Main Challenges in Networks Community Structure Validation

Luisa Cutillo

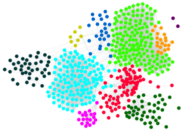
School of Mathematics, University of Leeds
l.cutillo@leeds.ac.uk

UNIVERSITY OF LEEDS

Objectives&Motivation

- **high throughput technologies** led to an increasing availability of data and to the need for novel statistical tools.
- **Biological networks**: Graphs provide a mathematical representation of patterns of interaction between appropriate biological elements.
- Need for methods to validate a Network.
- Need for methods to compare Networks.

Background



- a network is said to have a "community structure" if the vertices can be divided in g groups densely connected within them and loosely connected between them.
- Finding communities within an arbitrary complex network can be a computationally difficult task.

We **focus** on: **testing the robustness** of the recovered partition of a given community detection method and **comparing different networks** / partitions of the same network.

Questions

Our main research questions

- Q1: How can we test the robustness of the recovered partition of a given community detection method?

Questions

Our main research questions

- Q1: How can we test the robustness of the recovered partition of a given community detection method?
- Q2: How can we compare two (or more) networks and their community structures?

Questions

Our main research questions

- Q1: How can we test the robustness of the recovered partition of a given community detection method?
- Q2: How can we compare two (or more) networks and their community structures?
- Q3: Is it an advantage to integrate metadata to infer communities?

Robustness

Q1

- How can we test the robustness of the recovered partition of a given community detection method?

Robustness

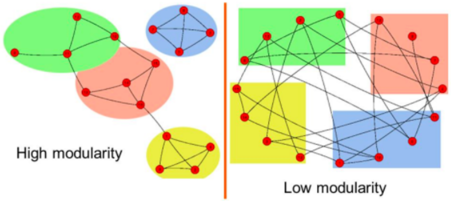
Q1

- How can we test the robustness of the recovered partition of a given community detection method?
  We need a metric and a Hypothesis testing!

This research question is related to my paper:
Carissimo, A, Cutillo L, Defeis, I **Validation of community robustness** *Computational Statistics and Data Analysis.* (2017).

Background: Modularity

- The modularity Q of Newman and Girvan [Newman, 2004] was the first attempt to give an answer to this question.
- It is the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random
- basic idea: a random graph is not expected to have a cluster structure.
- Limitation: [Fortunato, 2010] high modularity is a necessary but not sufficient condition for a strong community structure



Variation of Information

- The Variation of Information is an information theoretic criterion for comparing two partitions, or clusterings, of the same data set introduced in [Meila, 2007].
- It is a metric and measures the amount of information lost and gained in changing from clustering $\mathcal{C}$ to clustering $\mathcal{C}'$.
- The *VI* has been proven to be superior to metrics for comparing clusterings, namely $Rand$, $adjusted\ Rand$, $Fowlkes - Mallows$, $Jaccard$ and $Wallace$ and was used as a loss function in the context of Bayesian cluster analysis showing several desirable properties [Wade, 2015].

Variation of Information

VI can be expressed as:

$$VI\left(\mathcal{C},\mathcal{C}'\right) = H\left(\mathcal{C}|\mathcal{C}'\right) + H\left(\mathcal{C}'|\mathcal{C}\right). \tag{2.1}$$

The first term measures the amount of information about $\mathcal{C}$ that we loose, while the second measures the amount of information about $\mathcal{C}'$ that we gain, when going from clustering $\mathcal{C}$ to clustering $\mathcal{C}'$.

VI metric is the basis of the hypothesis testing procedures we propose to establish the statistical significance of a recovered community structure in a complex network

Our approach

Proposal

We propose to compare two different partitions on the same graph building on a special metric called Variation of Information (VI) [Meila, 2007].

VI curves

We will show how to build a VI curve (VIc) comparing the partition of our original network and the partition of a perturbed version of the original network.

Testing Procedure

We set up an hypothesis testing procedure to test if the VIc is significantly different from a random VI curve (referred to as VIc_{random}), obtained computing VI between the partition of a null random network and the partition of different perturbed version of such null network.

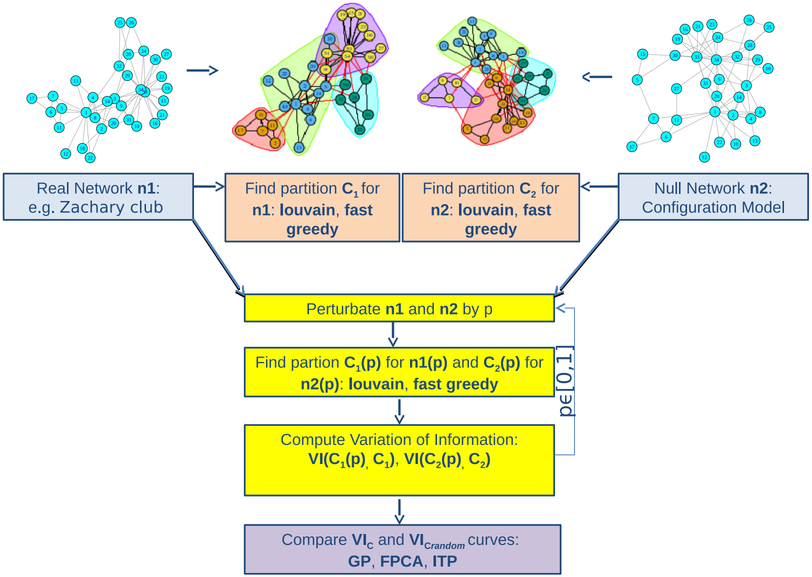

Real Network **n1**: e.g. Zachary club
Find partition C_1 for **n1**: **louvain**, **fast greedy**
Find partition C_2 for **n2**: **louvain**, **fast greedy**
Null Network **n2**: Configuration Model
Perturbate **n1** and **n2** by p
Find partion $C_1(p)$ for **n1(p)** and $C_2(p)$ for **n2(p)**: **louvain**, **fast greedy**
$p \in [0,1]$
Compute Variation of Information: $VI(C_1(p), C_1)$, $VI(C_2(p), C_2)$
Compare VI_C and $VI_{Crandom}$ curves: **GP**, **FPCA**, **ITP**

Figure: Overall procedure map

Build two VI curves

- The first curve *VIc* is obtained computing VI between the partition of our original network and the partition of different perturbed version of our original network.
- The second curve VIc_{random} is obtained computing VI between the partition of a null random network and the partition of different perturbed version of such null network. We use a rewiring procedure as suggested in [Karrer, 2008].

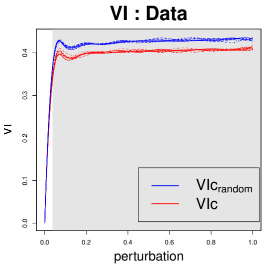


Figure: Lower Q=0.2

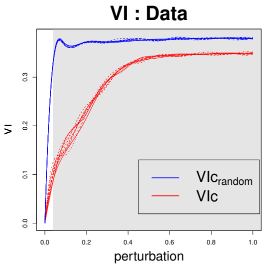


Figure: Higher Q=0.6

Choice of the null Model

The choice of the null random network is really delicate, because we expect that it has the same structure of our original graph but with completely random edges.

Null Model: Configuration and dk

CM

- The Configuration Model [Bender, 1978] is associated with the degree sequence of the observed graph $\mathbf{d} = \{d(1), \ldots, d(n)\}$, i.e. CM($\mathbf{d}$).
- It is able to preserve strongly heterogeneous degree distributions of real networks and is the standard null model for empirical patterns. It generates every possible graph with the given degree distribution with equal probability.

dk

- In the *dk* model [Orsini, 2015] a complete set of basic characteristics of the network structure, namely a $dk - series$, is employed to generate *dk*-random graphs whose degree distributions, degree correlations and clustering are as in the corresponding real network.
- The implementation *RandNetGen* of the *dk* model availabe on *github* (`https://github.com/polcolomer/RandNetGen`)(option $-dk$ 2.5).

Testing Procedure

Testing Procedure

We use a functional data analysis approach to test if the two groups of curves represent "the same process" or "different processes". The testing procedure we rely on is based on a tool set up for **time course** microarray, namely Gaussian Process (GP) regression [Kalaitzis, 2011].

GP for gene expression data

Aim to detect if the profile has a significant underlying signal or the observations are just random fluctuations. In this case we reformulate the testing problem working on $\log_2 \frac{VIc}{VIc_{random}}$

Testing Procedure Formulation

H_0 Formulation

The two measured *VI* curves are independent realisations of two underlying processes say X_1 and X_2 observed with noise on a finite grid of points $p \in [0, 1]$. We test:

$$H_0 : \mathrm{X}_1 \overset{\mathrm{d}}{=} \mathrm{X}_2, \tag{3.1}$$

versus the alternative hypothesis

$$H_1 : \mathrm{X}_1 \overset{\mathrm{d}}{\neq} \mathrm{X}_2,$$

where $\overset{d}{=}$ means that the processes on either side have the same distribution.

GP Testing

a gaussian process is a *collection of random variables, any finite number of which have a joint Gaussian distribution* and is completely specified by its mean function and its covariance function.

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')). \tag{3.2}$$

In this framework the hypothesis testing problem can be reformulated, over the perturbation interval $[0, 1]$, as:

$$H_0 : \log_2 \frac{VIc(x)}{VIc_{random}(x)} \sim \mathcal{GP}(0, k(x, x')),$$

against

$$H_1 : \log_2 \frac{VIc(x)}{VIc_{random}(x)} \sim \mathcal{GP}(m(x), k(x, x')).$$

The derived marginal likelihood, enables then to compare or rank different models by calculating the Bayes Factor (BF).

Alternative Testing Procedure

we also explore:

FPCA

the methodology developed in [Pomann, 2016] based on Functional Principal Components Analysis (FPCA) to test (3.1).

Interval Wise

the approach described in [Pini, 2016], addressing a domain-selective inferential procedure, providing an interval-wise non parametric functional testing able not only to assess (3.1), but also to point out specific differences.

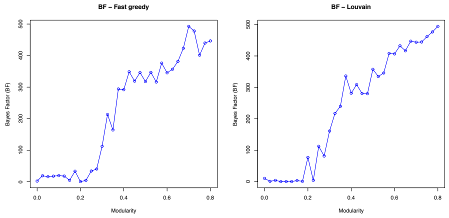
Application

- For each simulated or real Network we performed the community extraction step, using some tools embedded in the *R* package *igraph*
- we use a greedy optimisation of the modularity (*cluster_fast_greedy*) and another based on a multi-level optimisation of the modularity (*cluster_louvain*).
- Both these methods enables for an automatic definition of the optimal number of communities, are specific for large networks and are based on the optimisation of the modularity.

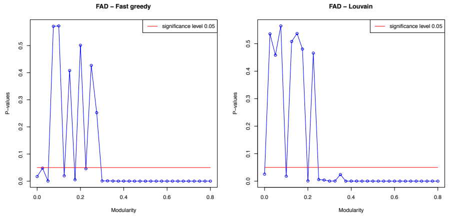
Simulation strategy

- We generated modular random network graphs using the model implemented in [Sah, 2014].
- The model generates undirected, simple, connected graphs with prescribed degree sequences and a specified level of community structure.
- Specifically, we generated a modular random graph on a grid of 33 modularity values $Q \in [0, \ldots, 0.8]$, grid step 0.025, using a power law for degree distribution and with size=2000, number of modules=10 and average degree=10.

Simulation study, Null model= CM

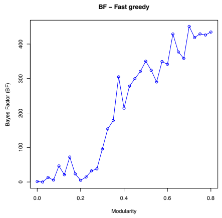


(a) test=GP, (b) test=GP

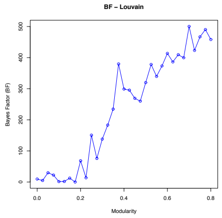


(c) test=FAD (d) test=FAD

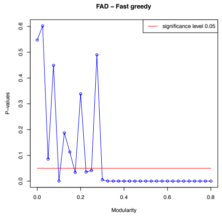
Simulation study, Null model= dk



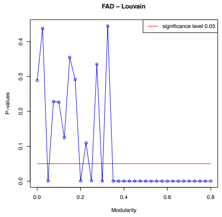
(e) test=GP,



(f) test=GP



(g) test=FAD



(h) test=FAD

Real Data application

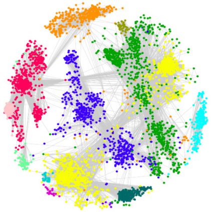
We selected four different publicly available datasets namely two biological (protein-protein interaction networks, *Nexus* 5 and *Barabasi*), one representing the western states power grid of United States (*Nexus* 15) and a social dataset (*Facebook*).

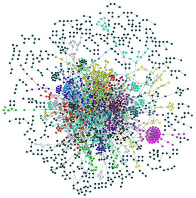
Real Networks summary

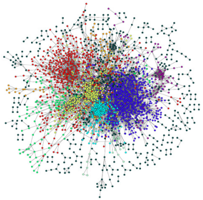
	FB	FB(0.29)	FB(0.68)	Nexus 5	Nexus 15	Barabasi
nv	4039	224	534	2617	4941	1870
ne	88234	3192	4813	11855	6594	2240

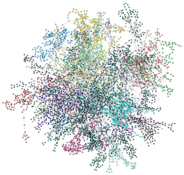
Table: Number of vertex (*nv*) and number of edges (*ne*) for each of the 4 real analysed datasets and for the two less modular ($Q = 0.29$) and the most modular ($Q = 0.68$) ego networks in Facebook (*FB*)

Real Networks plots


(i) Facebook

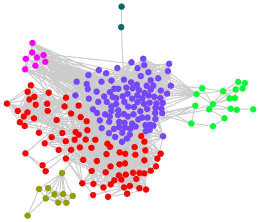

(j) Barabasi


(k) Nexus 5

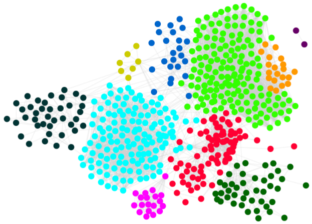

(l) Nexus 15

Two Facebook ego-networks

Comparison of two Facebook ego-networks



(m) $Q = 0.29$ Facebook ego-network



(n) $Q = 0.68$ Facebook ego-network

Figure: For both the less modular (a) and the most modular (b) ego networks, we show the extracted community found by the proposed method *Fast Greedy*. Only the community with more than the 5% of nodes are displayed.

GP table results

The resulting BF are very high for either *Fast Greedy* or *Louvain* clustering. This gives strong statistical evidence that the four analysed networks have a robust clustering structure hence the recovered community structures are not likely to be random.

Datasets	Fast Greedy	Louvain
Barabasi	284.411	243.816
FB (10 ego)	297.251	361.060
FB ($Q = 0.29$)	153.760	258.920
FB ($Q = 0.68$)	290.503	338.269
Nexus 5	340.795	431.477
Nexus 15	503.183	495.810

Table: GP Bayes Factor on the 4 datasets and for the two less modular ($Q = 0.29$) and the most modular ($Q = 0.68$) ego networks in Facebook (*FB*), after clustering via *Fast Greedy* or *Louvain*

Discussion about Q1

Q1

How can we test the robustness of the recovered partition for a given community detection method?

- We propose an effective procedure to evaluate the **robustness** of a clustering.
- We specify a perturbation strategy and a null model to build a set of procedures based on **VI as stability measure**.

To Do

An extension of the proposed approach would be using a different clustering stability measure, for example the *Normalized Mutual Information* measure. This would also lead to a comparison of the performance of different measures for clustering stability.

Comparing Networks

Q2

- Q2: How can we compare two (or more) networks and their community structures?

Comparing Networks

Q2

- Q2: How can we compare two (or more) networks and their community structures?

This research question is related to my paper:
arXiv:1710.06611 **An inferential procedure for community structure validation in networks** Luisa Cutillo, Mirko Signorelli

Backgroud

Given $\mathcal{G}_1 = (V, E_1)$ and $\mathcal{G}_2 = (V, E_2)$, with true community structures $\mathcal{P}_1$ and $\mathcal{P}_2$.

- Apply a clustering algorithm to $\mathcal{G}_1$ and $\mathcal{G}_2$ and obtain $\mathcal{P}_1^* = \mathcal{P}_{\mathcal{G}_1}(V)$ and $\mathcal{P}_2^* = \mathcal{P}_{\mathcal{G}_2}(V)$.

Do $\mathcal{G}_1$ and $\mathcal{G}_2$ share (roughly) the same community structure?

Backgroud

Given $\mathcal{G}_1 = (V, E_1)$ and $\mathcal{G}_2 = (V, E_2)$, with true community structures $\mathcal{P}_1$ and $\mathcal{P}_2$.

- Apply a clustering algorithm to $\mathcal{G}_1$ and $\mathcal{G}_2$ and obtain $\mathcal{P}_1^* = \mathcal{P}_{\mathcal{G}_1}(V)$ and $\mathcal{P}_2^* = \mathcal{P}_{\mathcal{G}_2}(V)$.

Do $\mathcal{G}_1$ and $\mathcal{G}_2$ share (roughly) the same community structure?

is $\mathcal{P}_1 \approx \mathcal{P}_2$?

- YES $\Rightarrow$ $\mathcal{P}_1$ is also a good partition for $\mathcal{G}_2$ (and $\mathcal{P}_2$ for $\mathcal{G}_1$);
- NO $\Rightarrow$ $\mathcal{P}_1$ does not partition $\mathcal{G}_2$ well.

How can we test this?

IDEA: test network enrichment between communities $\mathcal{P}_1 = \mathcal{P}_{\mathcal{G}_1}(V)$ in $\mathcal{G}_2$ using NEAT.

About Network Enrichment Analysis

Enrichment via NEAT: n_{AB}, the number of "arrows" between groups A and B, is the realization of a r. v. N_{AB} with $E(N_{AB}) = \mu_{AB}$.

$$H_0\text{: } \mu_{AB} = \mu_0 \text{ (no enrichment)}$$

$$H_1\text{: } \mu_{AB} \neq \mu_0 \text{ (enrichment: } > \text{over, } < \text{under)}$$

What are we testing?

Null model: in absence of enrichment (H_0 true)

$$N_{AB} \sim hypergeom(n = o_A, K = i_B, N = i_V),$$

`R` **package**: `neat` on `CRAN`.

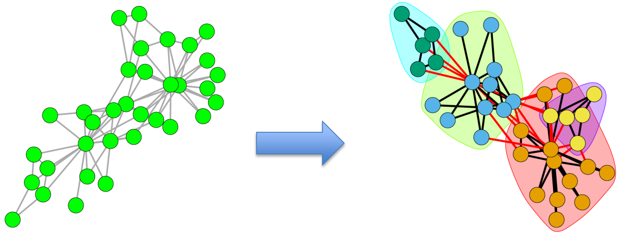
What are we testing?

- Apply clustering algorithm to $\mathcal{G}_1$ to obtain $\mathcal{P}_1^* = \{C_1^*, C_2^*, ..., C_q^*\}$
- Test network enrichment within and between C_r^* in $\mathcal{G}_2$

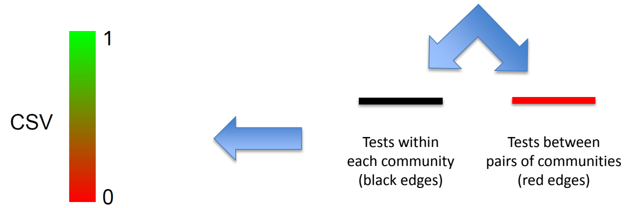
Over and Under enrichment

- **overenrichment within each community** C_r
- **underenrichment between each pair of communities** (C_r, C_s)

How to build a community structure validation index



(a) (b)



Tests within each community (black edges)

Tests between pairs of communities (red edges)

(d) (c)

Index definition

Community Structure Validation (CSV) index

$$UCSV = \frac{\sum_{r=1}^{q} I(\hat{p}_{rr} \leq \alpha) + \sum_{r>s} I(\hat{p}_{rs} \leq \alpha)}{q(q+1)/2}.$$

WCSV weights addends by adjusted p-values ($\hat{p}$).

- $UCSV \in [0,1]$, $UCSV = 1$ indicates that $\mathcal{P}_1 \equiv \mathcal{P}_2$ (or *best* partition if self tested).
- The higher $UCSV$, the more similar the community structures.

Compare two Networks

- Find a partition of $\mathcal{G}_1$: $\mathcal{P}_1 = \{C_{11}, ..., C_{1q}\}$. Similarly, obtain $\mathcal{P}_2$ from $\mathcal{G}_2$;
- Compute the relative indices

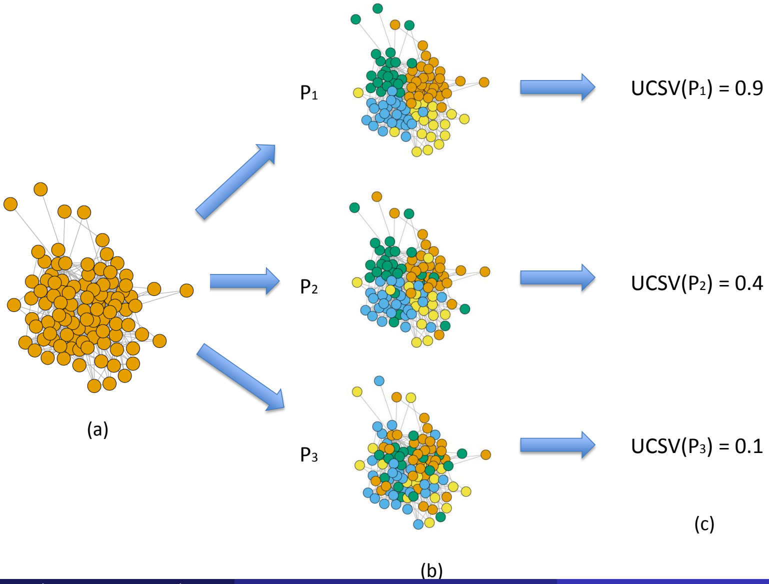
$$R_{UCSV}\left(\mathcal{P}_i|\mathcal{G}_j\right) = \frac{UCSV(\mathcal{P}_i|\mathcal{G}_j)}{UCSV(\mathcal{P}_i|\mathcal{G}_i)},\ i \neq j, \tag{5.1}$$

which compare the values of the UCSV index of partition $\mathcal{P}_i$ in graph $\mathcal{G}_j$ with the value of UCSV for $\mathcal{P}_i$ in $\mathcal{G}_i$.

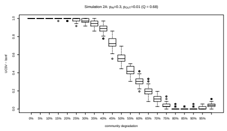
Compare more then two Networks

- We compute the relative indices $R_{i,j} = R_{UCSV}(\mathcal{P}_i|\mathcal{G}_j)$ a $\forall i, j \in \{1, \dots, n\}, i \neq j$.
- We build a similarity matrix $S = \left(R + R^T\right)/2$, and distance matrix $D = 1 - S$.

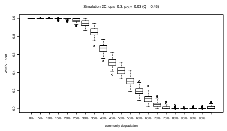
Use CSV to compare partitons



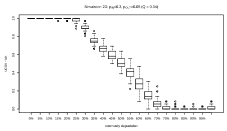
Sensitivity to community degradation



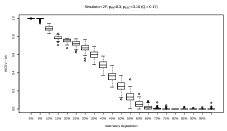
(a)



(b)



(c)

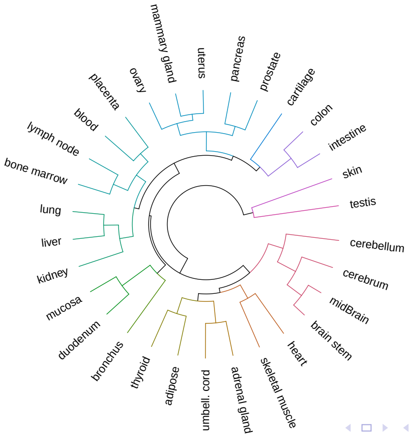


(d)

High Q: $UCSV \approx 1$ if comm. degradation $\leq 25\%$; Low Q: $UCSV \approx 1$ if comm. degradation $\leq 5\%$.

Case Study: Tissue comparison

30 tissue-specific human gene co-regulation networks were reverse engineered. [See Gambardella et al. 2013, Bioinformatics]. We find 13 clusters, highlighted in different colours. For each $\mathcal{G}_i$, we use Louvain community extraction method to obtain a partition $\mathcal{P}_i$.



Discussion about Q2

Q2

How can we compare two (or more) networks and their community structures?

Discussion about Q2

Q2

How can we compare two (or more) networks and their community structures?

- We propose an effective procedure based on an indicator CSV
- This procedure can be exploited to:
  - compare clusters
  - compare two or more networks by assessing the similarity between their community structure

To Do

An extension of the proposed approach would be proving theoretical property of CSV or to a modified version of it.

Use of Metadata

Q3

- Q3: Is it an advantage to integrate metadata to infer communities?

This research question is related to xxx paper....plenty of space!

Community detection and metadata

Reference paper

- Newman and Clauset (2016). Structure and inference in annotated networks. Nature Communications, 7:11863.
- Idea: a network $G = (V, E)$ often comes with metadata M about the nodes.
- can we use these metadata to improve community detection?

Caveats

- The method allows to consider 1 label at a time
- Implemented in C
- Number of clusters specified by user
- No clear definition of community structure in the paper, they don?t necessarily aim to find a partition where clusters are highly connected within and poorly connected between
- Assessment of community structure based on ground truth (synthetic data), typically not available on real networks
- **Advantage**: when network is not very informative about communities, the label can improve the accuracy of the clustering

Shall we use metadata?

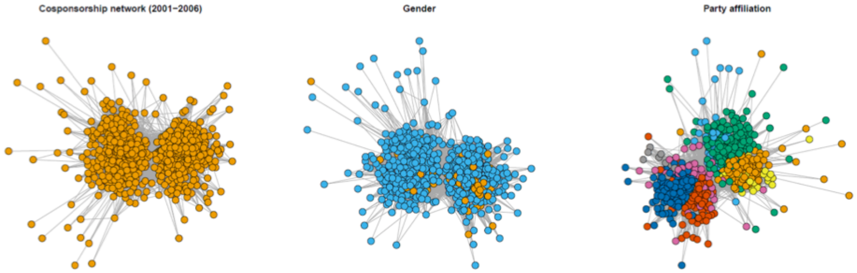
Question

Does the use of metadata, as proposed in Newman and Clauset (2016), really improve the quality of community detection on REAL networks where GROUND TRUTH is UNKNOWN?

On going answer

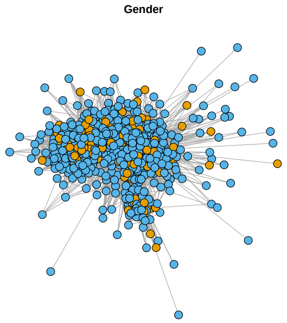
Let's have a look at a **case study**!

Case study: Italian Pariament



- Cosponsorship network between 609 Deputies in Italian Parliament (2001-2006)
- Metadata: gender (M/F) and party (8 groups)

Case study: Italian Pariament



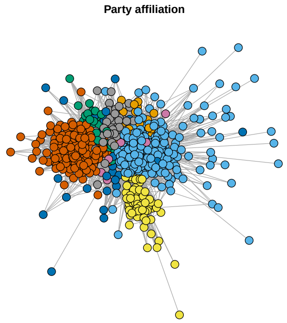


Figure: Bill cosponsorship network between Deputies in Italian Parliament during the XVI legislature (2008-2013) and metadata: gender (orange = female, lightblue = male) and party membership (colors correspond to the 8 parliamentary groups represented in the Chamber).

Fast greedy clustering algorithm detects 3 clusters, whereas Louvain and the leading eigenvalue methods detect 4 clusters; finally, walktrap extracts 4 clusters and a few isolate nodes.

Method	$WCSV_1$
Fast greedy	0.333
Newclau + party, 3 clusters	0.833
Newclau + gender, 3 clusters	0.833

Table: Value of the weighted community structure validation index for the partitions with 3 clusters.

Method	$WCSV_1$
Walktrap	0.90
Leading eigenvalue	0.70
Louvain	0.90
Newclau + party, 4 clusters	0.90
Newclau + gender, 4 clusters	0.90

Table: Value of the weighted community structure validation index for the partitions with 4 clusters.

Discussion on Q3

- How to define *community structure*? (network-based only?)
- What if a ground truth is not available?
- How do we choose metadata? Shall we combine more than 1?
- If a community structure does not exist, shall we rather use metadata and ignore the network?

References

Newman M. E. J. and Girvan M.
Finding and evaluating community structure in networks.
Phys. Rev. E. **69, 026113**. (2004).

Fortunato S.
Community detection in graphs.
Phys. Rep. 75–174 **486**. MR2580414. (2010).

Meilă M.
Comparing clusterings – an information based distance.
J. Multivariate Anal. **98** 873–895. (2007).

Wade,S. and Ghahrmani Z.
Bayesian cluster analysis: Point estimation and credible balls.
arXiv:1505.03339(2015)

Karrer B., Levina E. and Newman M. E. J.
Robustness of community structure in networks.
*Phys. Rev. E.***77, 046119**. (2008).

Bender, E. A. and Canfield, E. R.
The asymptotic number of labeled graphs with given degree sequences.
Journal of Combinatorial Theory A. **24** 296–307. MR0505796. (1978).

Orsini C. et al.
Quntifying randomness in real networks.
Nature Communications. **6, 8627**. (2015).

Kalaitzis A. A. and Lawrence N. D.
A Simple Approach to Ranking Differentially Expressed Gene Expression Time Courses through Gaussian Process Regression.
BMC Bioinformatics. **12:180**. (2011).

Pomann, G.-M., Staicu, A.-M. and Ghosh, S.
A two-sample distribution-free test for functional data with application to a diffusion tensor imaging study of multiple sclerosis.
Journal of the Royal Statistical Society: Series C (Applied Statistics) **65** 395–414. doi: 10.1111/rssc.12130. (2016).

Pini , A. and Vantini, S.
The interval testing procedure: A general framework for inference in functional data analysis.
Biometrics. doi: 10.1111/biom.12476 . (2016).

Sah P., Singh L. O., Clauset A. and Bansal S.
Exploring community structure in biological networks with random graphs.
*BMC Bioinformatics.***15:220. doi:10.1186/1471-2105-15-220**.(2014).

Carissimo, A, Cutillo L, Defeis, I
Validation of community robustness.
Computational Statistics and Data Analysis.. (2017).

The End

Remarks

$$I\left(\mathcal{C},\mathcal{C}'\right)=\sum_{k=1}^{K}\sum_{k'=1}^{K'}P\left(k,k'\right)\log\frac{P\left(k,k'\right)}{P\left(k\right)P\left(k'\right)}. \tag{6.1}$$

$P\left(k\right)$ is the probability of a point being in cluster C_k and $P\left(k,k'\right)$ is the probability that a point belongs to C_k in clustering $\mathcal{C}$ and to $C_{k'}$ in $\mathcal{C}'$, i.e. $P(k)=|C_k|/n$ and $P(k,k')=|C_k\cap C_{k'}|/n$.

Perturbation strategy

Mimicking the approach proposed by [Sah, 2014], we restrict our perturbed networks to having the same numbers of vertices and edges as the original unperturbed network, hence only the positions of the edges change. Our perturbation strategy relies on the *rewire* function belonging to the *R* package *igraph*, using the option *keeping_degseq*. Moreover, we expect that a network perturbed by only a small amount has just a few edges moved in different communities, while a maximally perturbed network produces completely random clusters.

Our perturbation strategy consists in randomly rewiring a percentage p of edges while preserving the original graph's degree distribution. The rewiring algorithm indeed chooses two arbitrary edges in each step (e.g. (a, b) and (c, d)) and substitutes them with (a, d) and (c, b), if they do not already exists in the graph. The algorithm does not create multiple edges.

A null percentage of permutation $p = 0$ corresponds to the original unperturbed graph, while $p = 1$ corresponds to the maximal perturbation level.