Graph Wavelets for Multiscale Community Mining

Nicolas Tremblay and Pierre Borgnat

Presented by Lorenzo Dall'Amico

May 12, 2020

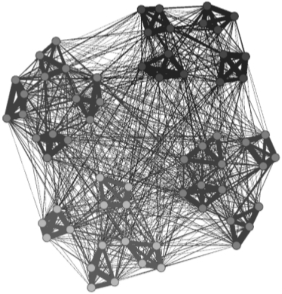
Outline

Table of Contents

Introduction

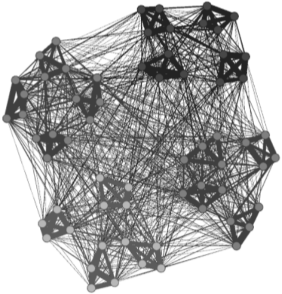
- What
- Why
- How

What: community mining



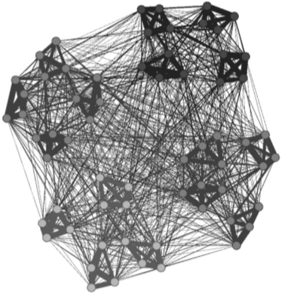
- Many real networks have a modular structure

What: community mining



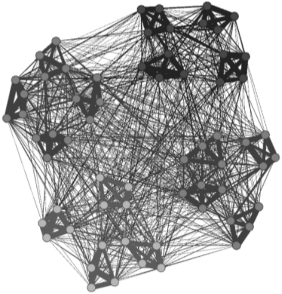
- Many real networks have a modular structure
- Items belonging to the same module (community) have common properties

What: community mining



- Many real networks have a modular structure
- Items belonging to the same module (community) have common properties
- The representation of a system through a weighted similarity graph is very general

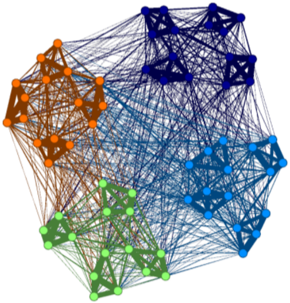
What: community mining



e.g. The PhD students in the same city

- Many real networks have a modular structure
- Items belonging to the same module (community) have common properties
- The representation of a system through a weighted similarity graph is very general

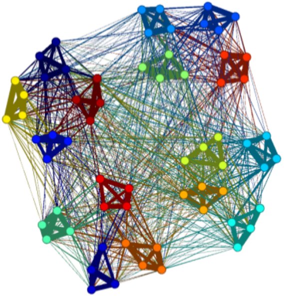
What: community mining



e.g. The PhD students in the same lab

- Many real networks have a modular structure
- Items belonging to the same module (community) have common properties
- The representation of a system through a weighted similarity graph is very general

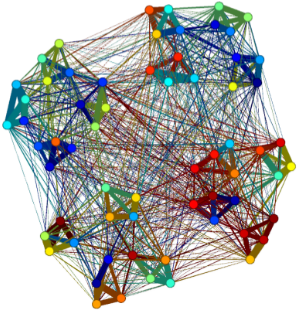
What: community mining



e.g. The PhD students in the same team

- Many real networks have a modular structure
- Items belonging to the same module (community) have common properties
- The representation of a system through a weighted similarity

What: community mining



e.g. The PhD students with the same advisor

- Many real networks have a modular structure
- Items belonging to the same module (community) have common properties
- The representation of a system through a weighted similarity graph is very general

Why: multiscale community mining

- The concept of community is intrinsically connected to its own scale.

Why: multiscale community mining

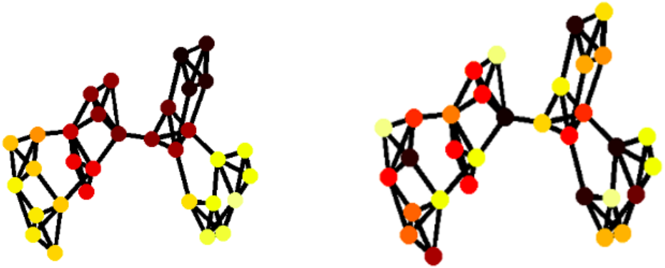
- The concept of community is intrinsically connected to its own scale.
- Most of the existing algorithms (at least back in 2014) impose the scale of the communities and do not allow to choose it.

Why: multiscale community mining

- The concept of community is intrinsically connected to its own scale.
- Most of the existing algorithms (at least back in 2014) impose the scale of the communities and do not allow to choose it.

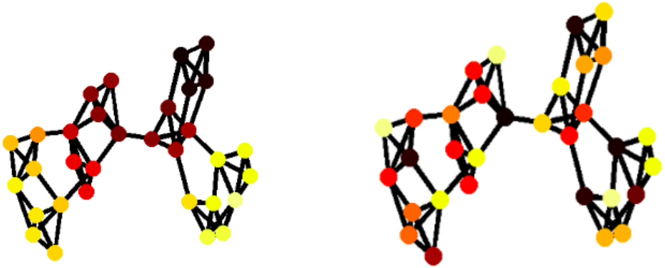
Need of a multiscale approach to community mining

How: graph wavelets



- Centered at a node and spreads over the graph at a given scale

How: graph wavelets



- Centered at a node and spreads over the graph at a given scale
- How the the nodes sees the graph at that scale

Table of Contents

- Spectral clustering
- Graph Fourier transform

Spectral clustering

- Weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$. $|\mathcal{V}| = n$, $A \in \mathbb{R}^{n \times n}$

Spectral clustering

- Weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$. $|\mathcal{V}| = n$, $A \in \mathbb{R}^{n \times n}$
- Diagonal degree matrix: $D_{ii} = d_i = \sum_j A_{ij}$

Spectral clustering

- Weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$. $|\mathcal{V}| = n$, $A \in \mathbb{R}^{n \times n}$
- Diagonal degree matrix: $D_{ii} = d_i = \sum_j A_{ij}$

Graph Laplacian matrix

$$\boldsymbol{L = D - A}$$

Spectral clustering

- Weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$. $|\mathcal{V}| = n$, $A \in \mathbb{R}^{n \times n}$
- Diagonal degree matrix: $D_{ii} = d_i = \sum_j A_{ij}$

Graph Laplacian matrix

$$\boldsymbol{L = D - A}$$

Properties

- $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Spectral clustering

- Weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$. $|\mathcal{V}| = n$, $A \in \mathbb{R}^{n \times n}$
- Diagonal degree matrix: $D_{ii} = d_i = \sum_j A_{ij}$

Graph Laplacian matrix

$$\boldsymbol{L = D - A}$$

Properties

- $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
- $L\mathbf{1}_n = 0$

Spectral clustering

- Weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$. $|\mathcal{V}| = n$, $A \in \mathbb{R}^{n \times n}$
- Diagonal degree matrix: $D_{ii} = d_i = \sum_j A_{ij}$

Graph Laplacian matrix

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$$

Properties

- $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
- $\boldsymbol{L}\mathbf{1}_n = 0$
- $\forall\ \boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} = \frac{1}{2} \sum_{i,j} A_{ij}(x_i - x_j)^2$

Spectral clustering

- Weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$. $|\mathcal{V}| = n$, $A \in \mathbb{R}^{n \times n}$
- Diagonal degree matrix: $D_{ii} = d_i = \sum_j A_{ij}$

Graph Laplacian matrix

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$$

Properties

- $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
- $\boldsymbol{L}\mathbf{1}_n = 0$
- $\forall\ \boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} = \frac{1}{2} \sum_{i,j} A_{ij}(x_i - x_j)^2$
- λ_2 Fiedler eigenvalue: community reconstruction

Spectral clustering

- Weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, A)$.
- Diagonal degree matrix: $D_{ii} = d_i = \sum_j A_{ij}$

Normalized Laplacian matrix

$$\boldsymbol{L}^{\mathrm{sym}} = \boldsymbol{I_n} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$$

Properties

- $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
- $\forall\, \boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x}^T L^{\mathrm{sym}} \boldsymbol{x} = \frac{1}{2} \sum_{i,j} A_{ij} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2$
- λ_2 community reconstruction

Fourier transform (recall)

The eigenvectors $\{\boldsymbol{u}_k\}$ of a matrix R are considered to be graph Fourier modes and the respective eigenvalues $\lambda_k$their associated graph frequency if

Fourier transform (recall)

The eigenvectors $\{\boldsymbol{u}_k\}$ of a matrix R are considered to be graph Fourier modes and the respective eigenvalues $\lambda_k$their associated graph frequency if

- Consistency: R is circulant on the ring graph

Fourier transform (recall)

The eigenvectors $\{\boldsymbol{u}_k\}$ of a matrix R are considered to be graph Fourier modes and the respective eigenvalues $\lambda_k$their associated graph frequency if

- Consistency: R is circulant on the ring graph
- Variational interpretation: λ_k is measures the variation of $\boldsymbol{u}_k$

Fourier transform (recall)

The eigenvectors $\{\boldsymbol{u}_k\}$ of a matrix R are considered to be graph Fourier modes and the respective eigenvalues $\lambda_k$their associated graph frequency if

- Consistency: R is circulant on the ring graph
- Variational interpretation: λ_k is measures the variation of $\boldsymbol{u}_k$

Both L and L^{sym} are a suitable choice for R.

Fourier transform (recall)

The eigenvectors $\{\boldsymbol{u}_k\}$ of a matrix R are considered to be graph Fourier modes and the respective eigenvalues $\lambda_k$their associated graph frequency if

- Consistency: R is circulant on the ring graph
- Variational interpretation: λ_k is measures the variation of $\boldsymbol{u}_k$

Both L and L^{sym} are a suitable choice for R.

Graph Fourier transform

$$\boldsymbol{U} = (\boldsymbol{u}_1|\boldsymbol{u}_2|\dots|\boldsymbol{u}_n) \in \mathbb{R}^{n\times n} \Longrightarrow \hat{\boldsymbol{f}} = \boldsymbol{U}^T\boldsymbol{f}$$

Table of Contents

- Graph wavelet filter
- Adapt the filter to community mining

Graph wavelet filter

We want to create the wavelet $\psi_{s,a}$ of scale s around node a.

Graph wavelet filter

We want to create the wavelet $\psi_{s,a}$ of scale s around node a.
Consider the band-pass kernel filter

$$G_s = \mathrm{diag}(g(s\lambda_1), g(s\lambda_2), \dots, g(s\lambda_n))$$

Graph wavelet filter

We want to create the wavelet $\psi_{s,a}$ of scale s around node a. Consider the band-pass kernel filter

$$G_s = \mathrm{diag}(g(s\lambda_1), g(s\lambda_2), \ldots, g(s\lambda_n))$$

We thus obtain the wavelet basis

$$\Psi_s = UG_sU^T$$

Graph wavelet filter

We want to create the wavelet $\psi_{s,a}$ of scale s around node a. Consider the band-pass kernel filter

$$G_s = \mathrm{diag}(g(s\lambda_1), g(s\lambda_2), \ldots, g(s\lambda_n))$$

We thus obtain the wavelet basis

$$\Psi_s = UG_sU^T$$

that applied on the Dirac gives the node wavelet

$$\boldsymbol{\psi}_{s,a} = UG_sU^T\boldsymbol{\delta}_a$$

Graph wavelet filter

We want to create the wavelet $\psi_{s,a}$ of scale s around node a.
Consider the band-pass kernel filter

$$G_s = \mathrm{diag}(g(s\lambda_1), g(s\lambda_2), \ldots, g(s\lambda_n))$$

We thus obtain the wavelet basis

$$\Psi_s = UG_sU^T$$

that applied on the Dirac gives the node wavelet

$$\boldsymbol{\psi}_{\boldsymbol{s},\boldsymbol{a}} = UG_sU^T\boldsymbol{\delta}_{\boldsymbol{a}}$$

thus the wavelet coefficient of node a at scale s

$$W_f(s,a) = \psi_{\boldsymbol{s},\boldsymbol{a}}^T\boldsymbol{f}$$

Graph wavelet filter

What are we doing?

$$\mathbf{\Psi}_f(s, a) = UG_sU^T\boldsymbol{\delta}_{\boldsymbol{a}}$$

Graph wavelet filter

What are we doing?

$$\boldsymbol{\Psi}_f(s, a) = U G_s U^T \boldsymbol{\delta_a}$$

- Take the Dirac

Graph wavelet filter

What are we doing?

$$\boldsymbol{\Psi}_f(s, a) = UG_s U^T \boldsymbol{\delta}_{\boldsymbol{a}}$$

- Take the Dirac
- Go to Fourier space

Graph wavelet filter

What are we doing?

$$\boldsymbol{\Psi}_f(s, a) = U G_s U^T \boldsymbol{\delta_a}$$

- Take the Dirac
- Go to Fourier space
- Apply the filter and select the scale s

Graph wavelet filter

What are we doing?

$$\Psi_f(s,a) = U G_s U^T \boldsymbol{\delta_a}$$

- Take the Dirac
- Go to Fourier space
- Apply the filter and select the scale s
- Make an inverse Fourier transform

Graph wavelet filter

What are we doing?

$$\boldsymbol{\Psi}_f(s,a) = UG_sU^T\boldsymbol{\delta_a}$$

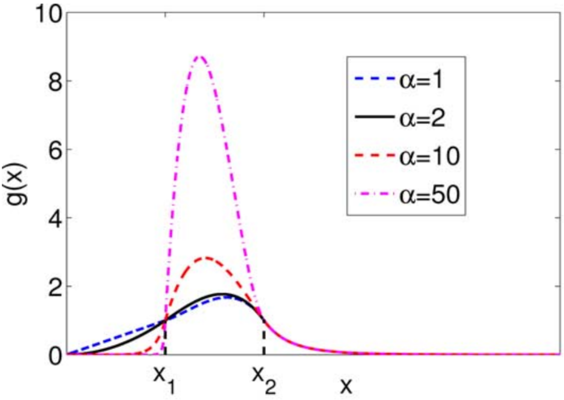
- Take the Dirac
- Go to Fourier space
- Apply the filter and select the scale s
- Make an inverse Fourier transform

$$W_f(s,a) = \psi_{\boldsymbol{s,a}}^{\boldsymbol{T}}\boldsymbol{f}$$

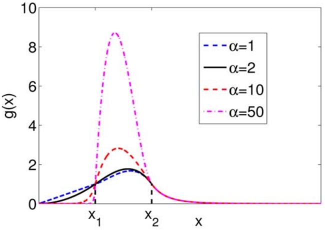
Project the signal on the wavelet

Adapt the filter to community mining

$$g(x;\alpha,\beta,x_1,x_2)=\begin{cases}\left(\frac{x}{x_1}\right)^{\alpha} & \text{for } x\le x_1\\ p(x) & x_1\le x\le x_2\\ \left(\frac{x_2}{x}\right)^{\beta} & \text{for } x\ge x_2\end{cases}$$

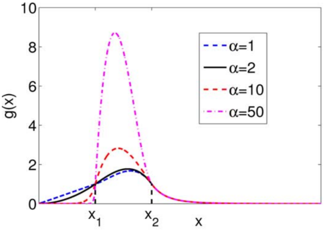


Adapt the filter to community mining



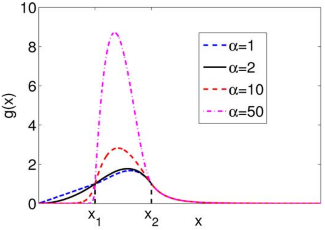
- Choose $s_{\max} = x_2/\lambda_2$: $g(s_{\max}x)$ decays after λ_2.

Adapt the filter to community mining



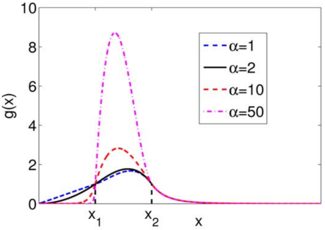
- Choose $s_{\mathrm{max}} = x_2/\lambda_2$: $g(s_{\mathrm{max}}x)$ decays after λ_2.
- $\beta = 1/\log_{10}(\lambda_3/\lambda_2)$: attenuate the modes beyond λ_2, $g(s_{\mathrm{max}}\lambda_2) = 10g(s_{\mathrm{max}}\lambda_3)$.

Adapt the filter to community mining



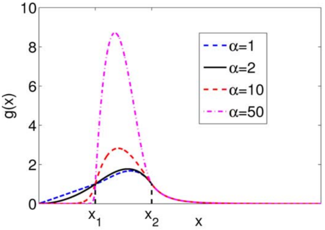
- Choose $s_{\max} = x_2/\lambda_2$: $g(s_{\max}x)$ decays after λ_2.
- $\beta = 1/\log_{10}(\lambda_3/\lambda_2)$: attenuate the modes beyond λ_2, $g(s_{\max}\lambda_2) = 10g(s_{\max}\lambda_3)$.
- $s_{\min} = x_1/\lambda_2$: $g(s\lambda_2) > 1$ for $s_{\min} \leq s \leq s_{\max}$

Adapt the filter to community mining



- Choose $s_{\max} = x_2/\lambda_2$: $g(s_{\max}x)$ decays after λ_2.
- $\beta = 1/\log_{10}(\lambda_3/\lambda_2)$: attenuate the modes beyond λ_2, $g(s_{\max}\lambda_2) = 10g(s_{\max}\lambda_3)$.
- $s_{\min} = x_1/\lambda_2$: $g(s\lambda_2) > 1$ for $s_{\min} \leq s \leq s_{\max}$
- $\alpha = 2$: it only determines the selectivity of the filter

Adapt the filter to community mining



- Choose $s_{\max} = x_2/\lambda_2$: $g(s_{\max}x)$ decays after λ_2.
- $\beta = 1/\log_{10}(\lambda_3/\lambda_2)$: attenuate the modes beyond λ_2, $g(s_{\max}\lambda_2) = 10g(s_{\max}\lambda_3)$.
- $s_{\min} = x_1/\lambda_2$: $g(s\lambda_2) > 1$ for $s_{\min} \leq s \leq s_{\max}$
- $\alpha = 2$: it only determines the selectivity of the filter

Fixing $x_1 = 1$, we finally obtain

$$s_{\min} = \frac{1}{\lambda_2}, \quad x_2 = \frac{1}{\lambda_2}, \quad s_{\max} = \frac{1}{\lambda_2^2}$$

Table of Contents

- Identifying the communities
- Fast community mining
- What are the most relevant scales?
- How relevant are the most relevant scales?

Identifying the communities

- We create a vector $\boldsymbol{s}$ between $s_{\min}$ and $s_{\max}$ with $M \sim \log(n)$ elements, logarithmically spaced.

Identifying the communities

- We create a vector $\boldsymbol{s}$ between $s_{\min}$ and $s_{\max}$ with $M \sim \log(n)$ elements, logarithmically spaced.
- For each s and each node, create the wavelet $\boldsymbol{\psi}_{\boldsymbol{s},\boldsymbol{a}}$.

Identifying the communities

- We create a vector $\boldsymbol{s}$ between $s_{\min}$ and $s_{\max}$ with $M \sim \log(n)$ elements, logarithmically spaced.
- For each s and each node, create the wavelet $\boldsymbol{\psi}_{\boldsymbol{s},\boldsymbol{a}}$.
- Create the correlation distance matrix $D_s(a,b) = 1 - \frac{\boldsymbol{\psi}_{s,a}^T \boldsymbol{\psi}_{s,b}}{\|\boldsymbol{\psi}_{s,a}\|_2 \|\boldsymbol{\psi}_{s,b}\|_2}$

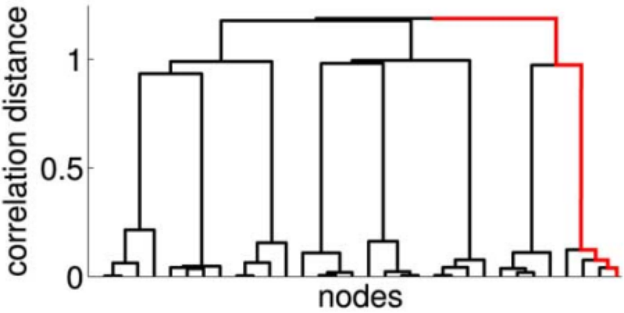
Identifying the communities

- We create a vector $\boldsymbol{s}$ between $s_{\min}$ and $s_{\max}$ with $M \sim \log(n)$ elements, logarithmically spaced.
- For each s and each node, create the wavelet $\boldsymbol{\psi_{s,a}}$.
- Create the correlation distance matrix $D_s(a,b) = 1 - \frac{\boldsymbol{\psi}_{s,a}^T \boldsymbol{\psi}_{s,b}}{\|\boldsymbol{\psi}_{s,a}\|_2 \|\boldsymbol{\psi}_{s,b}\|_2}$
- Use a hierarchical average linkage algorithm on D_s

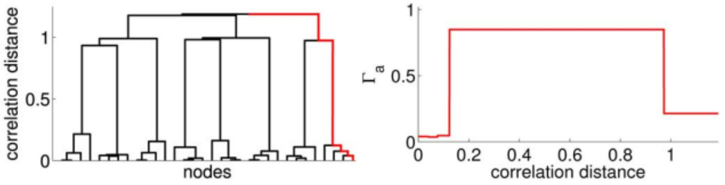
Identifying the communities

- We create a vector $\boldsymbol{s}$ between $s_{\min}$ and $s_{\max}$ with $M \sim \log(n)$ elements, logarithmically spaced.
- For each s and each node, create the wavelet $\psi_{\boldsymbol{s},\boldsymbol{a}}$.
- Create the correlation distance matrix $D_s(a,b) = 1 - \frac{\psi_{s,a}^T \psi_{s,b}}{\|\psi_{s,a}\|_2 \|\psi_{s,b}\|_2}$
- Use a hierarchical average linkage algorithm on D_s

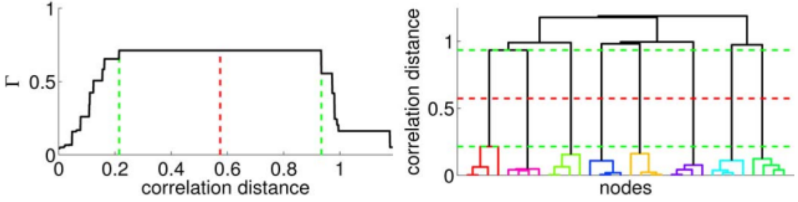
We obtain a dendogram



Identifying the communities



For each node a create the function Γ_a (why is decreasing after one? How is it defined?)



Average Γ_a over all a and obtain Γ. Cut at the maximum of Γ. Repeat for all s and obtain $\{\mathcal{P}_s\}$.

Fast community mining

As things stand now

Fast community mining

As things stand now

- Compute all the $\boldsymbol{\psi}_{s,a} = UG_sU^T\boldsymbol{\delta}_a \in \mathbb{R}^n$

Fast community mining

As things stand now

- Compute all the $\boldsymbol{\psi}_{s,a} = UG_s U^T \boldsymbol{\delta}_a \in \mathbb{R}^n$
- From this compute the matrix D_s

Fast community mining

As things stand now

- Compute all the $\boldsymbol{\psi}_{s,a} = UG_s U^T \boldsymbol{\delta}_a \in \mathbb{R}^n$
- From this compute the matrix D_s

... not so fast.

Fast community mining

As things stand now

- Compute all the $\boldsymbol{\psi}_{s,a} = UG_sU^T\boldsymbol{\delta}_a \in \mathbb{R}^n$
- From this compute the matrix D_s

... not so fast. Consider the following procedure:

Fast community mining

As things stand now

- Compute all the $\boldsymbol{\psi}_{s,a} = UG_s U^T \boldsymbol{\delta}_a \in \mathbb{R}^n$
- From this compute the matrix D_s

... not so fast. Consider the following procedure:

- $R = (\boldsymbol{r}_1 | \boldsymbol{r}_2 | \dots | \boldsymbol{r}_\eta) \in \mathbb{R}^{n \times \eta}$ vectors of zero mean, unit variance i.i.d. Gaussian r.v.

Fast community mining

As things stand now

- Compute all the $\boldsymbol{\psi}_{s,a} = UG_s U^T \boldsymbol{\delta}_a \in \mathbb{R}^n$
- From this compute the matrix D_s

... not so fast. Consider the following procedure:

- $R = (\boldsymbol{r}_1|\boldsymbol{r}_2|\ldots|\boldsymbol{r}_\eta) \in \mathbb{R}^{n\times\eta}$ vectors of zero mean, unit variance i.i.d. Gaussian r.v.
- Compute $\boldsymbol{f}_{s,a}^T = \boldsymbol{\psi}_{s,a}^T R \ \in \mathbb{R}^\eta$

Fast community mining

As things stand now

- Compute all the $\boldsymbol{\psi}_{s,a} = UG_s U^T \boldsymbol{\delta}_a \in \mathbb{R}^n$
- From this compute the matrix D_s

... not so fast. Consider the following procedure:

- $R = (\boldsymbol{r}_1 | \boldsymbol{r}_2 | \ldots | \boldsymbol{r}_\eta) \in \mathbb{R}^{n \times \eta}$ vectors of zero mean, unit variance i.i.d. Gaussian r.v.
- Compute $\boldsymbol{f}_{s,a}^T = \boldsymbol{\psi}_{s,a}^T R \ \in \mathbb{R}^\eta$
- Define $\hat{C}_{ab,\eta} = \mathrm{corr}(f_{s,a}, f_{s,b})$

Fast community mining

As things stand now

- Compute all the $\boldsymbol{\psi}_{s,a} = UG_s U^T \boldsymbol{\delta}_a \in \mathbb{R}^n$
- From this compute the matrix D_s

... not so fast. Consider the following procedure:

- $R = (\boldsymbol{r}_1 | \boldsymbol{r}_2 | \dots | \boldsymbol{r}_\eta) \in \mathbb{R}^{n \times \eta}$ vectors of zero mean, unit variance i.i.d. Gaussian r.v.
- Compute $\boldsymbol{f}_{s,a}^T = \boldsymbol{\psi}_{s,a}^T R \ \in \mathbb{R}^\eta$
- Define $\hat{C}_{ab,\eta} = \mathrm{corr}(f_{s,a}, f_{s,b})$
- For η sufficiently large $\hat{C}_{ab,\eta} \to 1 - D_s(a, b)$

Fast community mining

As things stand now

- Compute all the $\boldsymbol{\psi}_{s,a} = UG_s U^T \boldsymbol{\delta}_a \in \mathbb{R}^n$
- From this compute the matrix D_s

... not so fast. Consider the following procedure:

- $R = (\boldsymbol{r}_1 | \boldsymbol{r}_2 | \dots | \boldsymbol{r}_\eta) \in \mathbb{R}^{n \times \eta}$ vectors of zero mean, unit variance i.i.d. Gaussian r.v.
- Compute $\boldsymbol{f}_{s,a}^T = \boldsymbol{\psi}_{s,a}^T R \ \in \mathbb{R}^\eta$
- Define $\hat{C}_{ab,\eta} = \mathrm{corr}(f_{s,a}, f_{s,b})$
- For η sufficiently large $\hat{C}_{ab,\eta} \to 1 - D_s(a,b)$
- With Chebishev polynomial approximation, $\boldsymbol{f}_{s,a}$ can be computed efficiently.

$$UG_s U^T \boldsymbol{r} = U \left(\sum_{k=1}^{p} \alpha_k \Lambda_k^k \right) U^T \boldsymbol{r} = \sum_{k=1}^{p} \alpha_k L^k \boldsymbol{r}$$

What are the most relevant scales?

Consider $J(=20)$ sets of η random signals. Obtain the set of partitions $\{\mathcal{P}_s^j\}_{j=1,\dots,J}$.

$$\gamma(s) = \frac{2}{J(J-1)} \sum_{i \neq j} \texttt{ari}(P_s^i, P_s^j)$$

The closer $\gamma(s)$ is to one, the more stable is the partition.

How relevant are the most relevant scales?

- Generate a series of null graphs with the same degree distribution

How relevant are the most relevant scales?

- Generate a series of null graphs with the same degree distribution
- For each, compute $\gamma_0(s)$ and obtain the empirical distribution

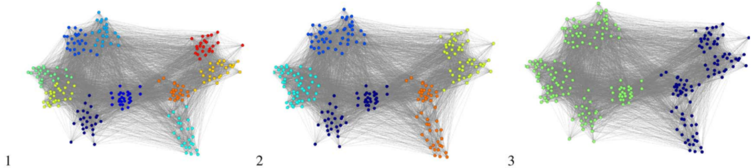
How relevant are the most relevant scales?

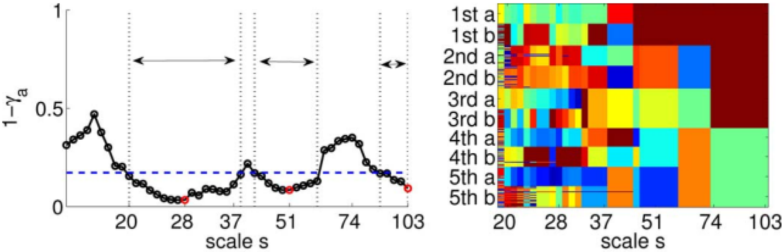
- Generate a series of null graphs with the same degree distribution
- For each, compute $\gamma_0(s)$ and obtain the empirical distribution
- Reject the configurations with $\gamma(s)$ to close to $\gamma_0(s)$

Table of Contents

An example

A Graph of Social Interactions Between Children in a Primary School





Reference

- Tremblay, Nicolas, and Pierre Borgnat. "Graph wavelets for multiscale community mining." IEEE Transactions on Signal Processing 62.20 (2014): 5227-5239.

Reference

- Tremblay, Nicolas, and Pierre Borgnat. "Graph wavelets for multiscale community mining." IEEE Transactions on Signal Processing 62.20 (2014): 5227-5239.

THANKS