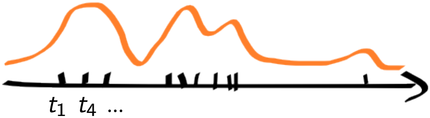
a	b	t_1
c	b	t_2
d	c	t_3
a	b	t_4
d	b	t_5
	...	

Smooth the discrete input data

Link stream between a and b

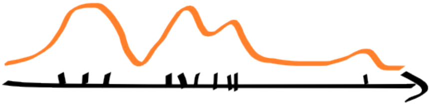
a	b	t_1
c	b	t_2
d	c	t_3
a	b	t_4
d	b	t_5
	...	

transforms to



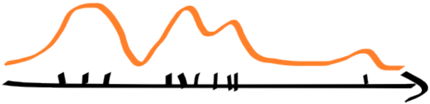
(by Parzen–Rosenblatt window method)

Smooth the discrete input data

Time-density between *a* and *b* 

Smooth the discrete input data

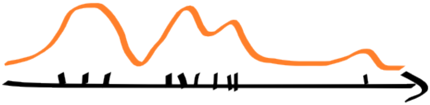
Time-density between *a* and *b*



Time-density between *c* and *b*

Smooth the discrete input data

Time-density between *a* and *b*



Time-density between *c* and *b*

Time-density between *a* and *c*

...

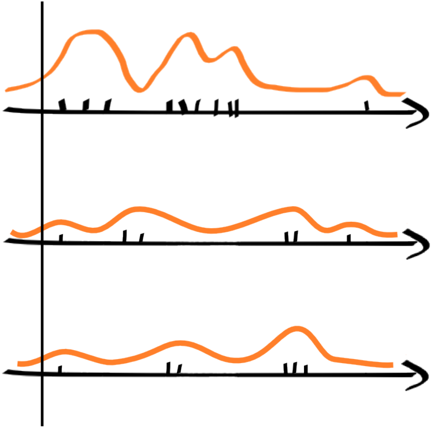
Cut & pagerank

Time-density between a and b

Time-density between c and b

Time-density between a and c

...



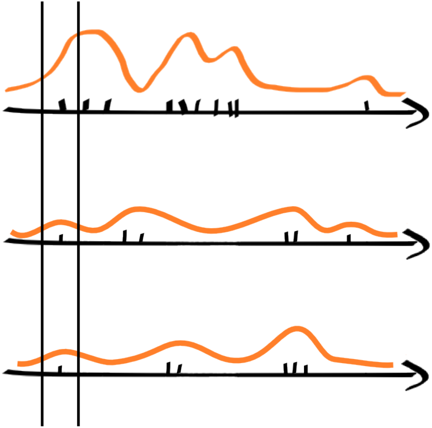
Cut & pagerank

Time-density between *a* and *b*

Time-density between *c* and *b*

Time-density between *a* and *c*

...



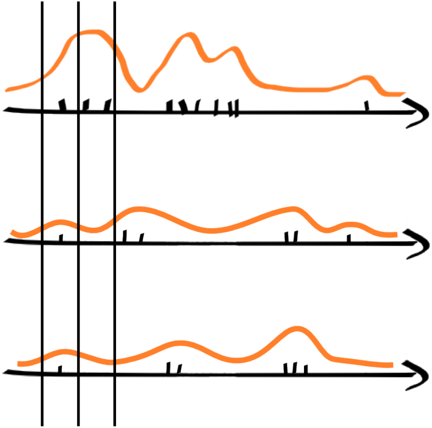
Cut & pagerank

Time-density between a and b

Time-density between c and b

Time-density between a and c

...



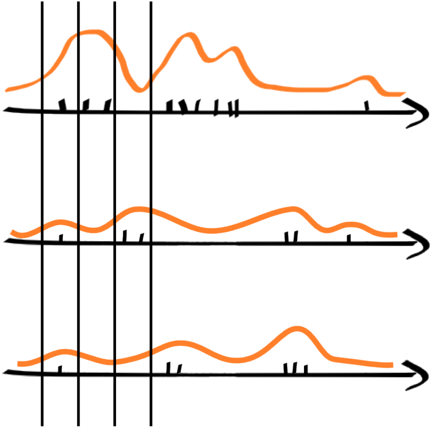
Cut & pagerank

Time-density between a and b

Time-density between c and b

Time-density between a and c

...



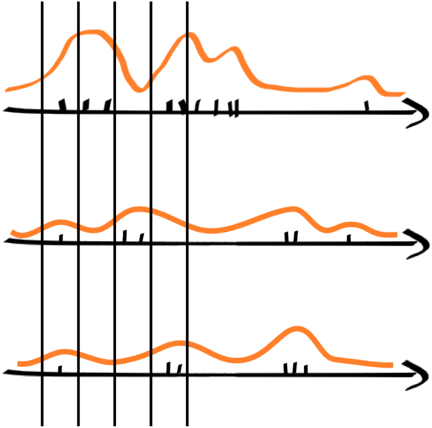
Cut & pagerank

Time-density between a and b

Time-density between c and b

Time-density between a and c

...



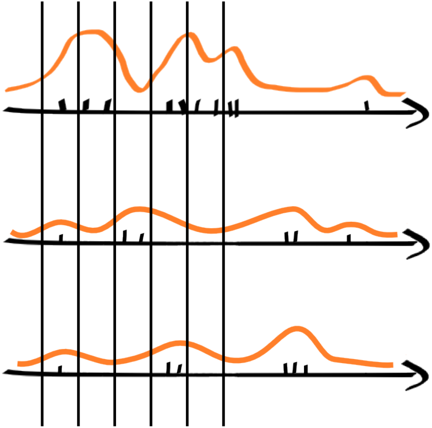
Cut & pagerank

Time-density between a and b

Time-density between c and b

Time-density between a and c

...



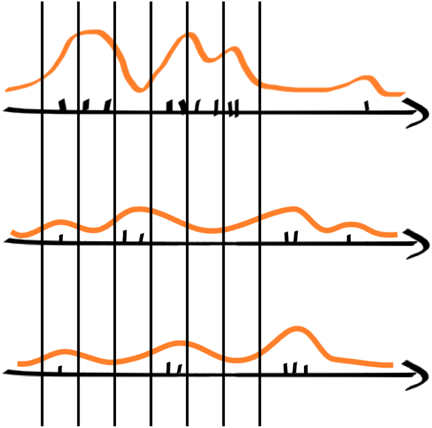
Cut & pagerank

Time-density between a and b

Time-density between c and b

Time-density between a and c

...



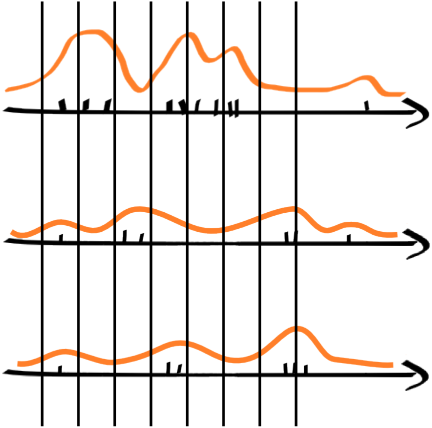
Cut & pagerank

Time-density between a and b

Time-density between c and b

Time-density between a and c

...



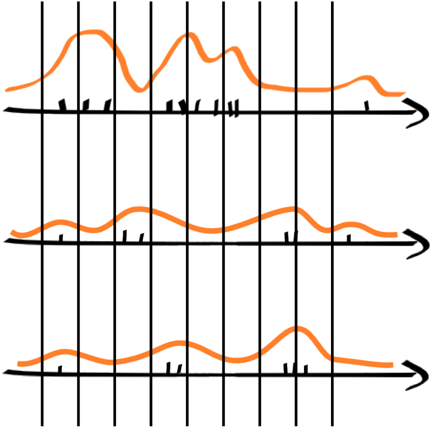
Cut & pagerank

Time-density between a and b

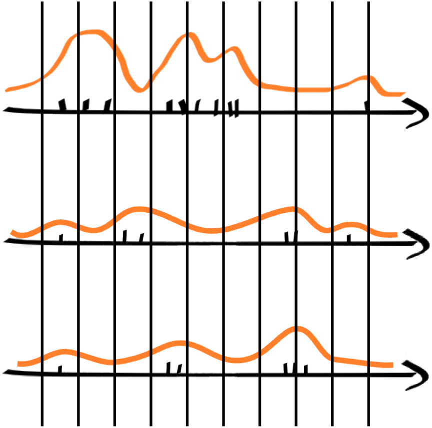
Time-density between c and b

Time-density between a and c

...



Cut & pagerank



Time-density between *a* and *b*

Time-density between *c* and *b*

Time-density between *a* and *c*

...

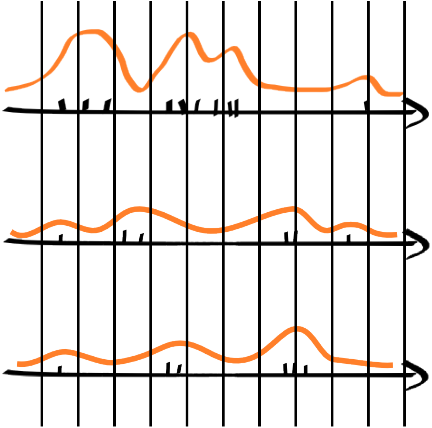
Cut & pagerank

Time-density between *a* and *b*

Time-density between *c* and *b*

Time-density between *a* and *c*

...



1. **Cut the smoothed data into timeslices**
2. **For every timeslice perform a personalised pagerank using smoothed values as link weights.**

Visualisation of ego-community dynamics

$p_{u,v}(t)$: "probability" that a node v is in the ego-community of node u at time t.

$p_{u,v}(t) =$ 

Proposition

Adjacency matrix $A(t)$ is smooth $\Rightarrow p_{u,v}(t)$ is smooth.

Schema of the proof

Pagerank : $A(t) \mapsto \widehat{M}(t) \mapsto p_{u,v}(t)$
$\widehat{M}$ is necessarily irreducible and aperiodic.
Use Theorem 3.2 from **A Note on Perturbations of Stochastic Matrices** by Huppert et Willems, 2000

Primary school contact network

ABOUT | GALLERY | PUBLICATIONS

DATASET: Primary school temporal network data

Release data: Sep 30, 2015

This data set contains the temporal network of contacts between the children and teachers used in the study published in BMC Infectious Diseases 2014, 14:695. The file contains a tab-separated list representing the active contacts during 20-second intervals of the data collection. Each line has the form “t i j Ci Cj”, where i and j are the anonymous IDs of the persons in contact, Ci and Cj are their classes, and the interval during which this contact was active is [t – 20s, t]. If multiple contacts are active in a given interval, you will see multiple lines starting with the same value of t. Time is measured in seconds.

Terms and conditions

The data are distributed to the public under a Creative Commons Attribution-NonCommercial-ShareAlike license. When this data is used in published research or for visualization purposes, please cite the following papers:

242 nodes, 125 773 links

Community dynamics visualisation

***u* is a fixed student from class “4A”, the centre of ego-community**

v and w are others students

$$p_{u,v}(t) =$$
$$p_{u,w}(t) =$$
$$\dots \quad \dots$$

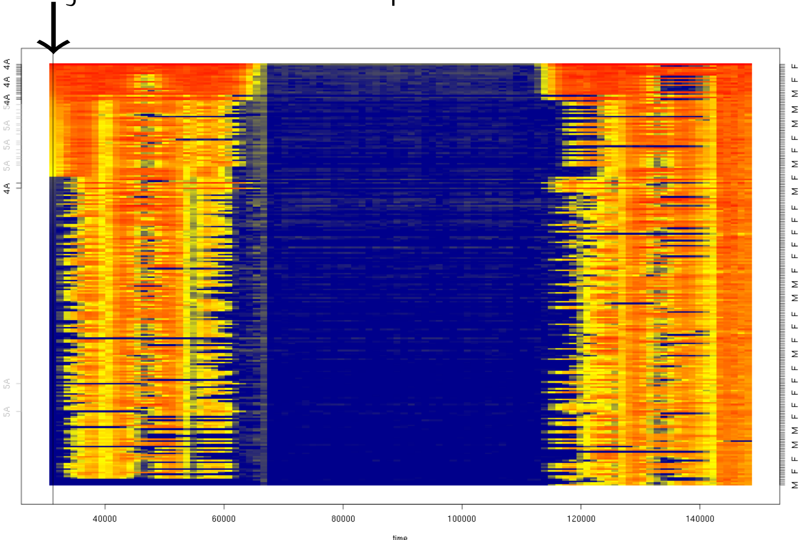
We sort these lines by values at the timestamp of interest.

Community dynamics visualisation

Lines correspond to students.

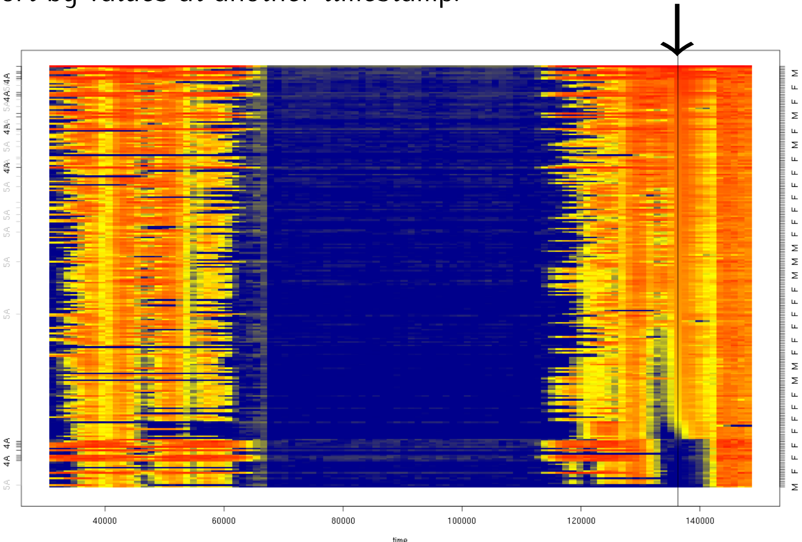
Columns are instantaneous ego-centred community structures $p_{u,t}(v)$

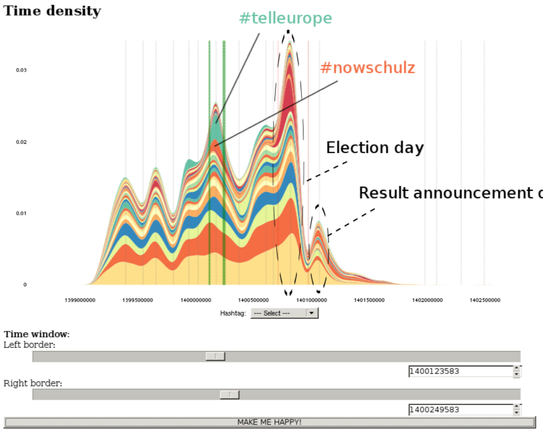
Sort by values at first timestamp.



Community dynamics visualisation

Lines correspond to students.
Columns are instantaneous ego-centred community structures $p_{u,t}(v)$
Sort by values at another timestamp.





Top 10 hashtags

	count	hashtag
1	570	ep2014
2	443	telleurope
3	438	ee2014
4	256	notreeurope
5	237	europeennes2014
6	200	eelv
7	190	nowschulz
8	108	europe
9	91	tafta
10	85	ump

Top 10 users

	count	from_user_name	from_user
1	121	JS Herpin	@jsherpin
2	108	Elodie Massé	@masselodie
3	81	Yannick Jadot	@yjadot
4	79	Béatrice DELGENDRE	@bdelgendre
5	74	Guillaume Balas	@BalasGuillaume
6	53	Raquel Garrido	@RaquelGarridoPG
7	47	Aleksander GLOGOWSKI	@Aleks_Paris
8	44	Sandrine Bélier	@sandrinebelier
9	39	Dolorès BAUDELOT	@Yodado
10	38	Franck Proust	@franckproust

Towards a Twitter Observatory: A multi-paradigm framework for collecting, storing and analysing tweets

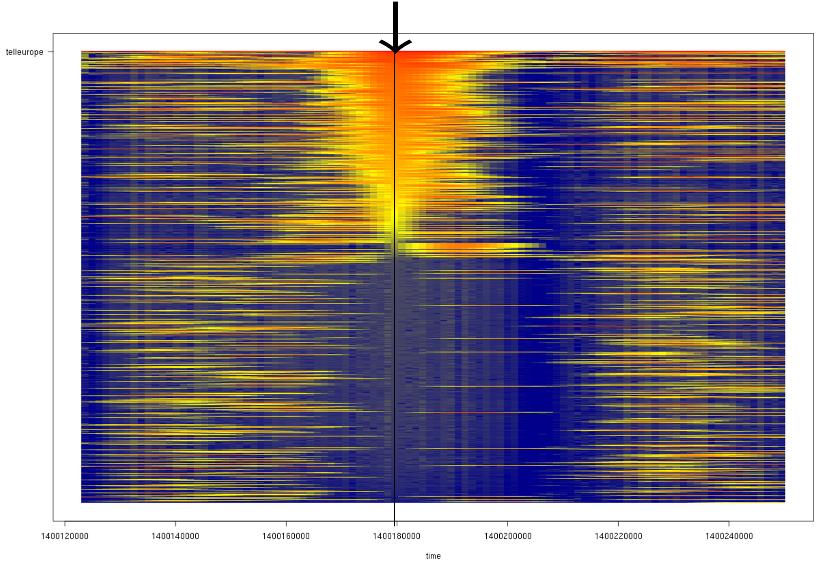
Ian Basaille, Kirgizov, Éric Leclercq, Marinette Savonnet, and Nadine Cullot

RCIS 2016

European Elections 2014 in Twitter

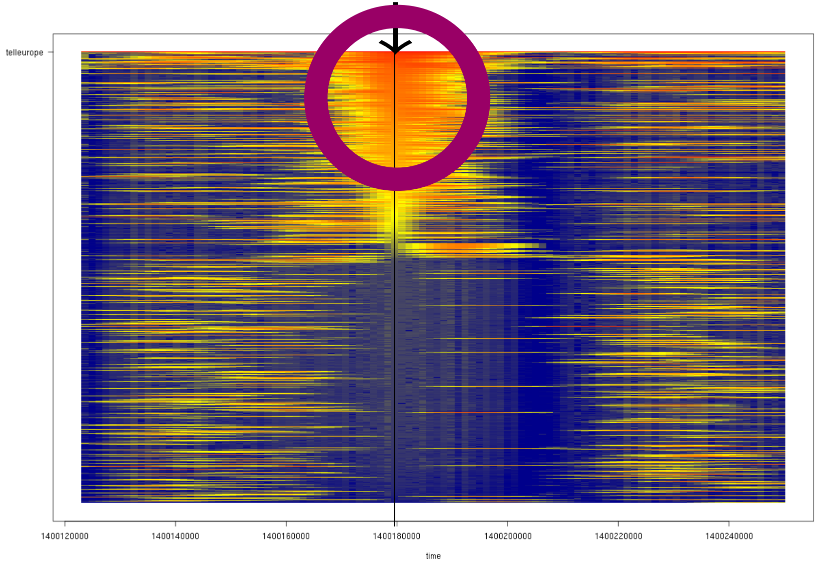
Dynamic community structure around **#telleurope**

Sort by values at this timestamp.

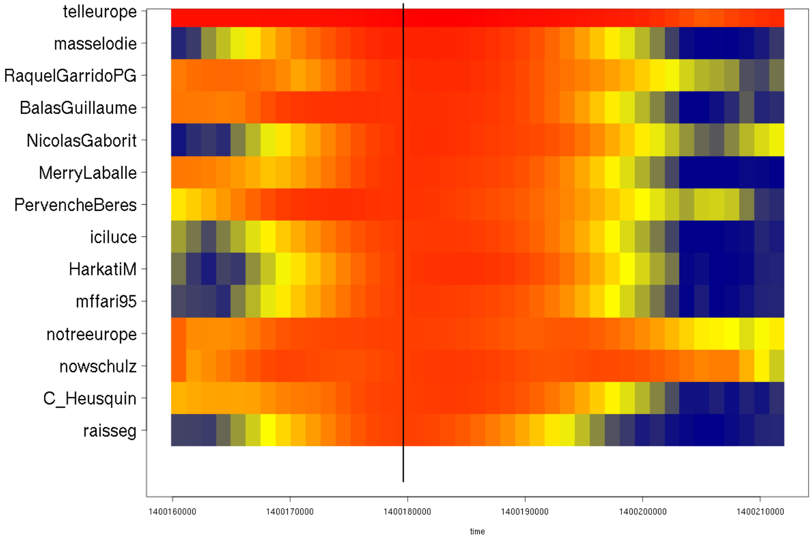


European Elections 2014 in Twitter

Dynamic community structure around **#telleurope**
Sort by values at this timestamp.



Dynamic community structure around **#telleurope**



Conclusion

Temporal density approach for

- ♣ Visualisation/study of community evolution
- ♠ Event detection and description

Future

- ♣ On-line data processing
- ♦ Better sorting (MDS, correlation based, etc)

Many thanks for your attention!

https://github.com/kerzol/ego-evolution

http://kirgizov.link

http://eric-leclercq.fr

Bonus slides!

Complexity

k : number of timeslices
n : number of nodes
m : total number of links
μ : maximal number of links between two nodes

Complexity

1. Apply kernel method to smooth all link-presence functions (Binned FFT), make k timeslices — $O\big(m(\mu + k \log k)\big)$
2. Perform pagerank for all timeslices $\approx k \cdot O(m \log m)$

in total... $\approx O\big(km \log(km) + m\mu\big)$

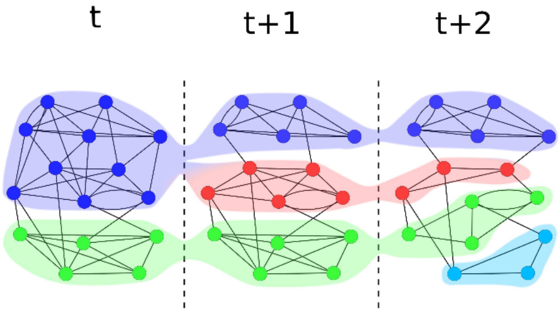
Fast computation of kernel estimators

Raykar, Duraiswami, et Zhao, 2010

Using pagerank to locally partition a graph

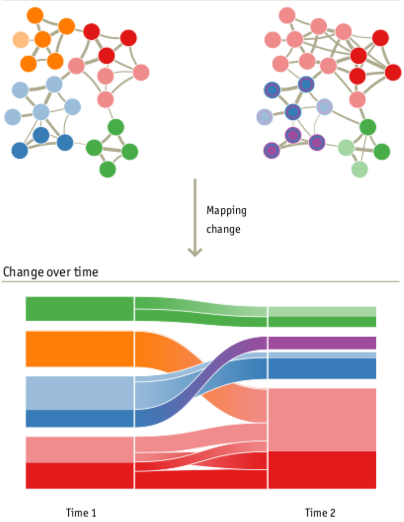
Andersen, Chung, Lang, 2007

Snapshot-based community dynamics

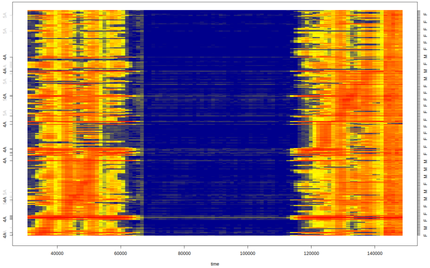


Snapshot-based community dynamics

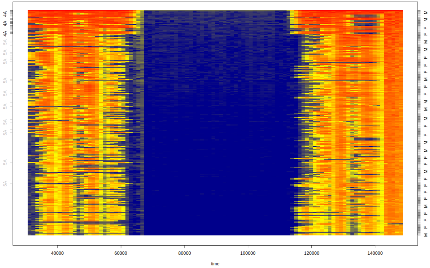
Mapping change in large networks
Rosvall, Bergstrom, 2010



Better sorting ?



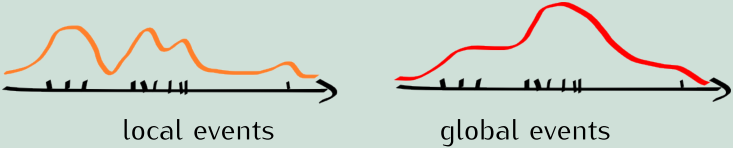
Sort by timestamp of maxima of every line



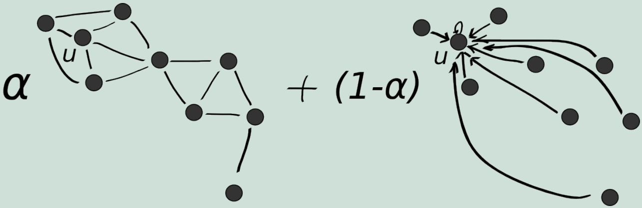
Sort by sum

Parameters

Smoothing degree, kernel bandwidth

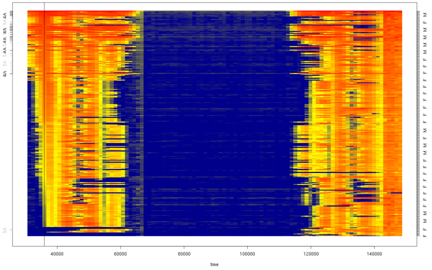


α, teleportation parameter of pagerank

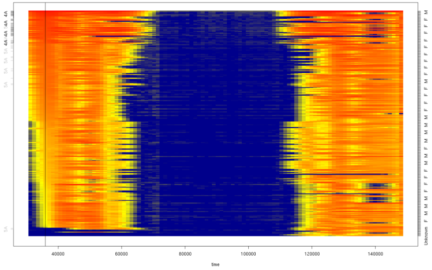


Small α means fast return to origins.

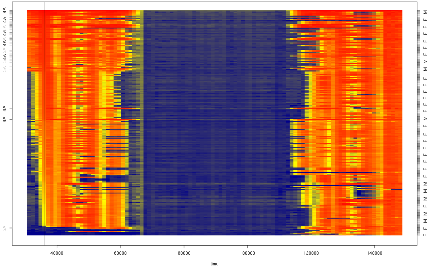
Visualisation



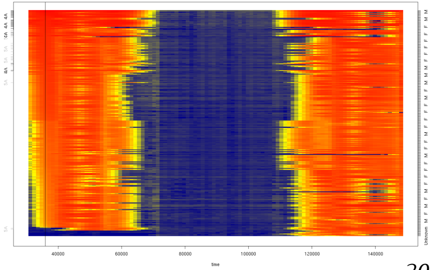
smoothing 600, $\alpha = 0.2$



smoothing 1200, $\alpha = 0.2$



smoothing 600, $\alpha = 0.8$



smoothing 1200, $\alpha = 0.8$