Graph visualization by organized clustering: application to social and biological networks

Nathalie Villa-Vialaneix
http://www.nathalievilla.org
& Fabrice Rossi

IUT de Carcassonne (UPVD) & Institut de Mathématiques de Toulouse

Challenging problems in Statistical Learning, Paris
January 28/29, 2010

Network data

Many sources of large networks

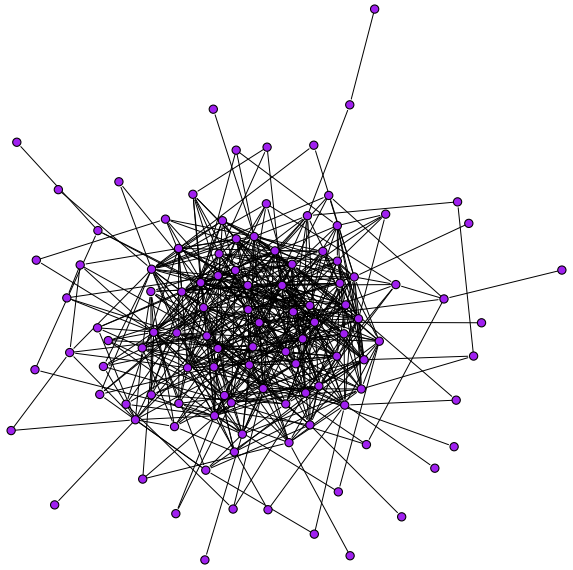
- social networks (emails, collaborations, phone calls, etc.)
- technological networks (Internet, etc.)
- biological networks (metabolic pathways, gene regulation, gene interactions, etc.)

Network data

Many sources of large networks

- social networks (emails, collaborations, phone calls, etc.)
- technological networks (Internet, etc.)
- biological networks (metabolic pathways, gene regulation, gene interactions, etc.)

Scope of the talk: A graph $\mathcal{G}$, with vertices $\{x_1, x_2, \ldots, x_n\}$, **undirected and weighted** with weights W such that: $w_{ii} = 0$ (no loop), $w_{ij} = w_{ji} \geq 0$ ($w_{ij} > 0 \Leftrightarrow \exists$ edge between nodes x_i and x_j).



Network data analysis

Over one hundred vertices, **pure manual analysis is infeasible**
$\Rightarrow$ need for automatic support for exploratory analysis:

- node/edge measures (e.g., degree distribution, betweenness, ...)
- visualization (e.g., force directed algorithm)
- node clustering (community extraction)
- supervised analysis
- ...

Network data analysis

Over one hundred vertices, **pure manual analysis is infeasible**
$\Rightarrow$ need for automatic support for exploratory analysis:

- node/edge measures (e.g., degree distribution, betweenness, ...)
- **visualization** (e.g., force directed algorithm)
- **node clustering** (community extraction)
- supervised analysis
- ...

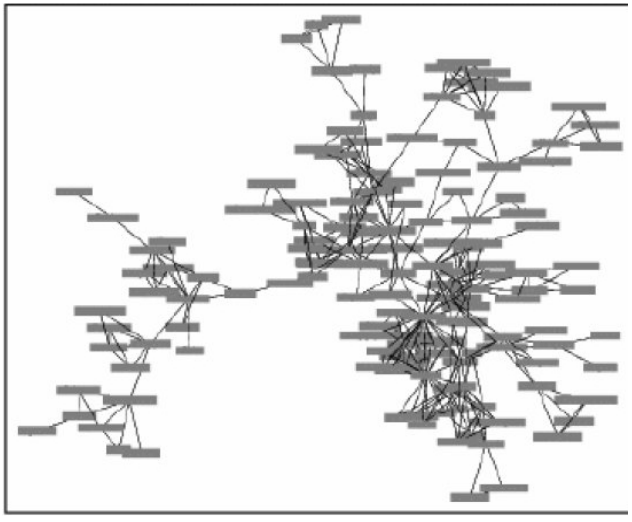
Visualization from clustering

Large graph visualization is difficult.

Visualization from clustering

Large graph visualization is difficult.
A standard solution: simplify the graph prior drawing. More precisely

1. identify dense clusters of nodes
2. draw the corresponding graph of clusters

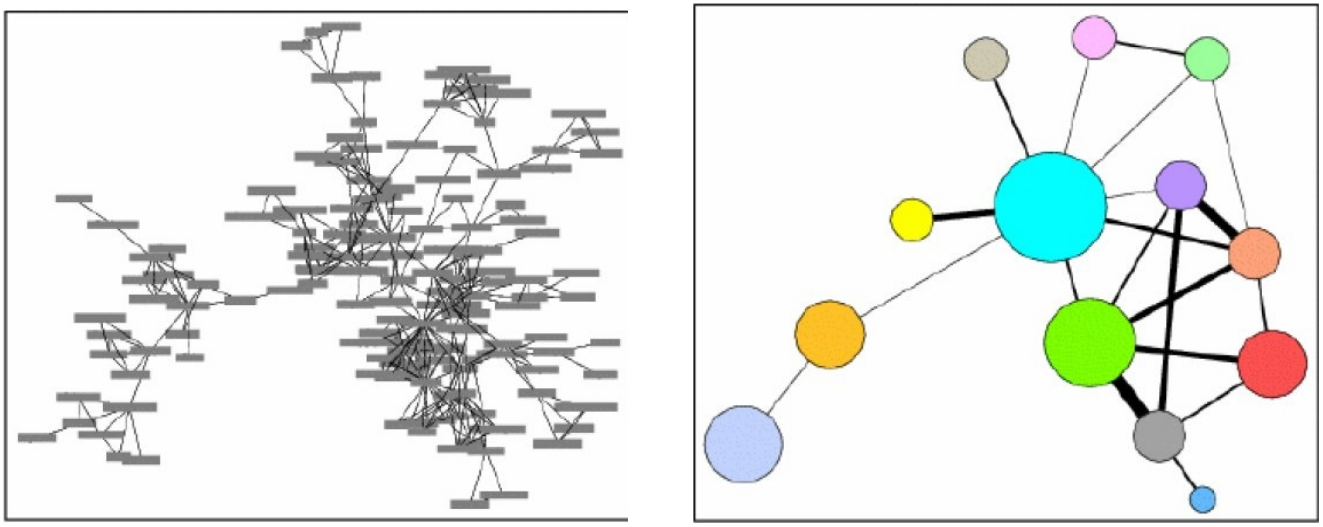


[Newman and Girvan, 2004]

Visualization from clustering

Large graph visualization is difficult.
A standard solution: simplify the graph prior drawing. More precisely

1. identify dense clusters of nodes
2. draw the corresponding graph of clusters



[Newman and Girvan, 2004]

Drawing a clustered graph

- Given a **partition** $(C_k)_{k=1,\dots,C}$
  - represent each cluster by a glyph (e.g., a circle) with surface proportional to $|C_k|$
  - draw a segment between glyphs C_k and C_l with thickness proportional to $\sum_{i \in C_k,\ j \in C_l} W_{ij}$

Drawing a clustered graph

- Given a **partition** $(C_k)_{k=1,\dots,C}$
  - represent each cluster by a glyph (e.g., a circle) with surface proportional to $|C_k|$
  - draw a segment between glyphs C_k and C_l with thickness proportional to $\sum_{i \in C_k,\ j \in C_l} W_{ij}$
- The **simplification induced by the clustering has to be faithful**: each cluster should be as dense as possible (i.e., $\sum_{i,j \in C_k} W_{ij}$ should be high compared to the other weights).

Drawing a clustered graph

- Given a **partition** $(C_k)_{k=1,\dots,C}$
  - represent each cluster by a glyph (e.g., a circle) with surface proportional to $|C_k|$
  - draw a segment between glyphs C_k and C_l with thickness proportional to $\sum_{i \in C_k,\ j \in C_l} W_{ij}$
- The **simplification induced by the clustering has to be faithful**: each cluster should be as dense as possible (i.e., $\sum_{i,j \in C_k} W_{ij}$ should be high compared to the other weights).
- The **graph induced by the clustering has to be readable**: edge crossing should be minimized.

Two approaches based on organizing maps

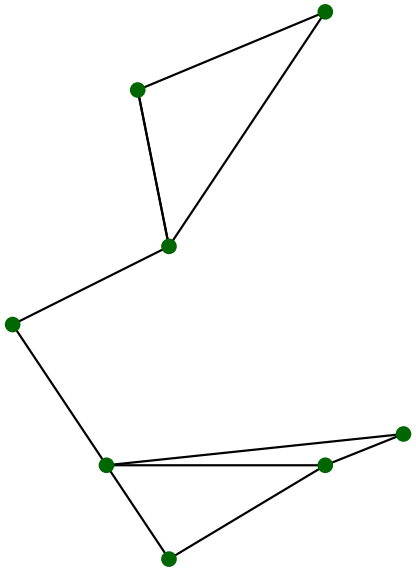
Main idea: organizing the clustering on a grid to constrain clusters' position and to represent the most connected clusters close to each others.

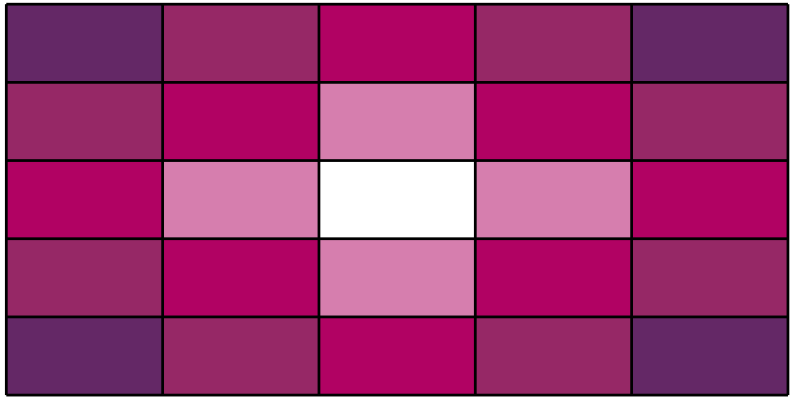
Two approaches based on organizing maps

Main idea: organizing the clustering on a grid to constrain clusters' position and to represent the most connected clusters close to each others.

1. **Kernel SOM**: generalization of Self-Organizing Maps to graph by the use of a kernel
2. **Organized modularity optimization**: extension of a well-known clustering measure for graphs to organized clustering

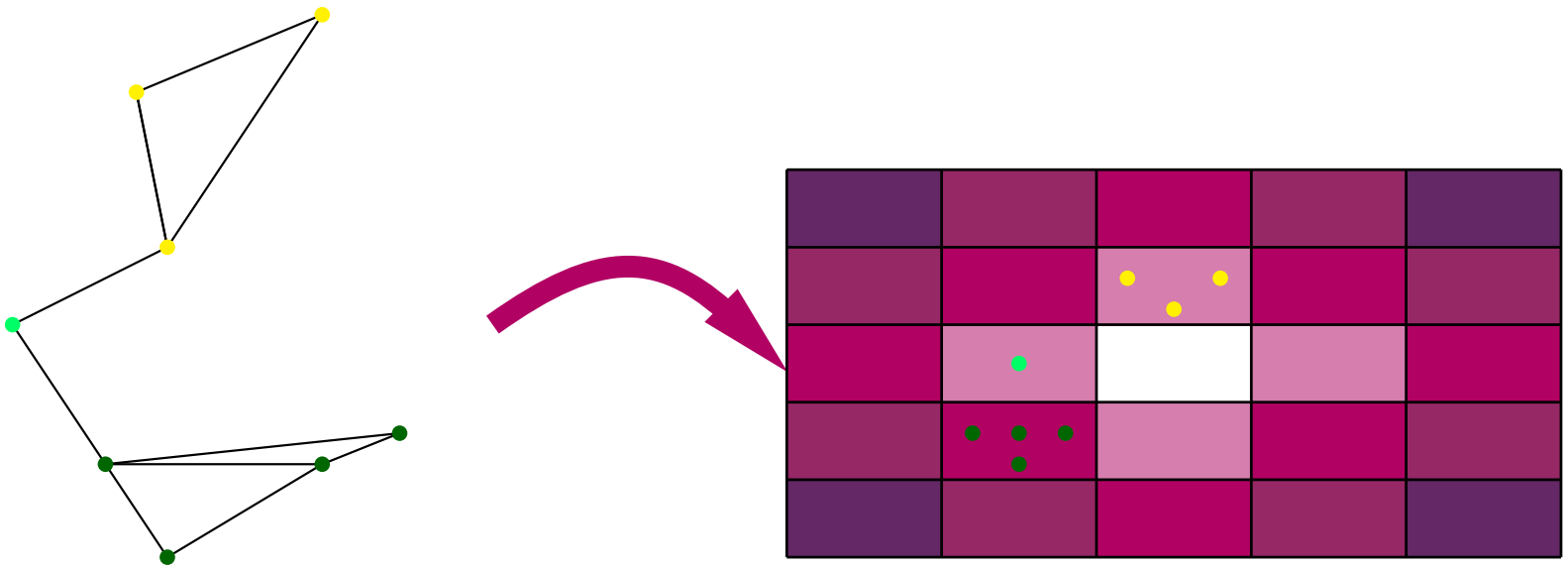
Basic ideas about SOM





Project the graph on a squared grid (each square of the grid is a cluster)

Basic ideas about SOM



Project the graph on a squared grid (each square of the grid is a cluster) such that:

- the nodes in a same cluster are highly connected
- the nodes in two close clusters are also (less) connected
- the nodes in two distant clusters are (almost) not connected

Original SOM algorithm (batch): $x_1, \ldots, x_n \in \mathbb{R}^d$

1. **Initalization**: Initialize randomly p_1^0, ..., p_M^0 in $\mathbb{R}^d$
2. **For** $l = 1, \ldots, L$ do
3. **Assignment**: for all $i = 1, \ldots, n$ do

$$f^l(x_i) = \arg \min_{j=1,\ldots,M} \|x_i - p_j^{l-1}\|_{\mathbb{R}^d}$$

4. **Representation**: for all $j = 1, \ldots, M$,

$$p_j^l = \arg \min_{p \in \mathbb{R}^d} \sum_{i=1}^{n} h^l(f^l(x_i), j) \|x_i - p\|_{\mathbb{R}^d}^2$$

SOM and kernel SOM

Kernel SOM (batch): $x_i \in \mathcal{G}$ defined by a kernel relation: $K(x_i, x_j)$
$\Rightarrow \exists \phi : \mathcal{G} \rightarrow (\mathcal{H}, \langle ., . \rangle_{\mathcal{H}})$: $K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$

1. **Initalization**: Initialize randomly $p_j^0 = \sum_{i=1}^n \gamma_{ji}^0 \phi(x_i)$
2. **For** $l = 1, \ldots, L$ do
3. **Assignment**: for all $i = 1, \ldots, n$ do

$$f^l(x_i) = \arg \min_{j=1,\ldots,M} \|\phi(x_i) - p_j^{l-1}\|_{\mathcal{H}}$$

4. **Representation**: for all $j = 1, \ldots, M$,

$$\gamma_j^l = \arg \min_{\gamma \in \mathbb{R}^n} \sum_{i=1}^n h^l(f^l(x_i), j) \|\phi(x_i) - \sum_{k=1}^n \gamma_k \phi(x_k)\|_{\mathcal{H}}^2$$

[Villa and Rossi, 2007, Hammer and Hasenfuss, 2007]

SOM and kernel SOM

Kernel SOM (batch): $x_i \in \mathcal{G}$ defined by a kernel relation: $K(x_i, x_j)$
$\Rightarrow \exists \phi : \mathcal{G} \rightarrow (\mathcal{H}, \langle ., . \rangle_{\mathcal{H}})$: $K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$

1. **Initalization**: Initialize randomly $p_j^0 = \sum_{i=1}^n \gamma_{ji}^0 \phi(x_i)$
2. **For** $l = 1, \ldots, L$ do
3. **Assignment**: for all $i = 1, \ldots, n$ do

$$f^l(x_i) = \arg \min_{j=1,\ldots,M} \sum_{k,k'=1}^{n} \gamma_{jk}^{l-1} \gamma_{jk'}^{l-1} K(x_k, x_{k'}) - 2 \sum_{k=1}^{n} \gamma_{jk}^{l-1} K(x_i, x_k)$$

4. **Representation**: for all $j = 1, \ldots, M$,

$$\gamma_{jk}^{l} = \frac{h(f^l(x_k), j)}{\sum_{k'=1}^{n} h(f^l(x_{k'}), j)}$$

[Villa and Rossi, 2007, Hammer and Hasenfuss, 2007]

SOM and kernel SOM

Kernel SOM (batch): $x_i \in \mathcal{G}$ defined by a kernel relation: $K(x_i, x_j)$
$\Rightarrow \exists \phi : \mathcal{G} \rightarrow (\mathcal{H}, \langle ., . \rangle_{\mathcal{H}})$: $K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$

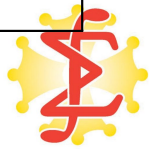
1. **Initalization**: Initialize randomly $p_j^0 = \sum_{i=1}^n \gamma_{ji}^0 \phi(x_i)$
2. **For** $l = 1, \ldots, L$ do
3. **Assignment**: for all $i = 1, \ldots, n$ do

$$f^l(x_i) = \arg \min_{j=1,\ldots,M} \sum_{k,k'=1}^{n} \gamma_{jk}^{l-1} \gamma_{jk'}^{l-1} K(x_k, x_{k'}) - 2 \sum_{k=1}^{n} \gamma_{jk}^{l-1} K(x_i, x_k)$$

4. **Representation**: for all $j = 1, \ldots, M$,

$$\gamma_{jk}^{l} = \frac{h(f^l(x_k), j)}{\sum_{k'=1}^{n} h(f^l(x_{k'}), j)}$$

Online versions by **[Lau et al., 2006]**



Which kernels?

Laplacian: $L = (L_{i,j})_{i,j=1,\dots,n}$ where

$$L_{i,j} = \begin{cases} -w_{i,j} & \text{if } i \neq j \\ d_i = \sum_{j \neq i} w_{i,j} & \text{if } i = j \end{cases};$$

Which kernels?

Laplacian: $L = (L_{i,j})_{i,j=1,\dots,n}$ where

$$L_{i,j} = \begin{cases} -w_{i,j} & \text{if } i \neq j \\ d_i = \sum_{j \neq i} w_{i,j} & \text{if } i = j \end{cases};$$

Regularized versions such as

- **Heat kernel**
  [Kondor and Lafferty, 2002, Smola and Kondor, 2003]: for $\beta > 0$, $K^\beta = e^{-\beta L} = \sum_{k=1}^{+\infty} \frac{(-\beta L)^k}{k!}$.

Which kernels?

Laplacian: $L = (L_{i,j})_{i,j=1,\dots,n}$ where

$$L_{i,j} = \begin{cases} -w_{i,j} & \text{if } i \neq j \\ d_i = \sum_{j \neq i} w_{i,j} & \text{if } i = j \end{cases};$$

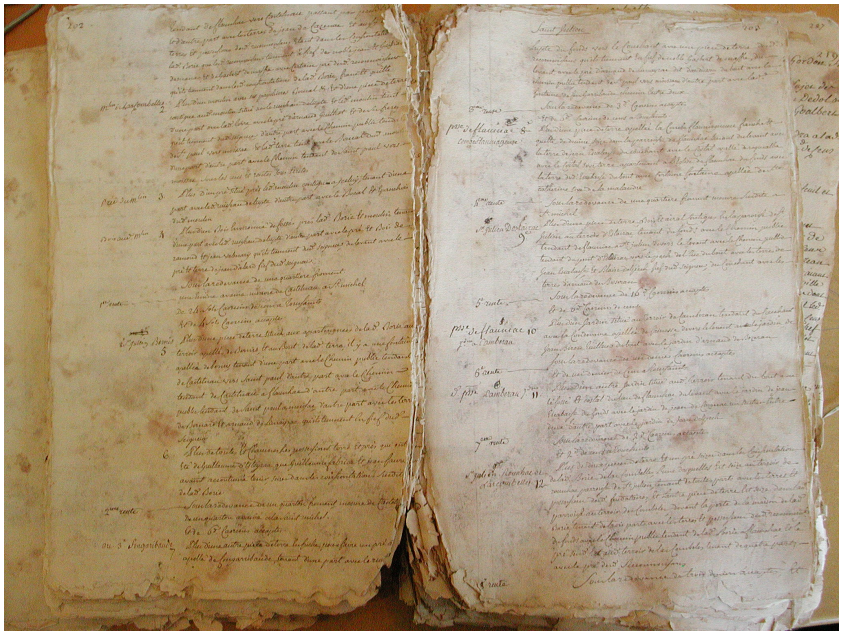
Regularized versions such as

- **Heat kernel**
  [Kondor and Lafferty, 2002, Smola and Kondor, 2003]: for $\beta > 0$, $K^\beta = e^{-\beta L} = \sum_{k=1}^{+\infty} \frac{(-\beta L)^k}{k!}$.
- **Generalized inverse of the Laplacian [Fouss et al., 2007]** : $K = L^+$.

A first example: a medieval social network

Example from [Boulet et al., 2008]

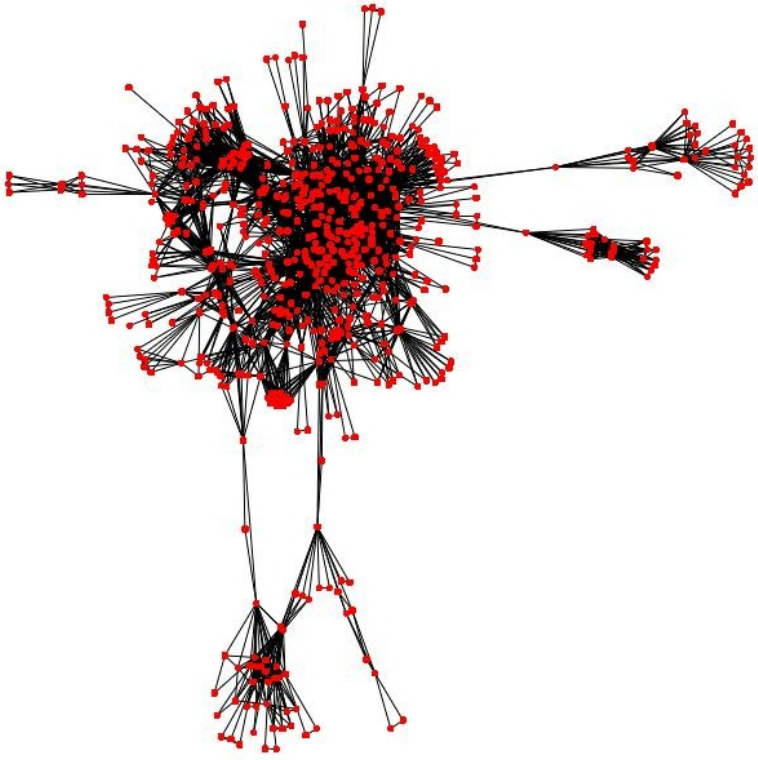
In Cahors (Lot, France), stands a big corpus of 5000 agrarian contracts coming from 4 seignories (about 25 little villages) and being established between 1240 and 1520 (just before and after the hundred years' war).



A first example: a medieval social network

Example from [Boulet et al., 2008]

In Cahors (Lot, France), stands a big corpus of 5000 agrarian contracts coming from 4 seignories (about 25 little villages) and being established between 1240 and 1520 (just before and after the hundred years' war).



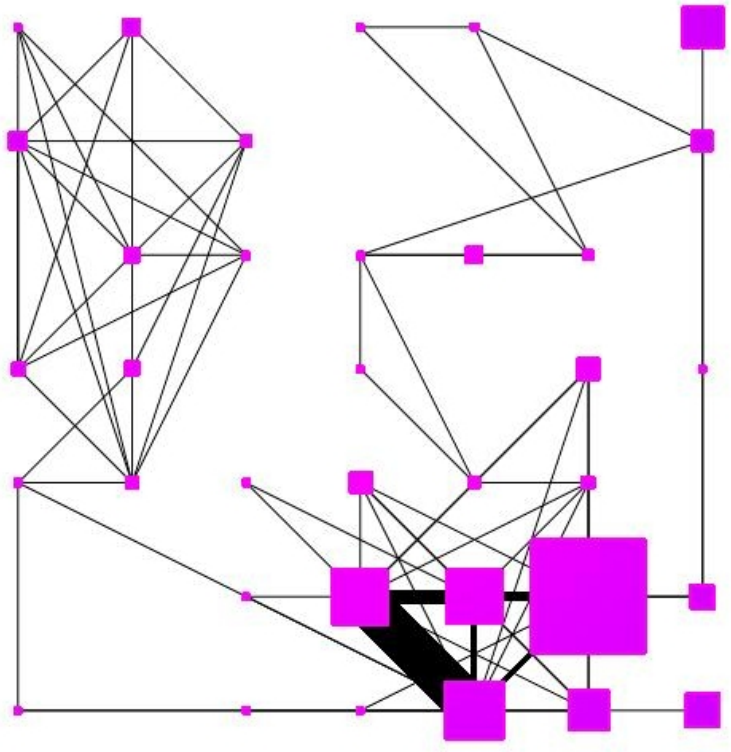
Network of relations between peasants based on common citations in a given contract.

A first example: a medieval social network

Example from [Boulet et al., 2008]

In Cahors (Lot, France), stands a big corpus of 5000 agrarian contracts coming from 4 seignories (about 25 little villages) and being established between 1240 and 1520 (just before and after the hundred years' war).

Graph of clusters: the communities have relations with time and space.
The leading people are emphasized.
But The biggest communities are still very complex.



Modularity [Newman and Girvan, 2004]

Popular **quality measure for graph clustering**: a partition of the vertices in C clusters, $(C_k)_{k=1,\dots,C}$ has modularity:

$$Q(C) = \frac{1}{2m} \sum_{k=1}^{C} \sum_{i,j \in C_k} (W_{ij} - P_{ij})$$

where P_{ij} are weights corresponding to a "null model" where the weights only depend on the nodes properties and not on the cluster they belong to.

Modularity [Newman and Girvan, 2004]

Popular **quality measure for graph clustering**: a partition of the vertices in C clusters, $(C_k)_{k=1,\dots,C}$ has modularity:

$$Q(C) = \frac{1}{2m} \sum_{k=1}^{C} \sum_{i,j \in C_k} (W_{ij} - P_{ij})$$

where P_{ij} are weights corresponding to a "null model" where the weights only depend on the nodes properties and not on the cluster they belong to.
More precisely,

$$P_{ij} = \frac{d_i d_j}{2m}$$

with $d_i = \frac{1}{2} \sum_{j \neq i} W_{ij}$ is the degree of a vertex x_i.

Modularity [Newman and Girvan, 2004]

Popular **quality measure for graph clustering**: a partition of the vertices in C clusters, $(C_k)_{k=1,\dots,C}$ has modularity:

$$Q(C) = \frac{1}{2m} \sum_{k=1}^{C} \sum_{i,j \in C_k} (W_{ij} - P_{ij})$$

where P_{ij} are weights corresponding to a "null model" where the weights only depend on the nodes properties and not on the cluster they belong to.
More precisely,

$$P_{ij} = \frac{d_i d_j}{2m}$$

with $d_i = \frac{1}{2} \sum_{j \neq i} W_{ij}$ is the degree of a vertex x_i.
A "good" clustering should **maximize** Q.

Interpretation

- Q increases when (x_i, x_j) are in a **same cluster** and have **true weight W_{ij} greater** than the ones expected in the null model, P_{ij}
- Q increases when (x_i, x_j) are in a **two different clusters** and have **true weight W_{ij} smaller** than the ones expected in the null model, P_{ij} because

$$Q(C) + \frac{1}{2m} \sum_{k \neq k'} \sum_{i \in C_k,\ j \in C_{k'}} (W_{ij} - P_{ij}) = 0.$$

Interpretation

- Q increases when (x_i, x_j) are in a **same cluster** and have **true weight W_{ij} greater** than the ones expected in the null model, P_{ij}
- Q increases when (x_i, x_j) are in a **two different clusters** and have **true weight W_{ij} smaller** than the ones expected in the null model, P_{ij} because

$$Q(C) + \frac{1}{2m} \sum_{k \neq k'} \sum_{i \in C_k,\ j \in C_{k'}} (W_{ij} - P_{ij}) = 0.$$

- Contrary to the minimization of the number of edges between clusters, modularity can help to **separate nodes with high degrees** into different clusters more easily

Drawing optimized clustering

Combine:

- **high modularity** to ensure high intra clusters density and low external connectivity
- **little edge crossing**

Drawing optimized clustering

Combine:

- **high modularity** to ensure high intra clusters density and low external connectivity
- **little edge crossing** by:
  - Classic solution: relying on graph drawing algorithm after maximization of the modularity

Drawing optimized clustering

Combine:

- **high modularity** to ensure high intra clusters density and low external connectivity
- **little edge crossing** by:
  - Classic solution: relying on graph drawing algorithm after maximization of the modularity
  - Extend the modularity to a criterium adapted to a prior structure (like a grid)

Self Organizing Map principle

For data in $\mathbb{R}^d$, SOM minimizes (over the clustering and the prototypes (p_k))

$$\sum_{k=1}^{C} \sum_{i=1}^{n} S_{f(x_i),k} \|x_i - p_k\|^2_{\mathbb{R}^d}$$

where:

- (p_k) are the prototypes (one for each cluster of the grid) representing the cluster in the original space ($\mathbb{R}^d$)
- $f(x_i)$ is the cluster, on the grid, where x_i is classified
- S_{kl} encodes the prior structure: close to 1 for close clusters and close to 0 for distant clusters

Self Organizing Map principle

For data in $\mathbb{R}^d$, SOM minimizes (over the clustering and the prototypes (p_k))

$$\sum_{k=1}^{C} \sum_{i=1}^{n} S_{f(x_i),k} \|x_i - p_k\|^2_{\mathbb{R}^d}$$

where:

- (p_k) are the prototypes (one for each cluster of the grid) representing the cluster in the original space ($\mathbb{R}^d$)
- $f(x_i)$ is the cluster, on the grid, where x_i is classified
- S_{kl} encodes the prior structure: close to 1 for close clusters and close to 0 for distant clusters

This corresponds to a **soft membership**: x_i belongs to C_k with membership $S_{f(x_i),k}$.

Organized modularity [Rossi and Villa-Vialaneix, 2010]

Same idea: encode a prior structure via a matrix S.
Maximize:

$$\mathcal{SQ} = \frac{1}{2m} \sum_{i,j} S_{f(i)f(j)} (W_{ij} - P_{ij})$$

Organized modularity [Rossi and Villa-Vialaneix, 2010]

Same idea: encode a prior structure via a matrix S.
Maximize:

$$\mathcal{SQ} = \frac{1}{2m} \sum_{i,j} S_{f(i)f(j)}(W_{ij} - P_{ij})$$

Hence:

- if a pair of vertices (x_i, x_j) is such that $W_{ij} > P_{ij}$, $\mathcal{SQ}$ increases with the closeness of $f(x_i)$ and $f(x_j)$ in the prior structure
- if a pair of vertices (x_i, x_j) is such that $W_{ij} < P_{ij}$, $\mathcal{SQ}$ increases if $f(x_i)$ and $f(x_j)$ are distant in the prior structure

Optimization

The clustering is represented by a $n \times C$ assignment matrix M with $M_{ik} = \delta_{f(i)=k}$. The goal is then to **maximize**

$$\mathcal{SQ} = F(M) = \frac{1}{2m} \sum_{i,j} \sum_{k,l} M_{ik} S_{kl} M_{lj} (W_{ij} - P_{ij})$$

Optimization

The clustering is represented by a $n \times C$ assignment matrix M with $M_{ik} = \delta_{f(i)=k}$. The goal is then to **maximize**

$$\mathcal{SQ} = F(M) = \frac{1}{2m} \sum_{i,j} \sum_{k,l} M_{ik} S_{kl} M_{lj} (W_{ij} - P_{ij})$$

Combinatorial problem is NP-complet $\Rightarrow$ use of **deterministic algorithm**:

- Given a temperature $\frac{1}{\beta}$, assume a Gibbs distribution on the solution space $P(M) = \frac{1}{Z_P} e^{\beta F(M)}$
- Compute $\mathbb{E}(M)$ with respect to P
- At the limit $\beta \to +\infty$, $\mathbb{E}(M)$ converges to M^* where M^* realizes the maximum of $F(M)$

Mean field approximation

Problem: $Z_P = \sum_M e^{\beta F(M)}$ is hard to compute (C^n values for M) except when the distribution factorizes (use of block calculations)

But $\mathcal{SQ}$ does not factorize!!!

Mean field approximation

Problem: $Z_P = \sum_M e^{\beta F(M)}$ is hard to compute (C^n values for M) except when the distribution factorizes (use of block calculations)

But $\mathcal{SQ}$ does not factorize!!!

Solution: approximate $P(M)$ by a distribution that factorizes:

- $P(M)$ **is approximated** by

$$R(M,E) = \frac{e^{\beta \sum_{i,k} M_{ik} E_{ik}}}{\sum_N e^{\beta \sum_{i,k} N_{ik} E_{ik}}}.$$

Mean field approximation

Problem: $Z_P = \sum_M e^{\beta F(M)}$ is hard to compute (C^n values for M) except when the distribution factorizes (use of block calculations)

But $\mathcal{SQ}$ does not factorize!!!

Solution: approximate $P(M)$ by a distribution that factorizes:

- $P(M)$ **is approximated** by

$$R(M,E) = \frac{e^{\beta \sum_{i,k} M_{ik} E_{ik}}}{\sum_N e^{\beta \sum_{i,k} N_{ik} E_{ik}}}.$$

  For the distribution $R(M,E)$, M_{ik} are **independants** for $i = 1, \ldots, n$.

Mean field approximation

Problem: $Z_P = \sum_M e^{\beta F(M)}$ is hard to compute (C^n values for M) except when the distribution factorizes (use of block calculations)

But $\mathcal{SQ}$ does not factorize!!!

Solution: approximate $P(M)$ by a distribution that factorizes:

- $P(M)$ **is approximated** by

$$R(M,E) = \frac{e^{\beta \sum_{i,k} M_{ik} E_{ik}}}{\sum_N e^{\beta \sum_{i,k} N_{ik} E_{ik}}}.$$

  For the distribution $R(M,E)$, M_{ik} are **independants** for $i = 1, \ldots, n$.

- The mean field E is tuned by **minimizing the Kullback-Leibler divergence** :

  $KL(R|P) = \sum_M R(M,E) \log \frac{R(M,E)}{P(M)} \Rightarrow$ **mean field equations**:

  $\frac{\partial \mathbb{E}_R(F(M))}{\partial E_{jl}} = \sum_k \frac{\partial \mathbb{E}_R(M_{jk})}{\partial E_{jl}} E_{jk}$

Mean field approximation

Problem: $Z_P = \sum_M e^{\beta F(M)}$ is hard to compute (C^n values for M) except when the distribution factorizes (use of block calculations)

But $\mathcal{SQ}$ does not factorize!!!

Solution: approximate $P(M)$ by a distribution that factorizes:

- $P(M)$ **is approximated** by

$$R(M,E) = \frac{e^{\beta \sum_{i,k} M_{ik} E_{ik}}}{\sum_N e^{\beta \sum_{i,k} N_{ik} E_{ik}}}.$$

  For the distribution $R(M,E)$, M_{ik} are **independants** for $i = 1, \ldots, n$.

- The mean field E is tuned by **minimizing the Kullback-Leibler divergence** :
  $KL(R|P) = \sum_M R(M,E) \log \frac{R(M,E)}{P(M)} \Rightarrow$ **mean field equations**:
  $\frac{\partial \mathbb{E}_R(F(M))}{\partial E_{jl}} = \sum_k \frac{\partial \mathbb{E}_R(M_{jk})}{\partial E_{jl}} E_{jk}$

- E_{ik} and $\mathbb{E}_R(M_{ik})$ are iteratively estimated by an EM-like algorithm; at the limit, $\mathbb{E}_R(M_{ik})$ gives the **probability of x_i to belong to cluster k for the optimal $\mathcal{SQ}$**

Deterministic annealing

Algorithm

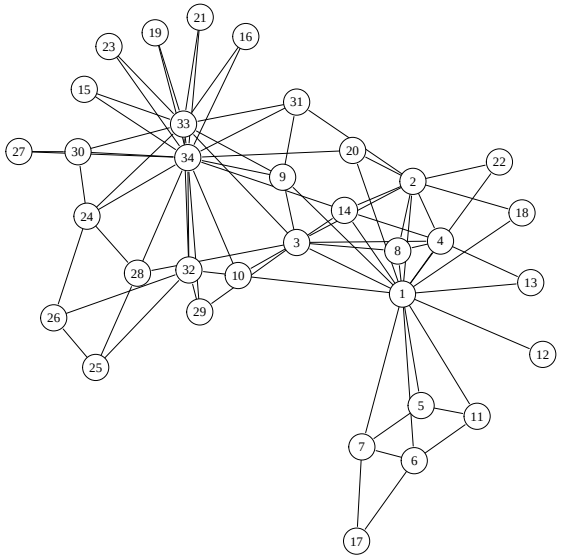
For increasing sequence $\beta_1, \beta_2, \ldots, \beta_L$,

1. **Initialize** $\mathbb{E}_R(M)$ randomly in $[0,1]$ such that $\sum_k \mathbb{E}_R(M_{ik}) = 1$
2. Repeat for $l = 1, \ldots, L$
   - 2.1 **Compute E** : $E_{ik} = 2 \sum_{j \neq i} \sum_{k'} \mathbb{E}_R(M_{jk'}) S_{kk'} B_{ji}$ where $B = \frac{1}{2m}(W - P)$;
   - 2.2 **Compute** $\mathbb{E}_R(M)$: $\mathbb{E}_R(M_{ik}) = \frac{e^{\beta_l E_{ik}}}{\sum_{k'} e^{\beta_l E_{ik'}}}$
3. Threshold $\mathbb{E}_R(M_{ik})$ into **clustering**:

$$M_{ik} = \arg \max_{k=1,\ldots,C} \mathbb{E}_R(M_{ik}).$$

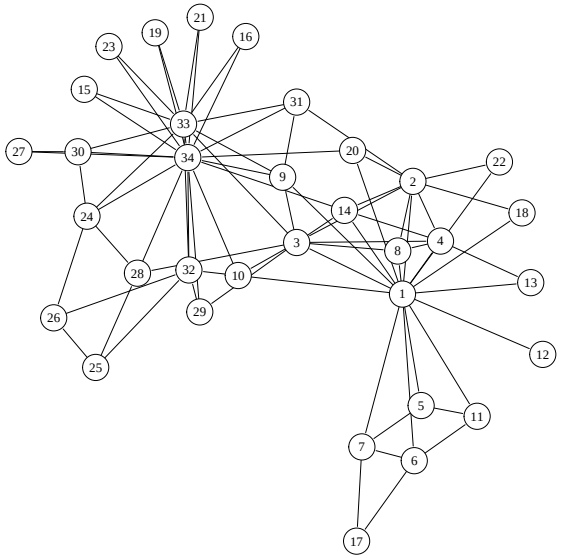
A toy example

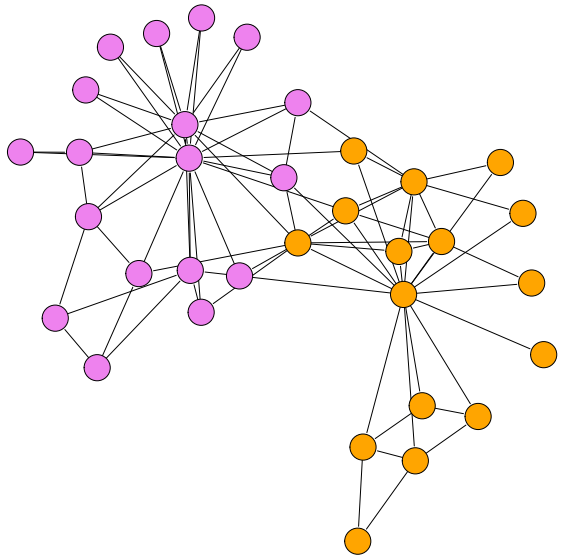
A toy example **[Zachary, 1977]**: **Zachary's karate club** (friendship social network between the **34 members** of a Karate club at a US university in the 70s).



A toy example

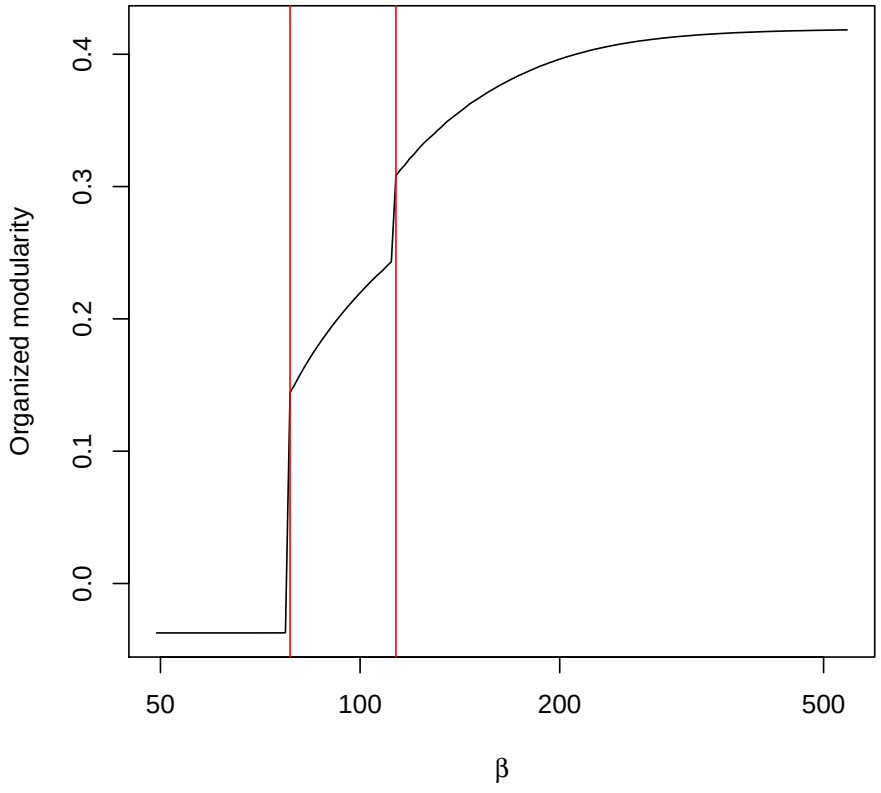
A toy example **[Zachary, 1977]**: **Zachary's karate club** (friendship social network between the **34 members** of a Karate club at a US university in the 70s).





Optimization of organized modularity on “Karate”

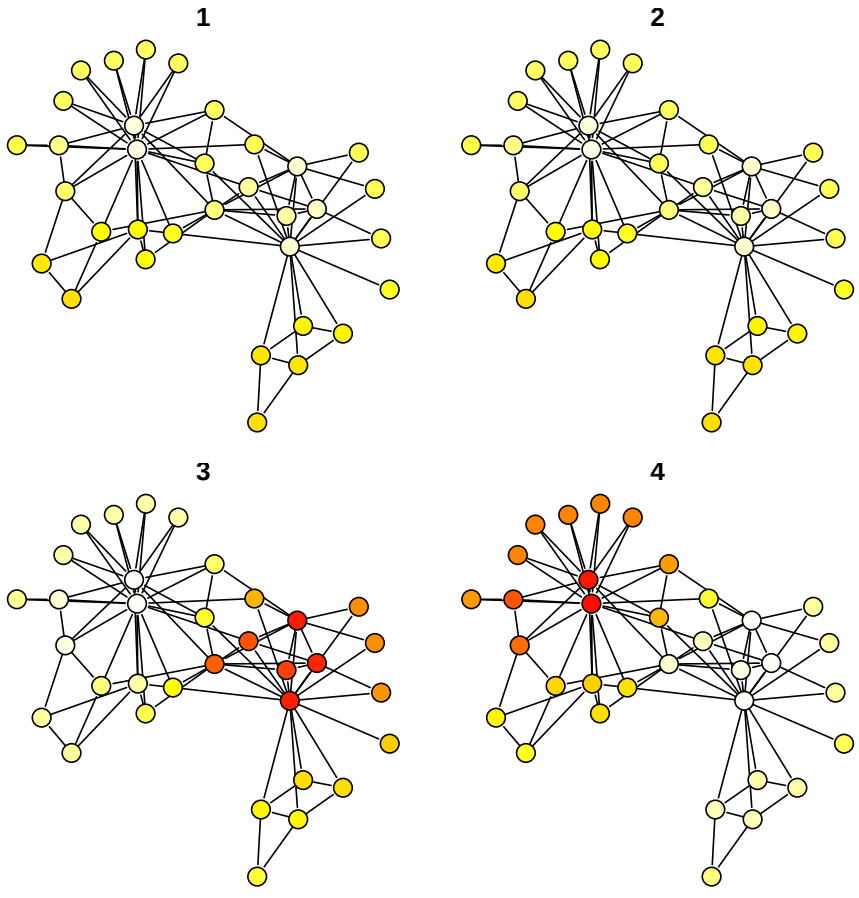
For a choice of neighborhood relationship leading to **4 non empty clusters** on a squared grid of size 2×2:



Evolution of the organized modularity during the annealing scheme

Optimization of organized modularity on “Karate”

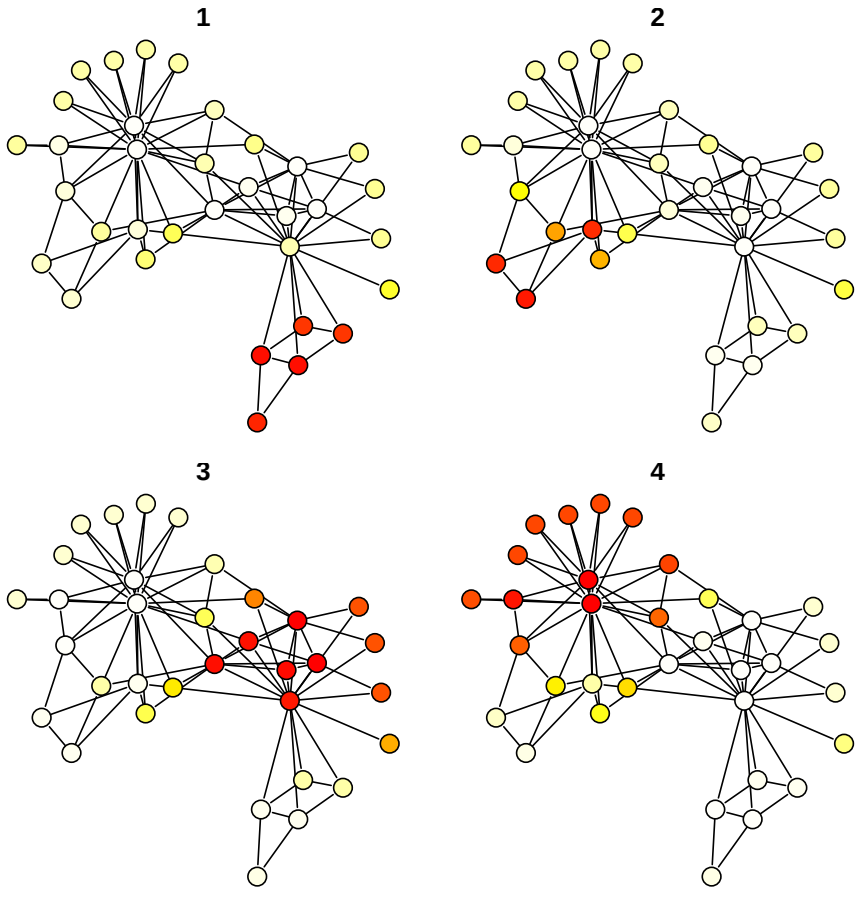
For a choice of neighborhood relationship leading to **4 non empty clusters** on a squared grid of size 2×2:



Probability of each node to be in a given classe just after the first phase transition

Optimization of organized modularity on “Karate”

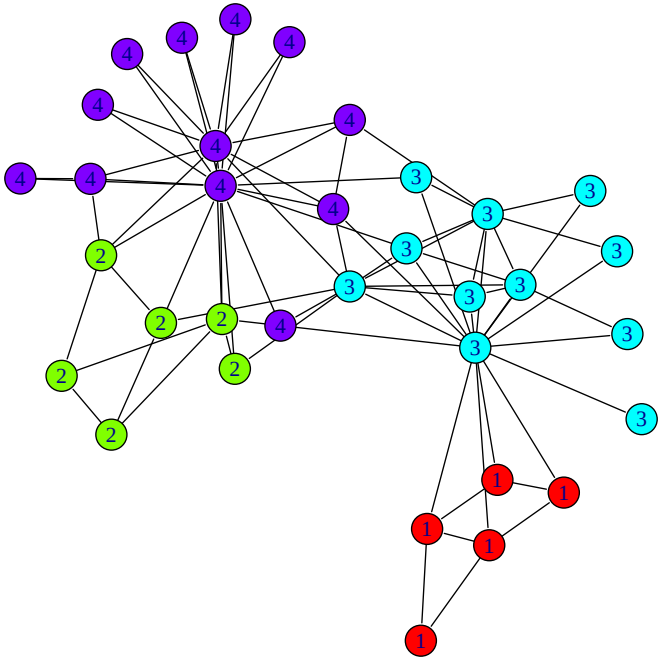
For a choice of neighborhood relationship leading to **4 non empty clusters** on a squared grid of size 2×2:

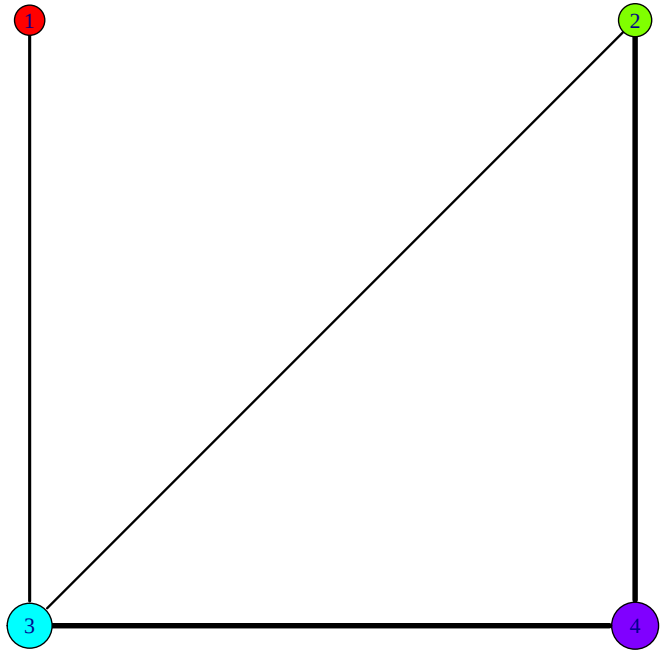


Probability of each node to be in a given classe just after the second phase transition

Optimization of organized modularity on “Karate”

For a choice of neighborhood relationship leading to **4 non empty clusters** on a squared grid of size 2×2:

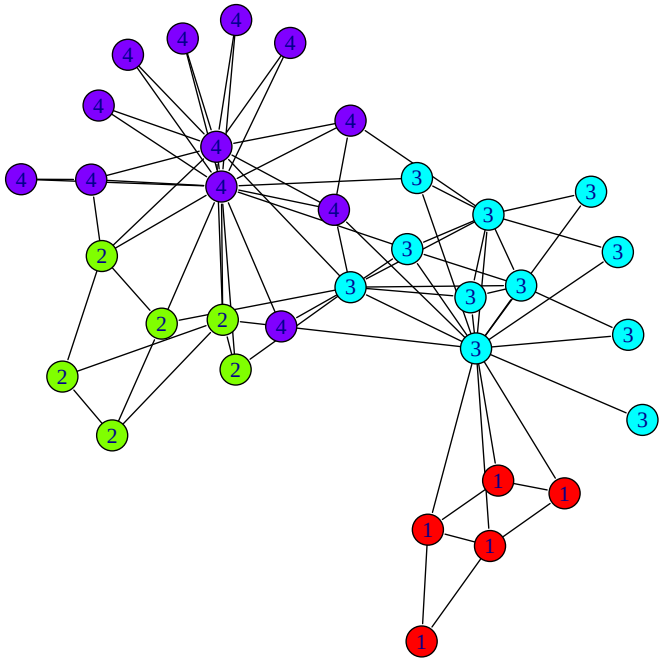


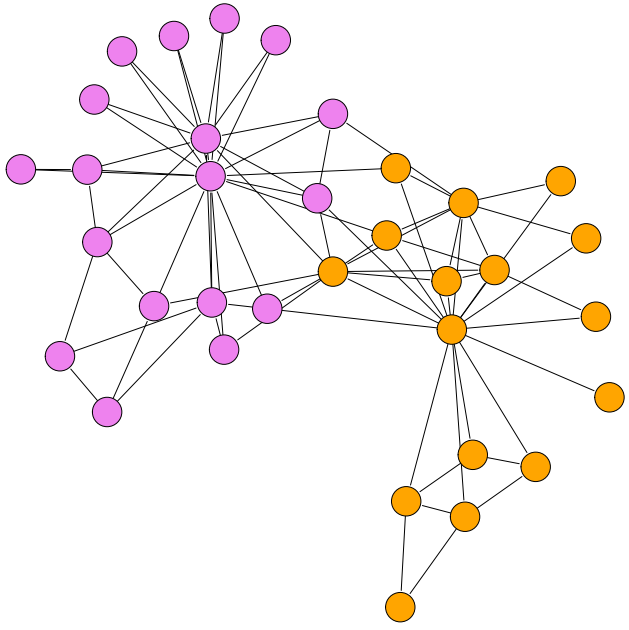


Final classification and layout

Optimization of organized modularity on “Karate”

For a choice of neighborhood relationship leading to **4 non empty clusters** on a squared grid of size 2×2:

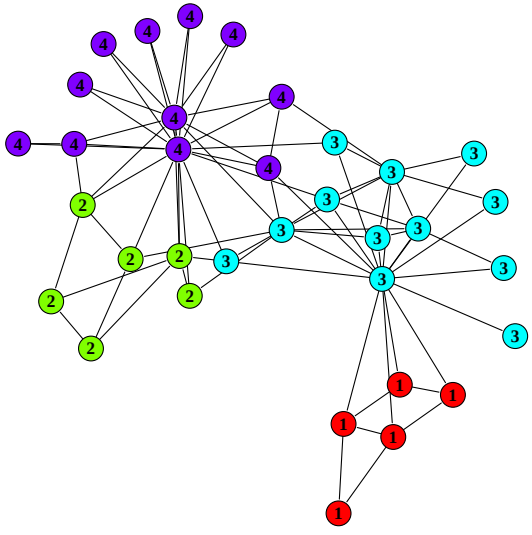




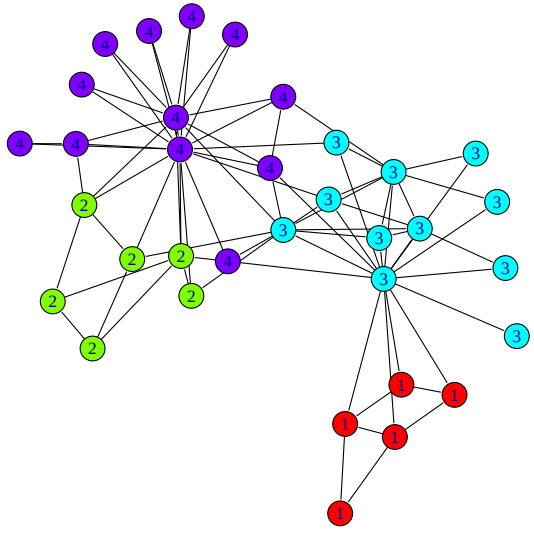
Final classification and layout

Comparisons on a toy example: “Karate”

Optimal solution obtained with SOM (various kernels tested):



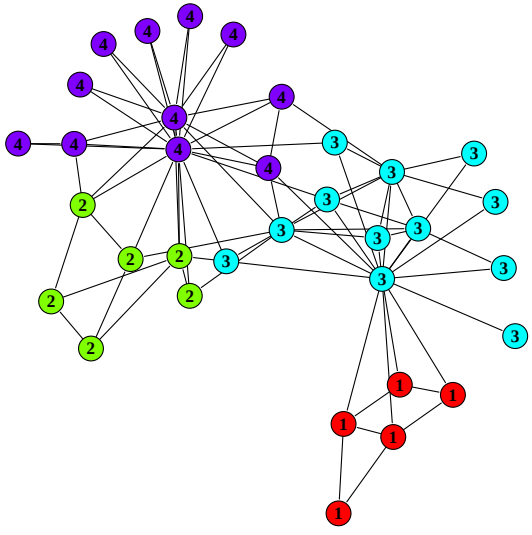
SOM (heat kernel)
Modularity = 0.4188



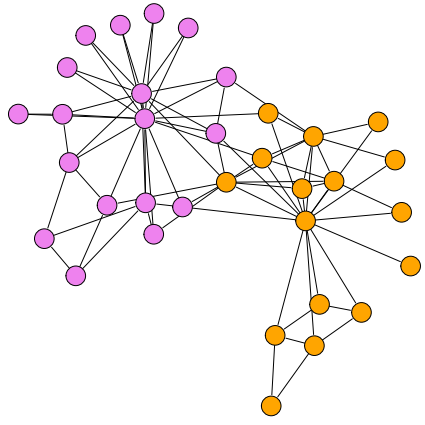
$\mathcal{SQ}$ optimization
Modularity = 0.4198
true optimum

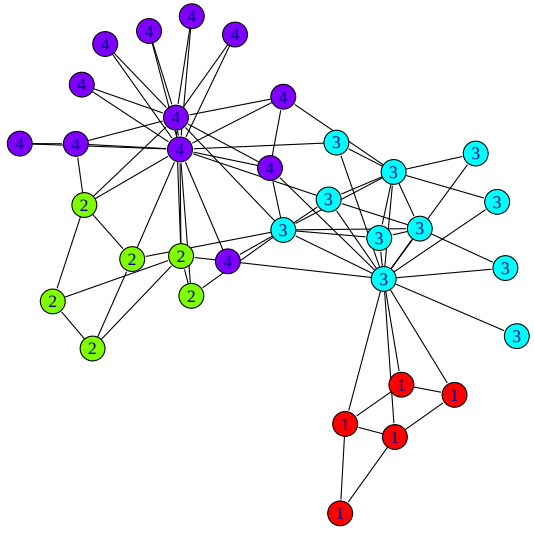
Comparisons on a toy example: “Karate”

Optimal solution obtained with SOM (various kernels tested):



SOM (heat kernel)
Modularity = 0.4188



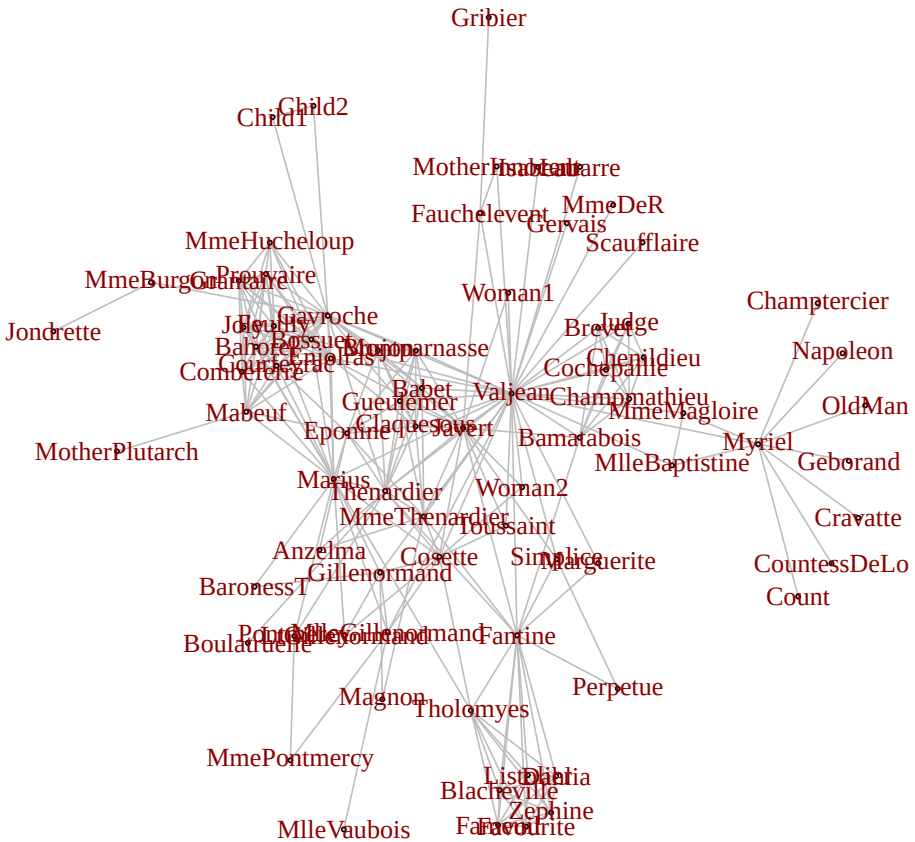


$\mathcal{SQ}$ optimization
Modularity = 0.4198
true optimum

$\mathcal{SQ}$ optimization solution is consistent with the true division of the social network

More comparisons on larger graphs

Co-appearance network from “Les Misérables” [Knuth, 1993]



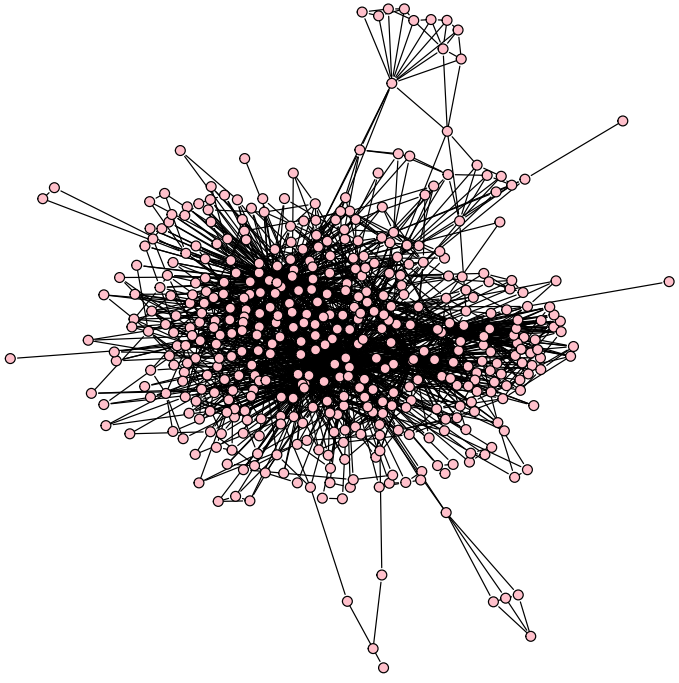
77 nodes

density = 8.7%

transitivity = 49.9 %

More comparisons on larger graphs

Neural network of worm C. Elegans (undirected version deduced from [Watts and Strogatz, 1998])



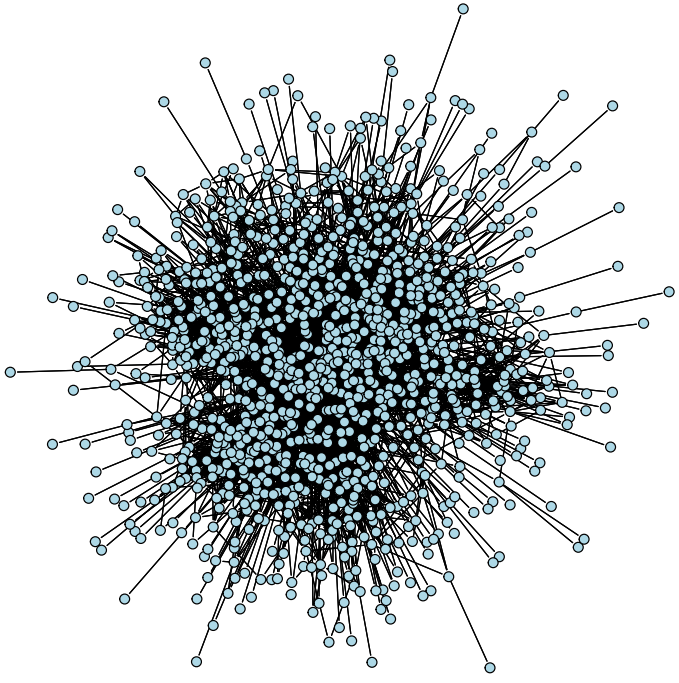
453 nodes

density = 2%

transitivity = 12.4%

More comparisons on larger graphs

E-mail exchanges between members of the University Rovira i Virgili (Tarragona) [Guimera et al., 2003]



1 133 nodes

density = 0.9%

transitivity = 16.6%

Methodology

Comparison of:

- **Direct approach** (modularity optimization + representation of the graph of clusters)
- **Kernel SOM** with various kernels: heat kernel, generalized inverse of the Laplacian, modularity kernel (i.e., the positive part of $W - P$ which mimics the optimization of the modularity) and spectral SOM (based on the first C eigenvectors of the Laplacian)
- $\mathcal{SQ}$ **optimization**

Methodology

Comparison of:

- **Direct approach** (modularity optimization + representation of the graph of clusters)
- **Kernel SOM** with various kernels: heat kernel, generalized inverse of the Laplacian, modularity kernel (i.e., the positive part of $W - P$ which mimics the optimization of the modularity) and spectral SOM (based on the first C eigenvectors of the Laplacian)
- $\mathcal{SQ}$ **optimization**

Parameters varied:

- size of the prior grid or number of clusters
- for organized clusterings, type of neighborhood on the grid
- for SOM, random or PCA initialization and kernel parameter for the heat kernel

Methodology

Comparison of:

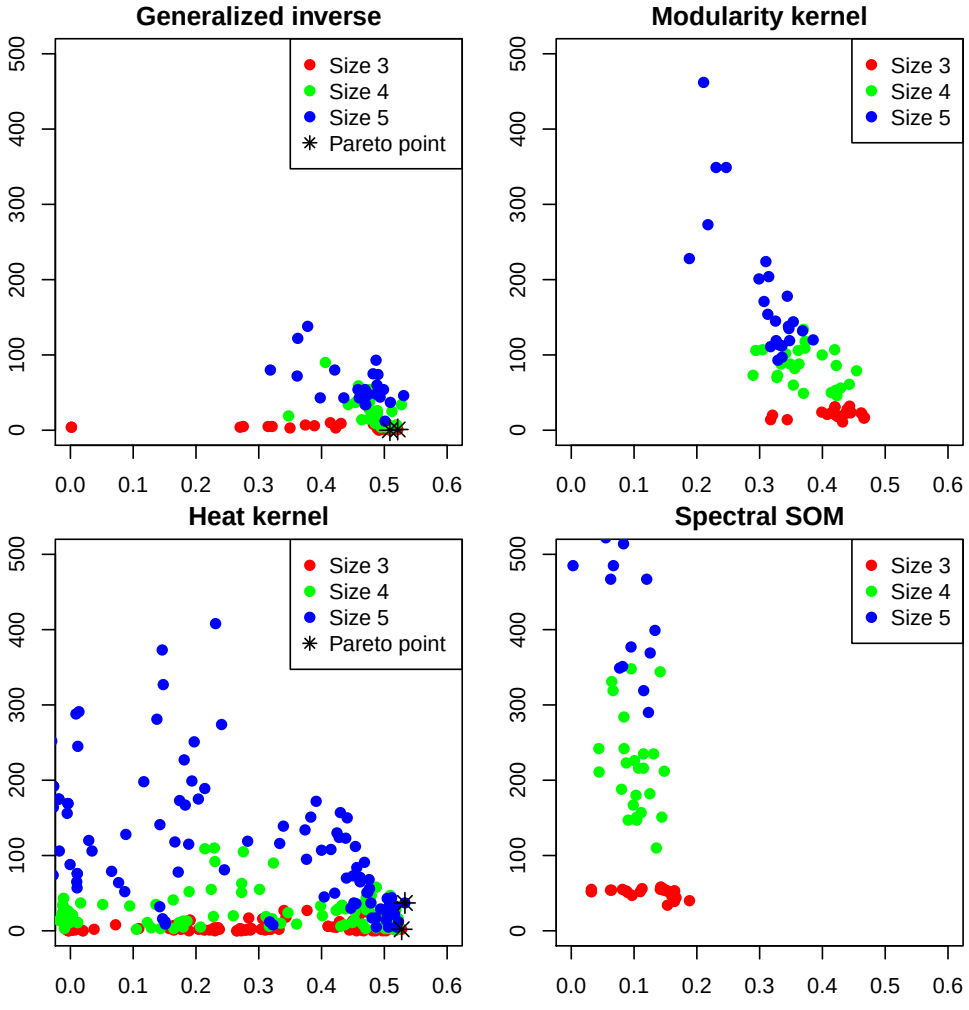
- **Direct approach** (modularity optimization + representation of the graph of clusters)
- **Kernel SOM** with various kernels: heat kernel, generalized inverse of the Laplacian, modularity kernel (i.e., the positive part of $W - P$ which mimics the optimization of the modularity) and spectral SOM (based on the first C eigenvectors of the Laplacian)
- $\mathcal{SQ}$ **optimization**

Parameters varied:

- size of the prior grid or number of clusters
- for organized clusterings, type of neighborhood on the grid
- for SOM, random or PCA initialization and kernel parameter for the heat kernel

Selection of the solutions: Pareto points according to modularity and number of edge crossing

A brief comment on SOM solutions with “Les Misérables”



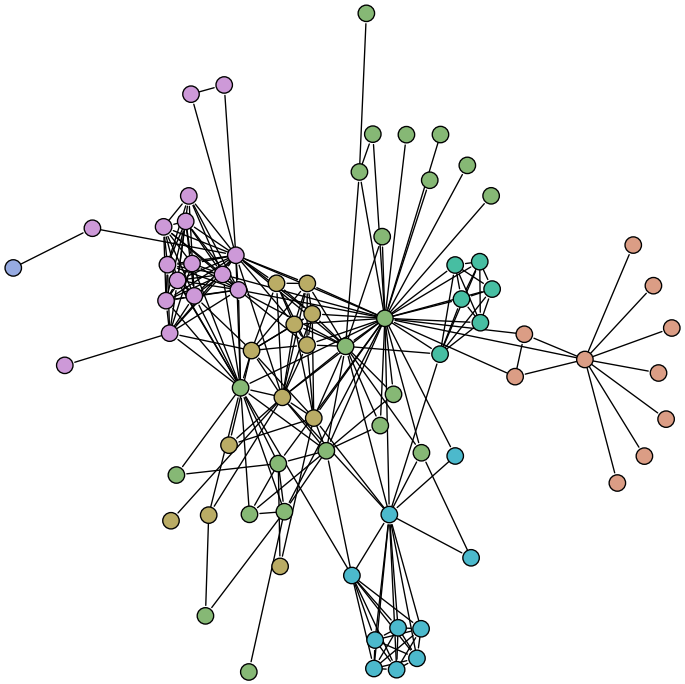
Spectral SOM and Modularity kernel obtain poor results

Analysis of the Pareto points for “Les Misérables”

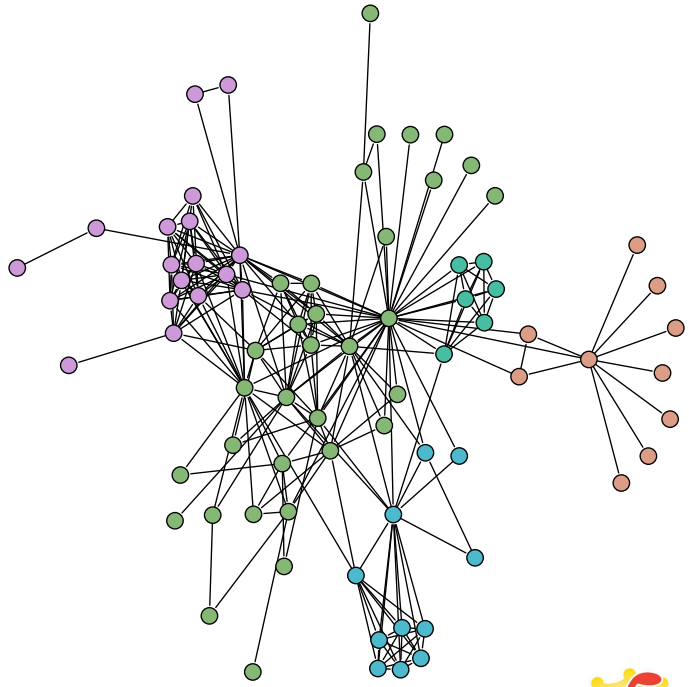
Method	Number of clusters	Modularity	Nb of pairs of cut edges	Id
Organized mod.	4^2 (7)	0.5638	1	**M5**
Organized mod.	5^2 (7)	0.5652	3	**M6**
	3^2 (6)	0.5472	0	**M7**
Modularity optimization	8 (5)	0.5472	0	**M8**

Analysis of the Pareto points for “Les Misérables”

Method	Number of clusters	Modularity	Nb of pairs of cut edges	Id
Organized mod.	4^2 (7)	0.5638	1	**M5**
Organized mod.	5^2 (7)	0.5652	3	**M6**
	3^2 (6)	0.5472	0	**M7**
Modularity optimization	8 (5)	0.5472	0	**M8**



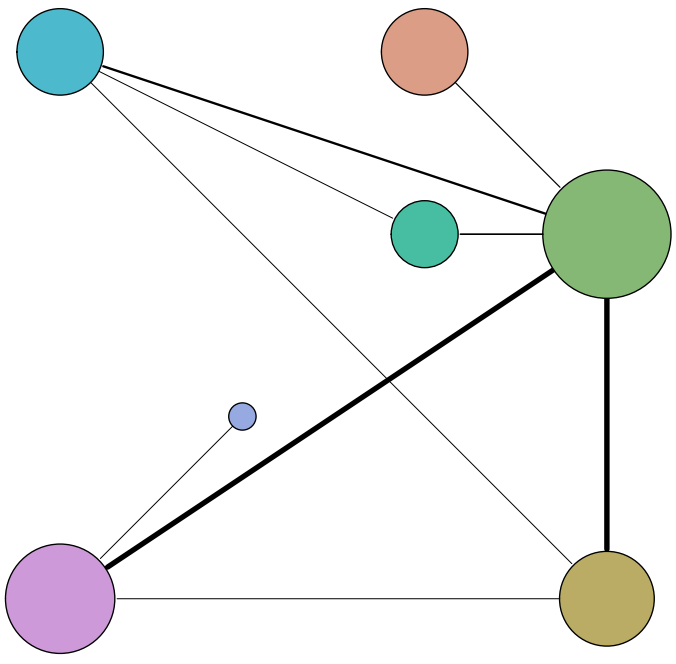
M5:



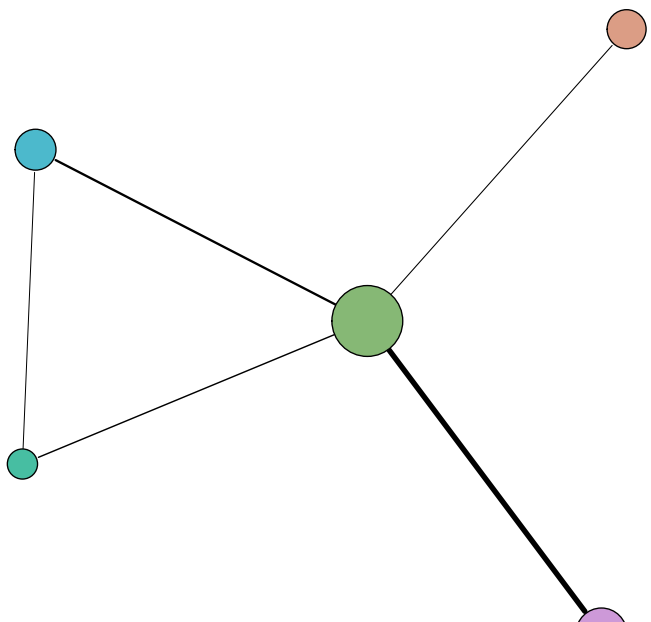
M8:

Analysis of the Pareto points for “Les Misérables”

Method	Number of clusters	Modularity	Nb of pairs of cut edges	Id
Organized mod.	4^2 (7)	0.5638	1	**M5**
Organized mod.	5^2 (7)	0.5652	3	**M6**
	3^2 (6)	0.5472	0	**M7**
Modularity optimization	8 (5)	0.5472	0	**M8**



M5:



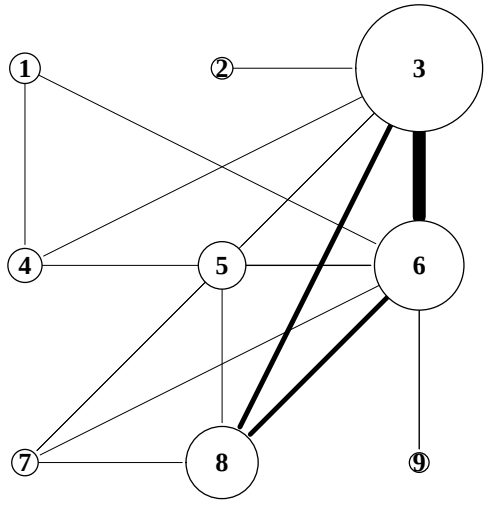
M8:

Analysis of the Pareto points for “C. Elegans”

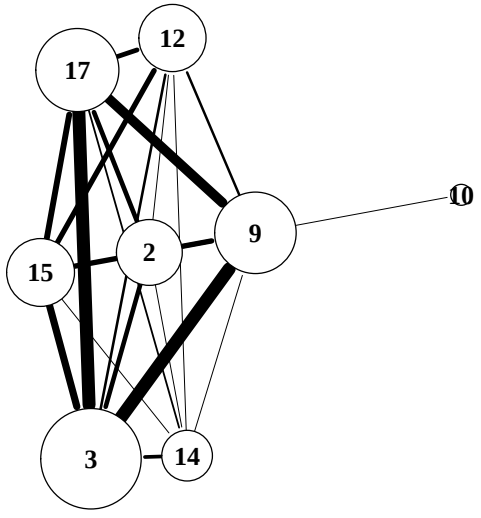
Method	Number of clusters	Modularity	Nb of pairs of cut edges	Id
SOM (GInv)	3^2 (9)	0.3228	14	**CE1**
	3^2 (9)	0.3000	7	**CE2**
	3^2 (8)	0.2936	1	**CE3**
Organized mod.	3^2 (7)	0.4321	19	**CE6**
Organized mod.	3^2 (8)	0.4063	15	**CE7**
Modularity optimization	18 (8)	0.4383	27	**CE8**

Analysis of the Pareto points for “C. Elegans”

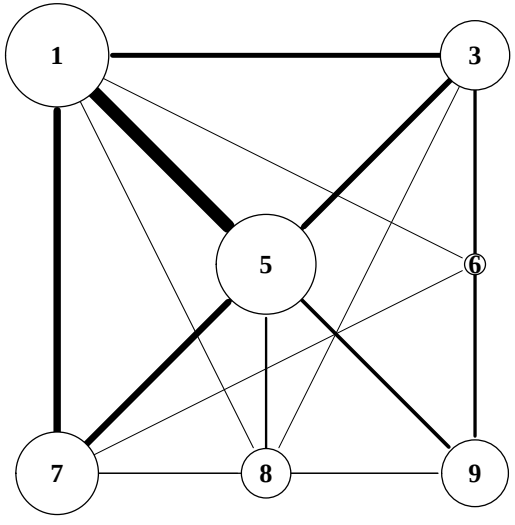
CE1



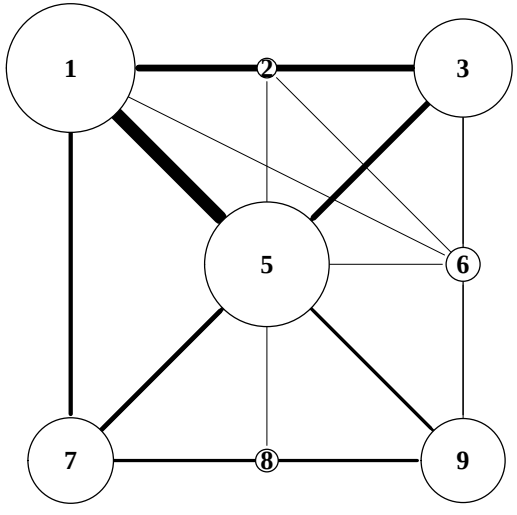
CE8



CE6



CE7



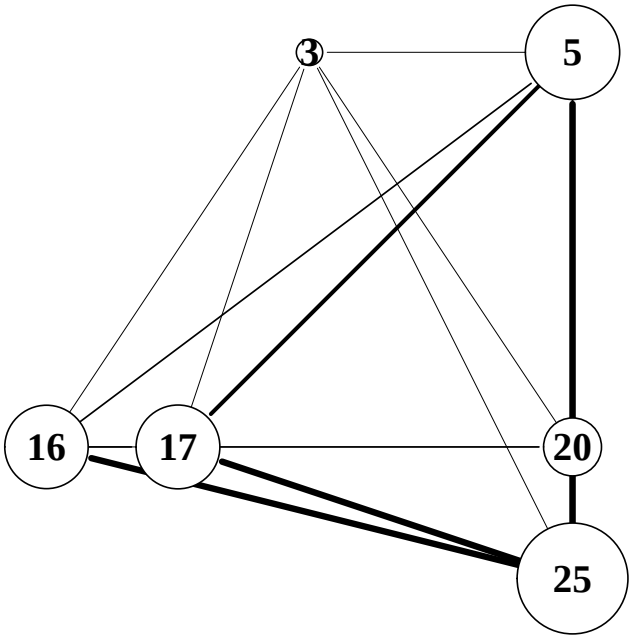
Analysis of the Pareto points for “Email”

Method	Number of clusters	Modularity	Nb of pairs of cut edges	Id
Organized mod. (Gaussian)	3^2 (8)	0.5694	47	**E6**
	4^2 (8)	0.5693	44	**E7**
	4^2 (7)	0.5554	25	**E8**
	3^2 (7)	0.5456	23	**E9**
	5^2 (6)	0.5401	11	**E10**
Modularity optimization	11 (8)	0.5736	56	**E11**

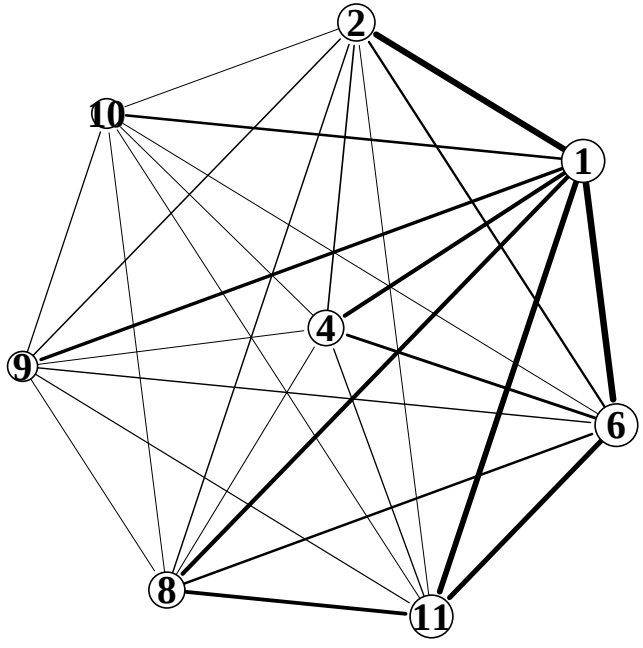
Analysis of the Pareto points for “Email”

Method	Number of clusters	Modularity	Nb of pairs of cut edges	Id
Organized mod. (Gaussian)	3^2 (8)	0.5694	47	**E6**
	4^2 (8)	0.5693	44	**E7**
	4^2 (7)	0.5554	25	**E8**
	3^2 (7)	0.5456	23	**E9**
	5^2 (6)	0.5401	11	**E10**
Modularity optimization	11 (8)	0.5736	56	**E11**

E10



E11



Conclusion

Organized clustering for graph

- finds clusters adapted to visualization
- competes with two steps approaches: with a little cost in clustering quality, can provide a more simplified graph

Conclusion

Organized clustering for graph

- finds clusters adapted to visualization
- competes with two steps approaches: with a little cost in clustering quality, can provide a more simplified graph

Organized modularity

- seems to be more performant than kernel SOM: hub are separated easier
- has a computational cost that remains acceptable: comparable to modularity optimization (but more parameters to tune)

Boulet, R., Jouve, B., Rossi, F., and Villa, N. (2008).

Batch kernel SOM and related laplacian methods for social network analysis.
Neurocomputing, 71(7-9):1257–1273.

Fouss, F., Pirotte, A., Renders, J., and Saerens, M. (2007).

Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation.
IEEE Transactions on Knowledge and Data Engineering, 19(3):355–369.

Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., and Arenas, A. (2003).

Self-similar community structure in a network of human interactions.
Physical Review E, 68(065103(R)).

Hammer, B. and Hasenfuss, A. (2007).

Relational topographic maps.
Technical Report IfI-07-01, Clausthal University of Technology.

Knuth, D. (1993).

The Stanford GraphBase: A Platform for Combinatorial Computing.
Addison-Wesley, Reading, MA.

Kondor, R. and Lafferty, J. (2002).

Diffusion kernels on graphs and other discrete structures.
In *Proceedings of the 19th International Conference on Machine Learning*, pages 315–322.

Lau, K., Yin, H., and Hubbard, S. (2006).

Kernel self-organising maps for classification.
Neurocomputing, 69:2033–2040.

Newman, M. and Girvan, M. (2004).

Finding and evaluating community structure in networks.
Physical Review, E, 69:026113.

Rossi, F. and Villa-Vialaneix, N. (2010).

Optimizing an organized modularity measure for topographic graph clustering : a deterministic annealing approach.

Neurocomputing.
To appear.

Smola, A. and Kondor, R. (2003).

Kernels and regularization on graphs.
In Warmuth, M. and Schölkopf, B., editors, *Proceedings of the Conference on Learning Theory (COLT) and Kernel Workshop*.

Villa, N. and Rossi, F. (2007).

A comparison between dissimilarity SOM and kernel SOM for clustering the vertices of a graph.
In *Proceedings of the 6th Workshop on Self-Organizing Maps (WSOM 07)*, Bielefield, Germany.

Watts, D. and Strogatz, S. (1998).

Collective dynamics of “small-world” networks.
Nature, 393:440–442.

Zachary, W. (1977).

An information flow model for conflict and fission in small groups.
Journal of Anthropological Research, 33(4):452–473.