Detecção de comunidades I

Métodos mono-camada

Autor: Alan Piovesana
Orientador: José Antônio Brum

Instituto de Física Gleb Wataghin
UNICAMP

13 de março de 2021

Índice

Introdução

Motivação

Geometric renormalization unravels self-similarity of the multiscale human connectome

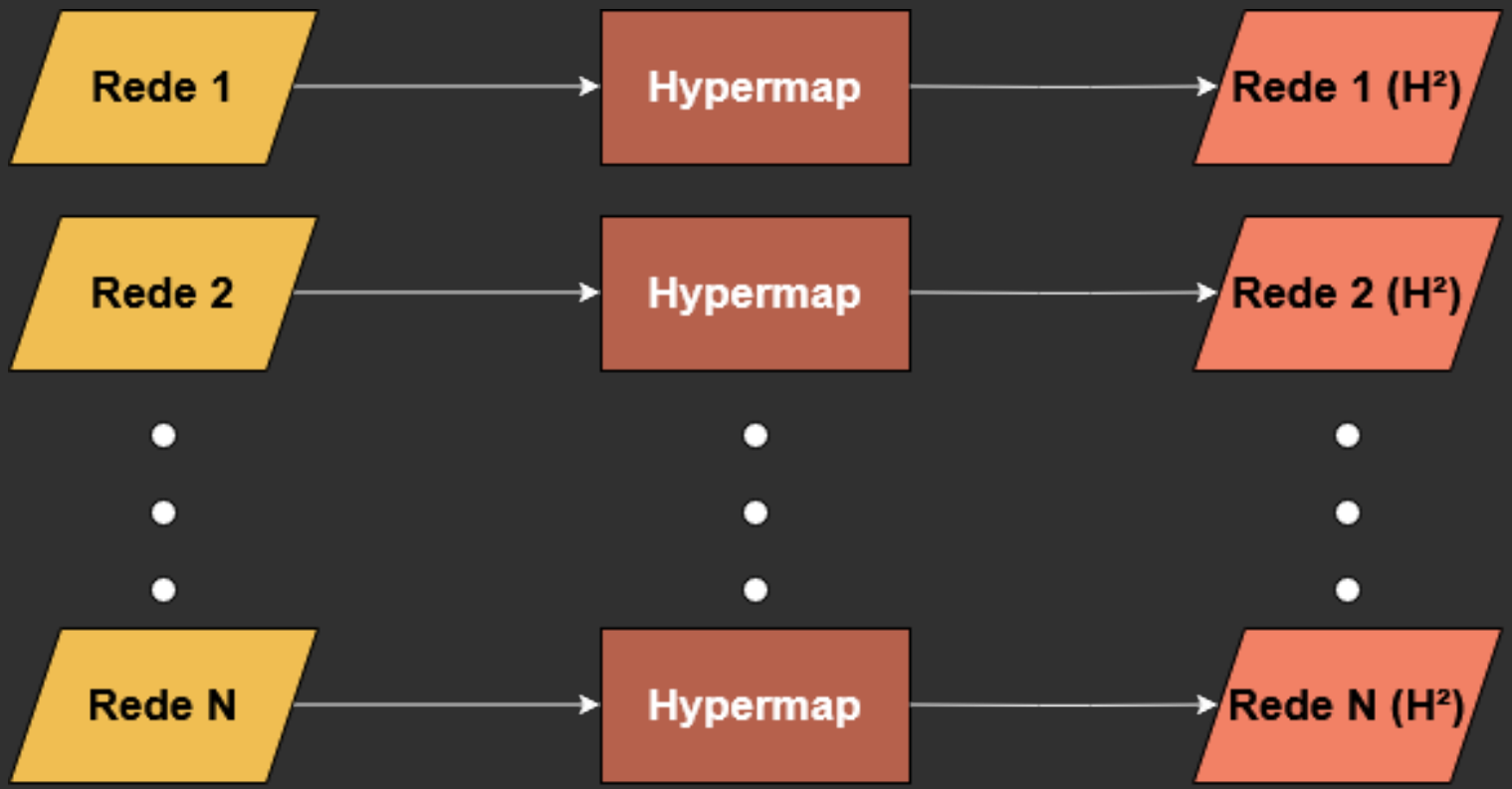
Muhua Zheng,[1,2] Antoine Allard,[3,4] Patric Hagmann,[5] and M. Ángeles Serrano[1,2,6,*]

[1] *Departament de Física de la Matèria Condensada, Universitat de Barcelona, Martí i Franquès 1, E-08028 Barcelona, Spain*
[2] *Universitat de Barcelona Institute of Complex Systems (UBICS), Universitat de Barcelona, Barcelona, Spain*
[3] *Département de physique, de génie physique et d'optique, Université Laval, Québec , Canada G1V 0A6*
[4] *Centre interdisciplinaire de modélisation mathématique, Université Laval, Québec, Canada G1V 0A6*
[5] *Department of Radiology, Lausanne University Hospital (CHUV) and University of Lausanne (UNIL), Lausanne, Switzerland*
[6] *ICREA, Passeig Lluís Companys 23, E-08010 Barcelona, Spain*

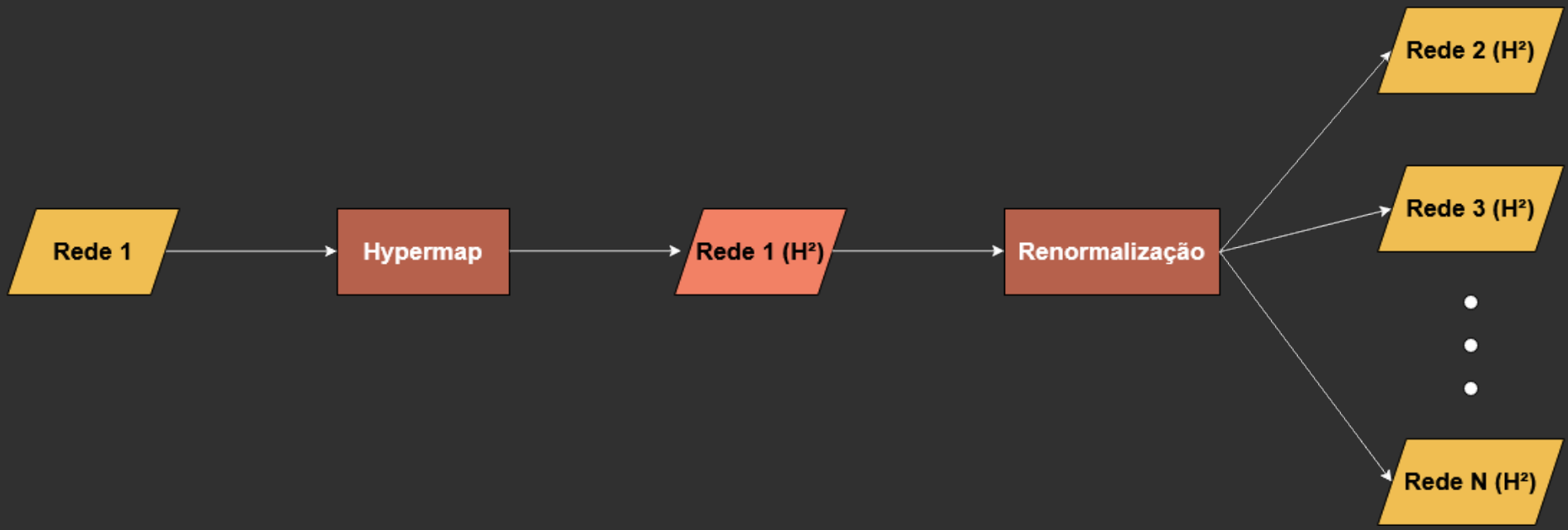
(Dated: April 29, 2019)

Structural connectivity in the brain is typically studied by reducing its observation to a single spatial resolution. However, the brain poses a rich architecture organized in multiple scales linked to one another. We explored the multiscale organization of the human connectome using a dataset of healthy subjects reconstructed at five different resolutions. We find that the structure of the human brain remains self-similar when the resolution length is progressively decreased by hierarchical coarse-graining of the anatomical regions. Strikingly, geometric renormalization of connectome maps in hyperbolic space, which decreases the resolution by coarse-graining and averaging over short similarity distances, predicts the properties of connectomes including self-similarity. Our results suggest that the same principles regulate connectivity between brain regions at different length scales and that the multiscale self-similarity of brain connectomes may offer an advantageous architecture for navigation purposes. The implications are varied and can affect fundamental debates, like whether the brain is working near a critical point, and lead to applications including advanced tools to simplify the digital reconstruction and simulation of the brain.

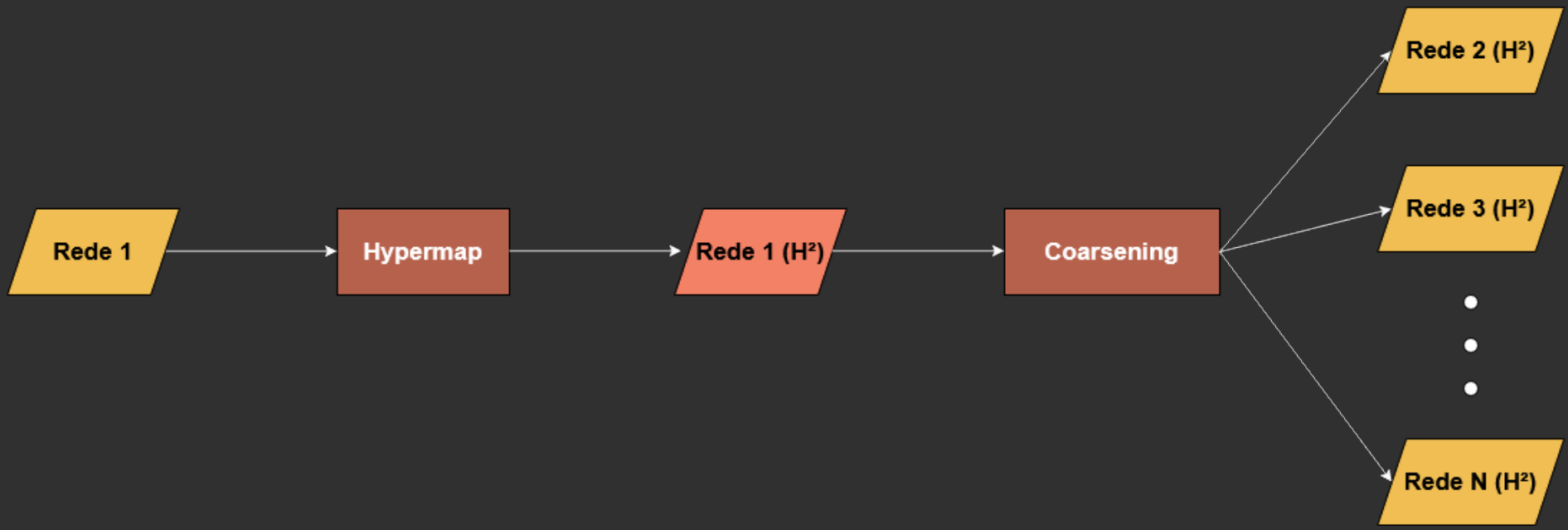
Estratégia do projeto - Serrano et. al.



Estratégia do projeto - Serrano et. al.



Estratégia do projeto - Nossa abordagem



Detecção de comunidades - o que é?

Modularidade

Benchmarking

Modularidade:

Q = (% dos links intra-comunidades/total) - (% esperado)

$$Q = \frac{1}{L} \sum_{\mathcal{C} \in \mathcal{P}} \sum_{i,j \in \mathcal{C}} [A_{ij} - P_{ij}]$$

com

$$P_{ij} = \frac{\langle k \rangle^2}{2m}$$

ou

$$P_{ij} = \frac{k_i k_j}{2m}$$

Benchmarking

Modularidade:

Q = (% dos links intra-comunidades/total) - (% esperado)

$$Q = \frac{1}{L} \sum_{C \in \mathcal{P}} \sum_{i,j \in C} [A_{ij} - P_{ij}]$$

com

$$P_{ij} = \frac{\langle k \rangle^2}{2m}$$

ou

$$P_{ij} = \frac{k_i k_j}{2m}$$

Benchmarking

Modularidade:

Q = (% dos links intra-comunidades/total) - (% esperado)

$$Q = \frac{1}{L} \sum_{C \in \mathcal{P}} \sum_{i,j \in C} [A_{ij} - P_{ij}]$$

com

$$P_{ij} = \frac{\langle k \rangle^2}{2m}$$

ou

$$P_{ij} = \frac{k_i k_j}{2m}$$

Maximização da modularidade - NP-Completo

Maximizing Modularity is hard*

Ulrik Brandes[1], Daniel Delling[2], Marco Gaertler[2], Robert Görke[2], Martin Hoefer[1**], Zoran Nikoloski[3], and Dorothea Wagner[2]

[1] Department of Computer & Information Science, University of Konstanz, Germany
[2] Faculty of Informatics, Universität Karlsruhe (TH), Germany
[3] Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

Abstract. Several algorithms have been proposed to compute partitions of networks into communities that score high on a graph clustering index called modularity. While publications on these algorithms typically contain experimental evaluations to emphasize the plausibility of results, none of these algorithms has been shown to actually compute optimal partitions. We here settle the unknown complexity status of modularity maximization by showing that the corresponding decision version is NP-complete in the strong sense. As a consequence, any efficient, i.e. polynomial-time, algorithm is only heuristic and yields suboptimal partitions on many instances.

Método Espectral de Newman

Artigo

Modularity and community structure in networks

M. E. J. Newman
Department of Physics and Center for the Study of Complex Systems, Randall Laboratory, University of Michigan, Ann Arbor, MI 48109–1040

Many networks of interest in the sciences, including a variety of social and biological networks, are found to divide naturally into communities or modules. The problem of detecting and characterizing this community structure has attracted considerable recent attention. One of the most sensitive detection methods is optimization of the quality function known as "modularity" over the possible divisions of a network, but direct application of this method using, for instance, simulated annealing is computationally costly. Here we show that the modularity can be reformulated in terms of the eigenvectors of a new characteristic matrix for the network, which we call the modularity matrix, and that this reformulation leads to a spectral algorithm for community detection that returns results of better quality than competing methods in noticeably shorter running times. We demonstrate the algorithm with applications to several network data sets.

Caso simples: 2 comunidades

$$s_i = \begin{cases} +1, & se \quad i \in Grupo\ 1 \\ -1, & se \quad i \in Grupo\ 2 \end{cases}$$

$$Q = \frac{1}{4m} \sum_{ij} s_i \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_j = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s}$$

com

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

Caso simples: 2 comunidades

$$s_i = \begin{cases} +1, & se \quad i \in Grupo\ 1 \\ -1, & se \quad i \in Grupo\ 2 \end{cases}$$

$$Q = \frac{1}{4m} \sum_{ij} s_i \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_j = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s}$$

com

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

Caso simples: 2 comunidades

$$s_i = \begin{cases} +1, & se \quad i \in Grupo\ 1 \\ -1, & se \quad i \in Grupo\ 2 \end{cases}$$

$$Q = \frac{1}{4m} \sum_{ij} s_i \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_j = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s}$$

com

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

Caso simples: 2 comunidades

Expandindo $\vec{s}$ nos autovetores $\vec{u}_i$ de $\mathbf{B}$:

$$\mathbf{s} = \sum_{i=1}^{n} a_i \vec{u}_i \implies a_i = \vec{u}_i^T \cdot \vec{s}$$

$$Q = \left(\sum_i a_i \vec{u}_i \right)^T \mathbf{B} \left(\sum_j a_j \vec{u}_j \right)$$

$$\Downarrow$$

$$Q = \sum_{ij} a_i a_j \beta_j \underbrace{\vec{u}_i^T \vec{u}_j}_{\delta_{ij}} = \boxed{\sum_i \left(\vec{u}_i^T \cdot \vec{s} \right)^2 \beta_i}$$

Caso simples: 2 comunidades

Expandindo $\vec{s}$ nos autovetores $\vec{u}_i$ de $\mathbf{B}$:

$$\mathbf{s} = \sum_{i=1}^{n} a_i \vec{u}_i \implies a_i = \vec{u}_i^T \cdot \vec{s}$$

$$Q = \left(\sum_i a_i \vec{u}_i \right)^T \mathbf{B} \left(\sum_j a_j \vec{u}_j \right)$$

$$\Downarrow$$

$$Q = \sum_{ij} a_i a_j \beta_j \underbrace{\vec{u}_i^T \vec{u}_j}_{\delta_{ij}} = \boxed{\sum_i \left(\vec{u}_i^T \cdot \vec{s} \right)^2 \beta_i}$$

Caso simples: 2 comunidades

Expandindo $\vec{s}$ nos autovetores $\vec{u}_i$ de $\mathbf{B}$:

$$\mathbf{s} = \sum_{i=1}^{n} a_i \vec{u}_i \implies a_i = \vec{u}_i^T \cdot \vec{s}$$

$$Q = \left(\sum_i a_i \vec{u}_i \right)^T \mathbf{B} \left(\sum_j a_j \vec{u}_j \right)$$

$$\Downarrow$$

$$Q = \sum_{ij} a_i a_j \beta_j \underbrace{\vec{u}_i^T \vec{u}_j}_{\delta_{ij}} = \boxed{\sum_i \left(\vec{u}_i^T \cdot \vec{s} \right)^2 \beta_i}$$

Novamente: MAXQ é NP-Completo!

Solução não-ótima:

Sinal de s_i = sinal de $u_1^{(i)}$

Algoritmo

1. Calcule o autovetor associado a β_1
2. Divida os nós seguindo o sinal correspondente em $\vec{u}_1$
3. Repita o processo para cada componente
4. Pare caso a próxima divisão gere $\Delta Q \leq 0$

Algoritmo

1. Calcule o autovetor associado a β_1
2. Divida os nós seguindo o sinal correspondente em $\vec{u}_1$
3. Repita o processo para cada componente
4. Pare caso a próxima divisão gere $\Delta Q \leq 0$

Algoritmo

1. Calcule o autovetor associado a β_1
2. Divida os nós seguindo o sinal correspondente em $\vec{u}_1$
3. Repita o processo para cada componente
4. Pare caso a próxima divisão gere $\Delta Q \leq 0$

Algoritmo

1. Calcule o autovetor associado a β_1
2. Divida os nós seguindo o sinal correspondente em $\vec{u}_1$
3. Repita o processo para cada componente
4. Pare caso a próxima divisão gere $\Delta Q \leq 0$

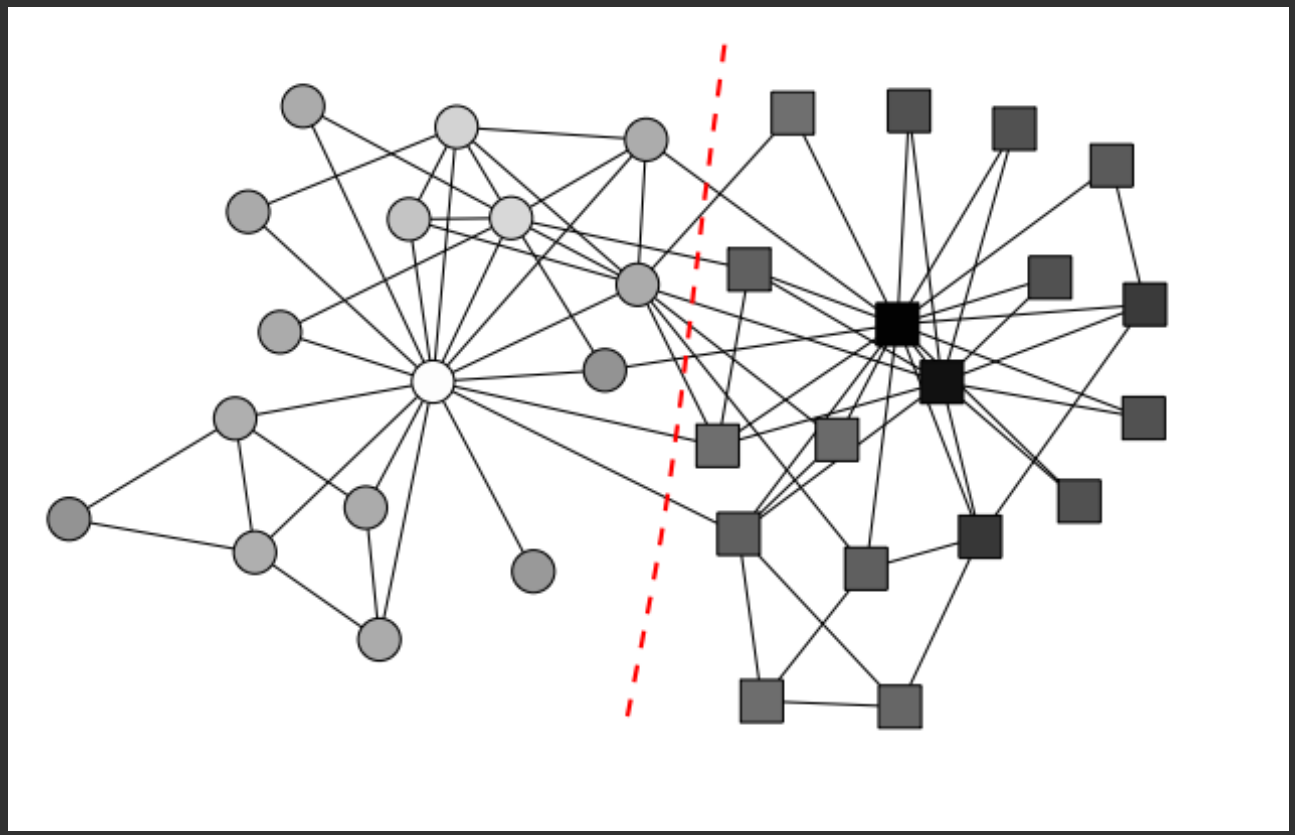
Vantagens

- A partição trivial $(1, 1, 1, ...)$ tem autovalor zero
- Magnitude de $u_1^{(i)}$ é uma métrica de "pertencimento"

Vantagens

- A partição trivial $(1, 1, 1, ...)$ tem autovalor zero
- Magnitude de $u_1^{(i)}$ é uma métrica de "pertencimento"

Magnitude das entradas - pertencimento



Desvantagens

- Não garante ótimos globais
- Resolver problema de autovalores
- Algoritmo mais lento - complexidade estimada: $\mathcal{O}(n^2 log n)$

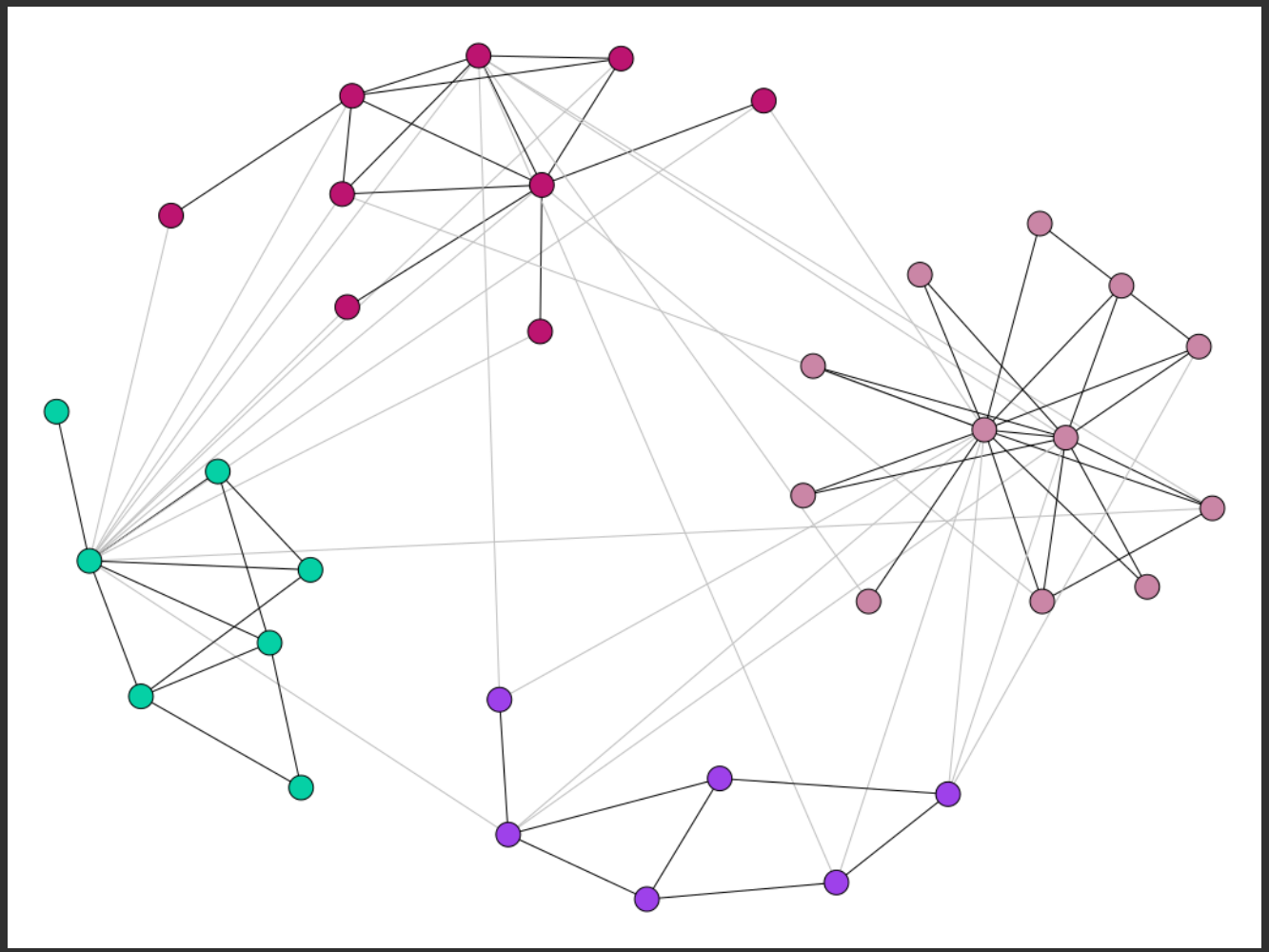
Desvantagens

- Não garante ótimos globais
- Resolver problema de autovalores
- Algoritmo mais lento - complexidade estimada: $\mathcal{O}(n^2 log n)$

Desvantagens

- Não garante ótimos globais
- Resolver problema de autovalores
- Algoritmo mais lento - complexidade estimada: $\mathcal{O}(n^2 log n)$

Simulações



Método de Louvain

Artigo

Fast unfolding of communities in large networks

Vincent D. Blondel[1;a], Jean-Loup Guillaume[1,2;b], Renaud Lambiotte[1,3;c] and Etienne Lefebvre[1]

[1]Department of Mathematical Engineering, Université catholique de Louvain, 4 avenue Georges Lemaitre, B-1348 Louvain-la-Neuve, Belgium
[2] LIP6, Université Pierre et Marie Curie, 4 place Jussieu, 75005 Paris, France
[3] Institute for Mathematical Sciences, Imperial College London, 53 Prince's Gate, South Kensington campus, SW72PG, UK

E-mail: [a]vincent.blondel@uclouvain.be; [b]jean-loup.guillaume@lip6.fr; [c]r.lambiotte@imperial.ac.uk;

Abstract. We propose a simple method to extract the community structure of large networks. Our method is a heuristic method that is based on modularity optimization. It is shown to outperform all other known community detection method in terms of computation time. Moreover, the quality of the communities detected is very good, as measured by the so-called modularity. This is shown first by identifying language communities in a Belgian mobile phone network of 2.6 million customers and by analyzing a web graph of 118 million nodes and more than one billion links. The accuracy of our algorithm is also verified on ad-hoc modular networks.

Algoritmo

1. Escolha um nó e tente agregar a algum cluster $\Leftarrow$ "greedy step"
2. **Escolha o cluster que maximizar ΔQ**
3. Itere em todos os nós
4. Use cada comunidade formada como um super-nó e recomece
5. Pare caso $\Delta Q \leq 0$

Algoritmo

1. Escolha um nó e tente agregar a algum cluster $\Leftarrow$ "greedy step"
2. **Escolha o cluster que maximizar ΔQ**
3. Itere em todos os nós
4. Use cada comunidade formada como um super-nó e recomece
5. Pare caso $\Delta Q \leq 0$

Algoritmo

1. Escolha um nó e tente agregar a algum cluster $\Leftarrow$ "greedy step"
2. **Escolha o cluster que maximizar ΔQ**
3. Itere em todos os nós
4. Use cada comunidade formada como um super-nó e recomece
5. Pare caso $\Delta Q \leq 0$

Algoritmo

1. Escolha um nó e tente agregar a algum cluster $\Leftarrow$ "greedy step"
2. **Escolha o cluster que maximizar ΔQ**
3. Itere em todos os nós
4. Use cada comunidade formada como um super-nó e recomece
5. Pare caso $\Delta Q \leq 0$

Algoritmo

1. Escolha um nó e tente agregar a algum cluster $\Leftarrow$ "greedy step"
2. **Escolha o cluster que maximizar ΔQ**
3. Itere em todos os nós
4. Use cada comunidade formada como um super-nó e recomece
5. Pare caso $\Delta Q \leq 0$

Algoritmo

1. Escolha um nó e tente agregar a algum cluster $\Leftarrow$ "greedy step"
2. **Escolha o cluster que maximizar ΔQ**
3. Itere em todos os nós
4. Use cada comunidade formada como um super-nó e recomece
5. Pare caso $\Delta Q \leq 0$

Variação de modularidade

$$Q = \sum_{C_i \in \mathcal{C}} \left[\frac{k_{in}^{C_i}}{2m} - \frac{(k_{tot}^{C_i})^2}{4m^2} \right]$$

$$\Downarrow$$

$$\Delta Q = \left[\frac{k_{in}^{C_j} + 2k_{in}^{i}}{2m} - \frac{\left(k_{tot}^{C_j} + k_i\right)^2}{4m^2} \right] - \left[\left(\frac{k_{in}^{C_j}}{2m} - \frac{(k_{tot}^{C_j})^2}{4m^2} \right) + \left(0 - \frac{k_i^2}{4m^2} \right) \right]$$

Variação de modularidade

$$Q = \sum_{C_i \in \mathcal{C}} \left[\frac{k_{in}^{C_i}}{2m} - \frac{(k_{tot}^{C_i})^2}{4m^2} \right]$$

$$\Downarrow$$

$$\Delta Q = \left[\frac{k_{in}^{C_j} + 2k_{in}^{i}}{2m} - \frac{\left(k_{tot}^{C_j} + k_i\right)^2}{4m^2} \right] - \left[\left(\frac{k_{in}^{C_j}}{2m} - \frac{(k_{tot}^{C_j})^2}{4m^2} \right) + \left(0 - \frac{k_i^2}{4m^2} \right) \right]$$

Vantagens

- Calcular ΔQ pela fórmula é rápido
- Não computa autovetores - complexidade estimada: $\mathcal{O}(nlog^2n)$
- Caráter hierárquico multiescala

Vantagens

- Calcular ΔQ pela fórmula é rápido
- Não computa autovetores - complexidade estimada: $\mathcal{O}(nlog^2 n)$
- Caráter hierárquico multiescala

Vantagens

- Calcular ΔQ pela fórmula é rápido
- Não computa autovetores - complexidade estimada: $\mathcal{O}(nlog^2n)$
- Caráter hierárquico multiescala

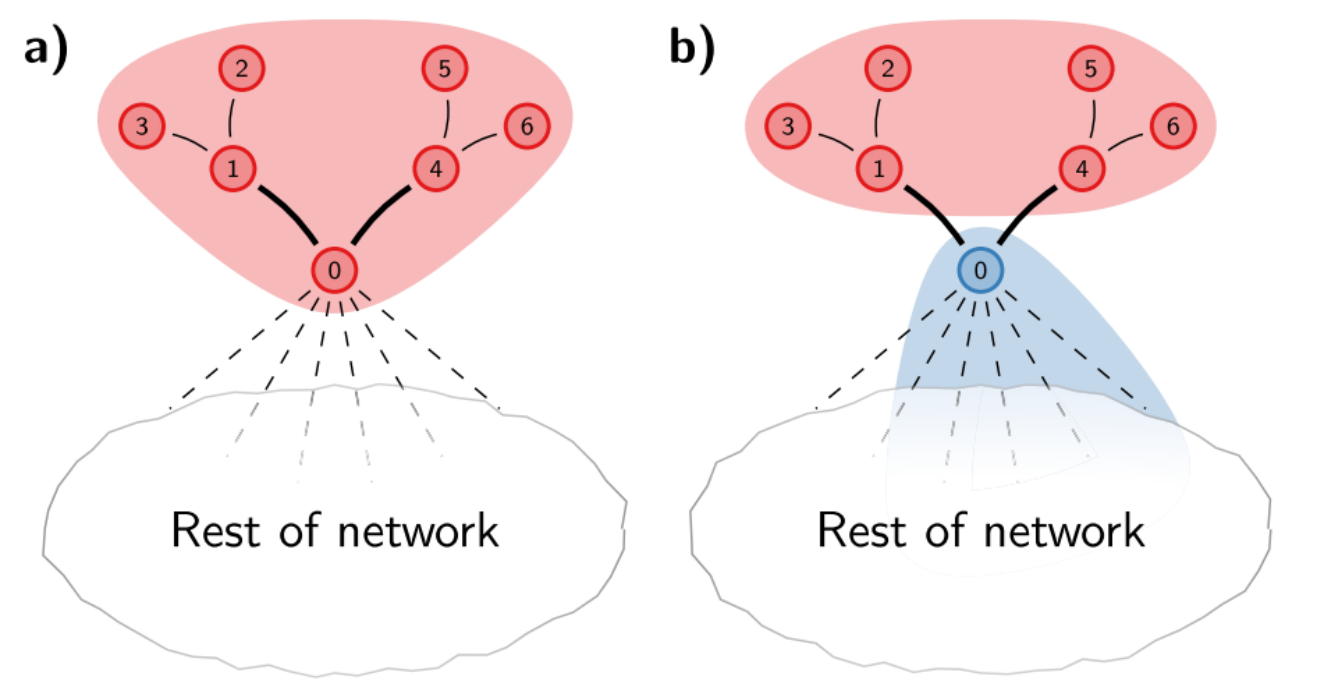
Desvantagens

- Limite de granularidade: grupos não-independentes ou não-mínimos
- Pode agrupar comunidades disjuntas

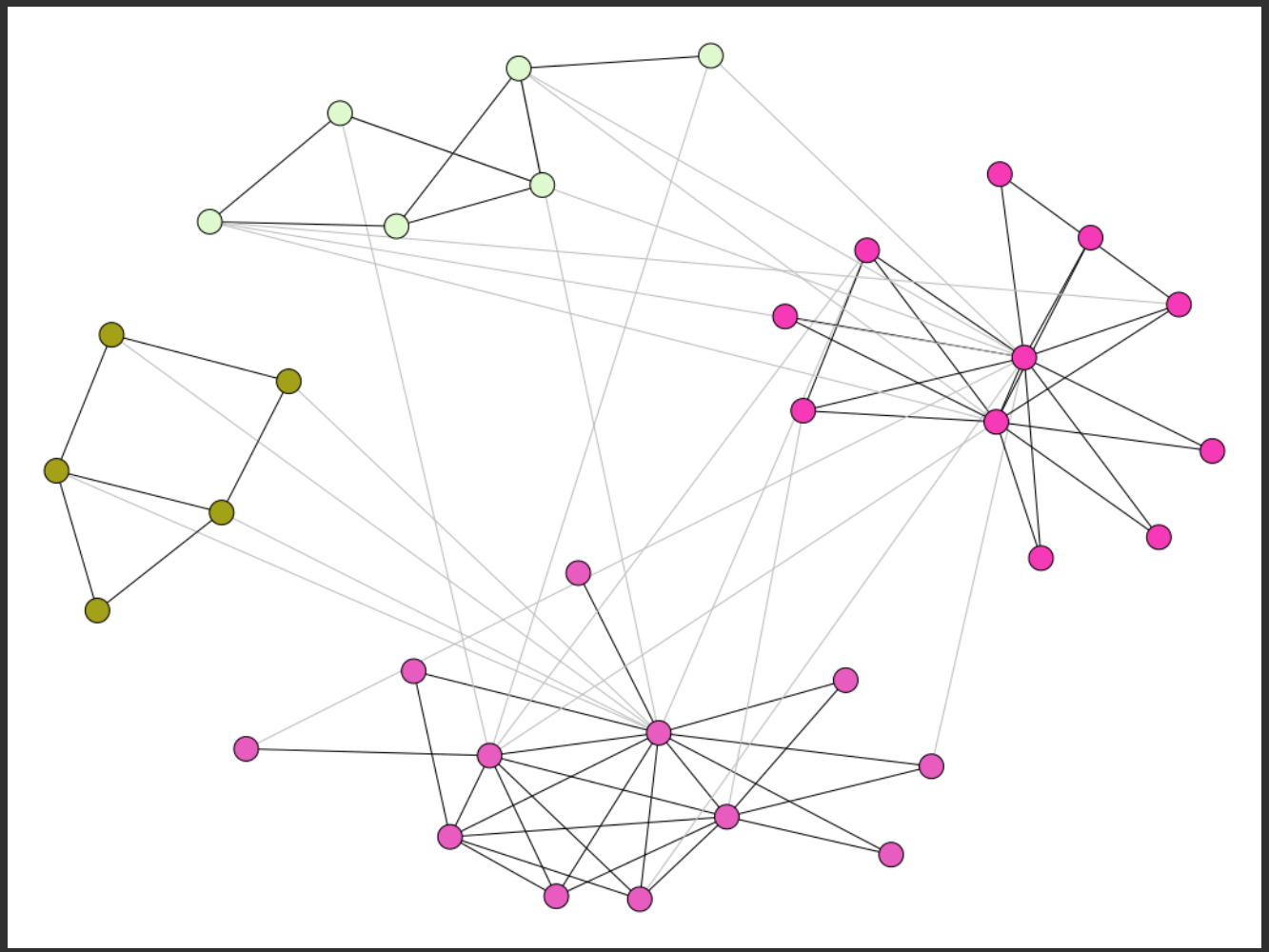
Desvantagens

- Limite de granularidade: grupos não-independentes ou não-mínimos
- Pode agrupar comunidades disjuntas

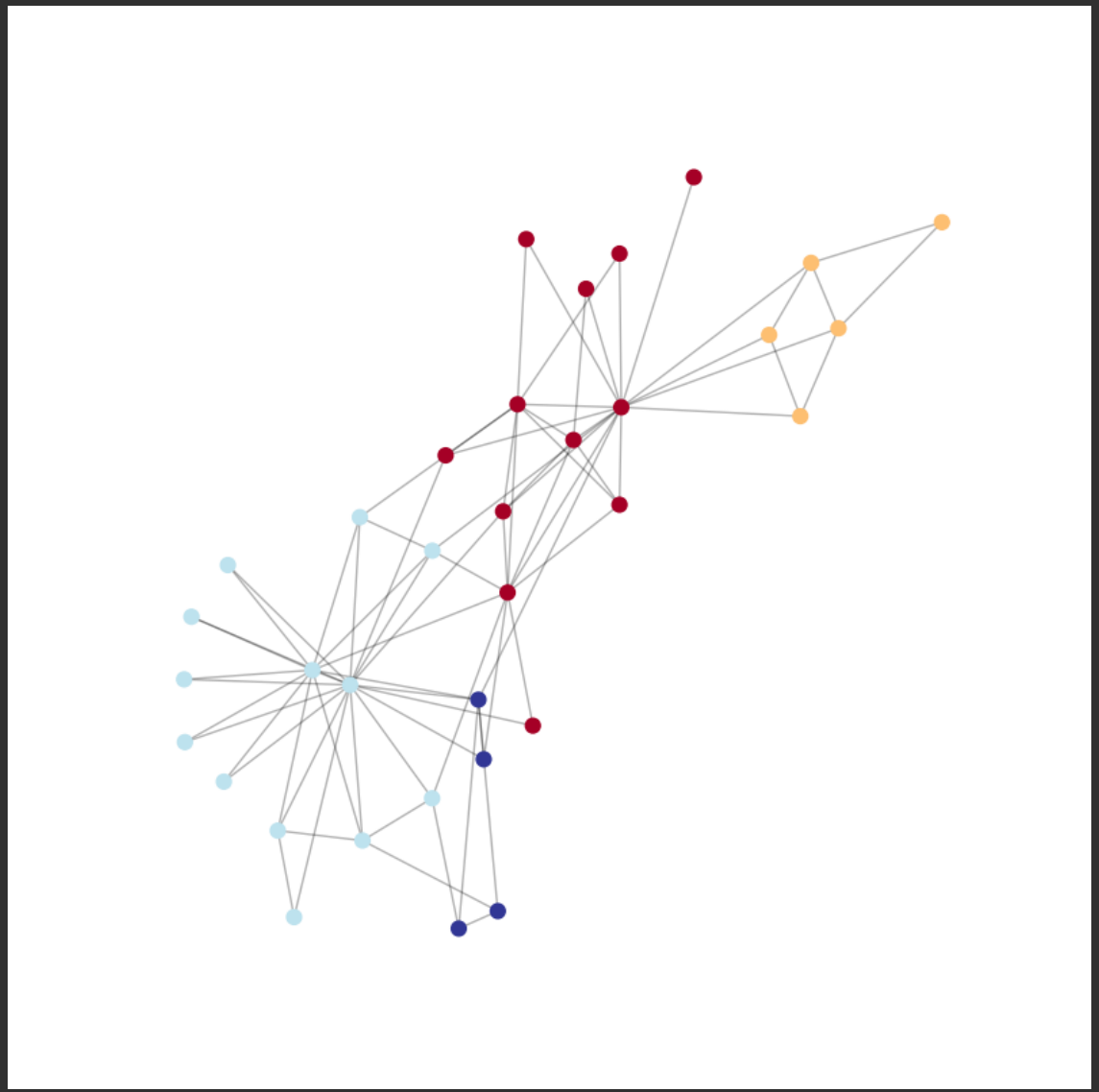
Desvantagens



Simulação - Igraph



Simulação - NetworkX



Método de Leiden

Artigo

SCIENTIFIC REPORTS

OPEN

From Louvain to Leiden: guaranteeing well-connected communities

Received: 20 November 2018
Accepted: 11 March 2019
Published online: 26 March 2019

V. A. Traag, L. Waltman & N. J. van Eck

Community detection is often used to understand the structure of large and complex networks. One of the most popular algorithms for uncovering community structure is the so-called Louvain algorithm. We show that this algorithm has a major defect that largely went unnoticed until now: the Louvain algorithm may yield arbitrarily badly connected communities. In the worst case, communities may even be disconnected, especially when running the algorithm iteratively. In our experimental analysis, we observe that up to 25% of the communities are badly connected and up to 16% are disconnected. To address this problem, we introduce the Leiden algorithm. We prove that the Leiden algorithm yields communities that are guaranteed to be connected. In addition, we prove that, when the Leiden algorithm is applied iteratively, it converges to a partition in which all subsets of all communities are locally optimally assigned. Furthermore, by relying on a fast local move approach, the Leiden algorithm runs faster than the Louvain algorithm. We demonstrate the performance of the Leiden algorithm for several benchmark and real-world networks. We find that the Leiden algorithm is faster than the Louvain algorithm and uncovers better partitions, in addition to providing explicit guarantees.

Algoritmo

1. Ordene a rede em uma fila
2. Remova 1 nó e mova-o para o cluster C que maximiza Q ($\Delta Q > 0$)
3. Os vizinhos do nó que não pertencem a C vão para o fim da fila
4. Refine a partição, com escolha aleatória de clusters bem-conectados
5. Use cada comunidade como um nó e recomece
6. Pare caso $\Delta Q \leq 0$

Algoritmo

1. Ordene a rede em uma fila
2. Remova 1 nó e mova-o para o cluster C que maximiza Q ($\Delta Q > 0$)
3. Os vizinhos do nó que não pertencem a C vão para o fim da fila
4. Refine a partição, com escolha aleatória de clusters bem-conectados
5. Use cada comunidade como um nó e recomece
6. Pare caso $\Delta Q \leq 0$

Algoritmo

1. Ordene a rede em uma fila
2. Remova 1 nó e mova-o para o cluster C que maximiza Q ($\Delta Q > 0$)
3. Os vizinhos do nó que não pertencem a C vão para o fim da fila
4. Refine a partição, com escolha aleatória de clusters bem-conectados
5. Use cada comunidade como um nó e recomece
6. Pare caso $\Delta Q \leq 0$

Algoritmo

1. Ordene a rede em uma fila
2. Remova 1 nó e mova-o para o cluster C que maximiza Q ($\Delta Q > 0$)
3. Os vizinhos do nó que não pertencem a C vão para o fim da fila
4. Refine a partição, com escolha aleatória de clusters bem-conectados
5. Use cada comunidade como um nó e recomece
6. Pare caso $\Delta Q \leq 0$

Algoritmo

1. Ordene a rede em uma fila
2. Remova 1 nó e mova-o para o cluster C que maximiza Q ($\Delta Q > 0$)
3. Os vizinhos do nó que não pertencem a C vão para o fim da fila
4. Refine a partição, com escolha aleatória de clusters bem-conectados
5. Use cada comunidade como um nó e recomece
6. Pare caso $\Delta Q \leq 0$

Algoritmo

1. Ordene a rede em uma fila
2. Remova 1 nó e mova-o para o cluster C que maximiza Q ($\Delta Q > 0$)
3. Os vizinhos do nó que não pertencem a C vão para o fim da fila
4. Refine a partição, com escolha aleatória de clusters bem-conectados
5. Use cada comunidade como um nó e recomece
6. Pare caso $\Delta Q \leq 0$

Vantagens

- Garante clusters bem-conectados
- Mais rápido que o Louvain (movimentos locais rápidos) - complexidade estimada: $\mathcal{O}(n)$ em grafos esparsos (?)

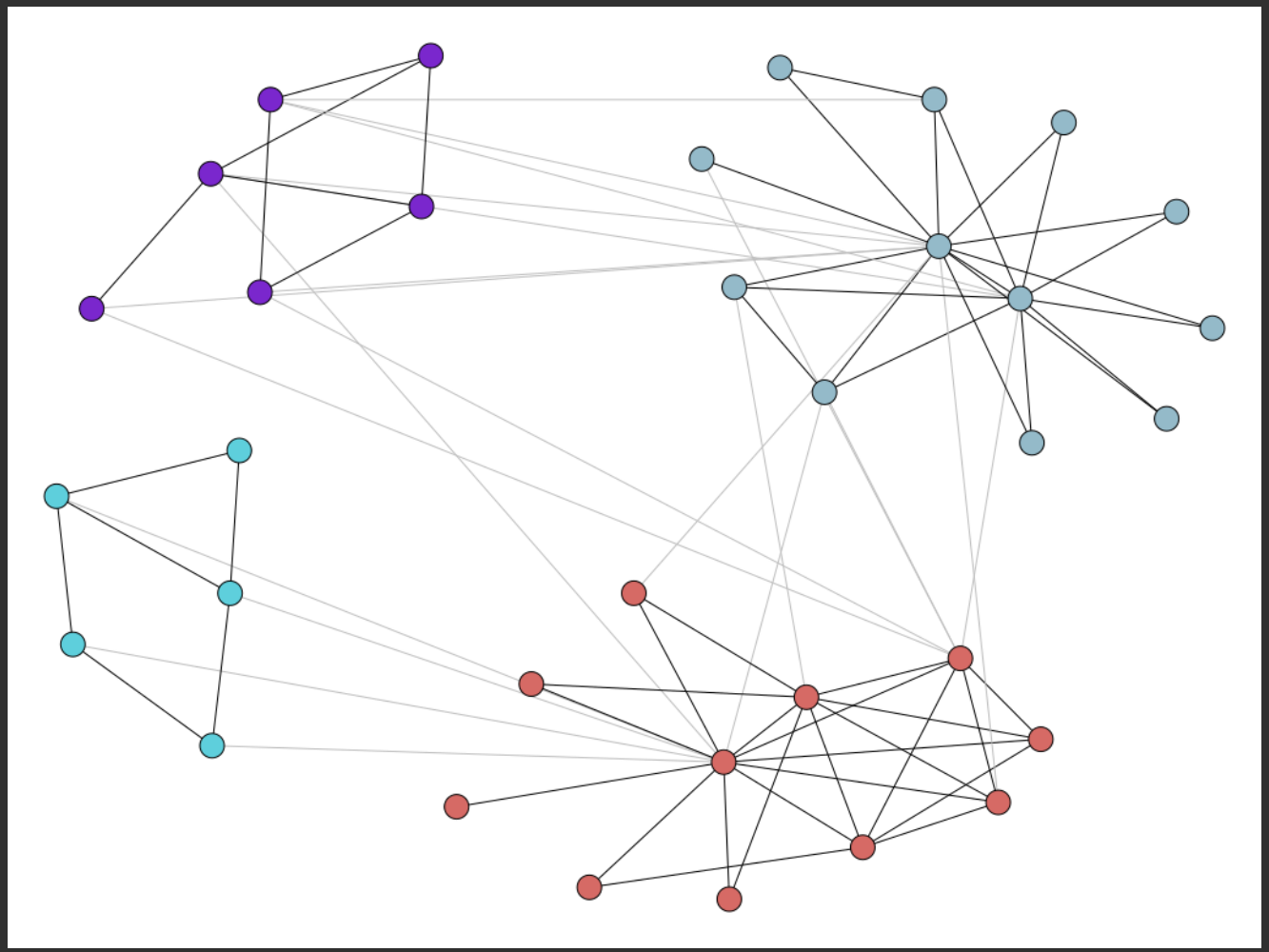
Vantagens

- Garante clusters bem-conectados
- Mais rápido que o Louvain (movimentos locais rápidos) - complexidade estimada: $\mathcal{O}(n)$ em grafos esparsos (?)

Desvantagens

- Ainda sofre do problema do limite na modularidade

Simulação

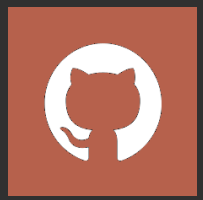


Conclusão

Comparação

	N. Vértices	N. Arestas	Modularidade máxima			
			Louvain (NetX)	Louvain (Igraph)	Leiden (Igraph)	Newman (Igraph)
Karate	34	77	0,416 ± 0,001	0,416	0,417	0,391
C. Elegans	453	2025	0,436 ± 0,002	0,428	0,443	0,348
Drosófila	1781	8911	0,418 ± 0,002	0,409	0,416	0,324
IMDB	896305	3782447	0,691 ± 0,002	0,692	0,689	-
			Tempo [ms]			
			2,41 ± 0,04	0,23 ± 0,03	0,5 ± 0,1	13,5 ± 0,6
			(6 ± 1)E+01	3,4 ± 0,3	3,4 ± 0,2	71 ± 1
			(34 ± 5)E+01	16 ± 2	17 ± 4	(47 ± 2)E+01
			(7 ± 1)E+05	(46 ± 2)E+03	(29 ± 2)E+03	-

Obrigado!



https://github.com/AIandroid/CommunityDetection

Referências

- Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008,** P10008 (out. de 2008).
- Brandes, U. *et al.* Maximizing modularity is hard. (Ago. de 2006).
- Lancichinetti, A. & Fortunato, S. Limits of modularity maximization in community detection. *Physical Review E* **84** (dez. de 2011).
- Mukherjee, A., Choudhury, M., Peruani, F., Ganguly, N. & Mitra, B. *Dynamics On and Of Complex Networks.* (Birkhäuser, NY, 2013).
- Newman, M. *Networks.* 2ª ed. (Oxford University Press, 2018).
- Newman, M. E. J. Modularity and community structure in networks. *Proc. of the National Ac. of Sciences* **103,** 8577–8582 (mai. de 2006).
- Rossi, R. A. & Ahmed, N. K. *The Network Data Repository with Interactive Graph Analytics and Visualization.* em *AAAI* (2015). http://networkrepository.com.