university of groningen

UNIVERSITÀ DEGLI STUDI DI PADOVA

UNIVERSITY OF GRONINGEN
Johann Bernoulli Institute for Mathematics and Computer Science
UNIVERSITY OF PADOVA
Department of Statistical Sciences

INFERRING COMMUNITY-DRIVEN STRUCTURE IN COMPLEX NETWORKS

A dissertation supervised by promotors

PROF. ERNST C. WIT
PROF. MONICA CHIOGNA

and submitted by

MIRKO SIGNORELLI

in fulfillment of the requirements for the Degree of

PHILOSOPHIAE DOCTOR (PhD)

Inferring Community-driven Structure in Complex Networks
PhD dissertation of Mirko Signorelli

ISBN: 978-90-367-9576-0 (printed version), 978-90-367-9575-3 (electronic version).

Copyright © 2017 by Mirko Signorelli.
All rights reserved. No parts of this elaborate may be reproduced or transmitted in any form or by any means without prior permission of the author.

university of groningen

Inferring Community-driven Structure in Complex Networks

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. E. Sterken
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

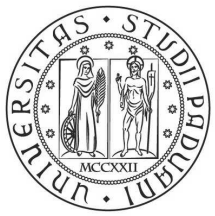
by

Mirko Signorelli

born on 11 May 1989
in Calcinate, Italy

Supervisors
Prof. E. C. Wit
Prof. M. Chiogna

Assessment committee
Prof. A. Brazzale
Prof. M. C. M. de Gunst
Prof. G. Scalia Tomba
Prof. T. A. B. Snijders



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova
Sede Consorziata: Università di Groningen

Dipartimento di Scienze Statistiche

SCUOLA DI DOTTORATO DI RICERCA IN SCIENZE STATISTICHE
CICLO XXIX

INFERRING COMMUNITY-DRIVEN STRUCTURE IN COMPLEX NETWORKS

Direttore della Scuola: Ch.mo Prof. Monica Chiogna

Supervisori: Ch.mi Prof. Monica Chiogna e Prof. Ernst C. Wit

Dottorando/a: Mirko Signorelli

17 gennaio 2017

Contents

Contents **vii**

Chapter 0: Preamble **3**
0.1 Overview . . . 3
0.2 Main contributions of the thesis . . . 4

Chapter 1: Introduction **5**
1.1 Foreword . . . 5
1.2 Graphs . . . 7
1.2.1 Relations and their properties . . . 7
1.2.2 Representing relations with graphs . . . 8
1.2.3 The adjacency matrix of a graph . . . 9
1.3 One network, multiple graphs? . . . 9
1.4 Chapter summaries . . . 11
1.4.1 Outline of Chapter 2 . . . 11
1.4.2 Outline of Chapter 3 . . . 11
1.4.3 Outline of Chapter 4 . . . 12
1.4.4 Outline of Chapter 5 . . . 13
Bibliography . . . 14

Chapter 2: NEAT: an efficient network enrichment analysis test **17**
2.1 Background . . . 17
2.2 Methods . . . 20
2.2.1 Enrichment test for directed networks . . . 22
2.2.2 Enrichment test for undirected networks . . . 25
2.2.3 Enrichment test for partially directed networks 26
2.2.4 Software . . . 26
2.3 Performance evaluation . . . 27

2.3.1 Simulation with directed networks 29
2.3.2 Simulation with undirected networks 30
2.4 Network enrichment analysis: an application to yeast . 32
2.4.1 Network enrichment analysis of environmental stress response in yeast 33
2.4.2 Network enrichment analysis of GO Slim sets: overlap does not imply enrichment 35
2.5 Conclusion 37
Bibliography 39

Chapter 3: A penalized inference approach to stochastic blockmodelling of community structure in the Italian Parliament 49
3.1 Introduction 49
3.1.1 Stochastic blockmodels 50
3.2 Bill cosponsorship in the Italian Parliament 52
3.3 Poisson process model of bill cosponsorship 53
3.3.1 Data generating process 53
3.3.2 Identifiability 55
3.3.3 Extendibility 57
3.4 Inference 58
3.4.1 Parameter estimation 58
3.4.2 Model selection 61
3.4.3 Reduced graph 62
3.5 Analysis of bill cosponsorship networks of the Italian Chamber of Deputies 65
3.6 Conclusion and discussion 69
Bibliography 72

Chapter 4: Joint modelling of community structure and nodal heterogeneity in networks 75
4.1 Introduction 75
4.2 Joint modelling of community structure and nodal heterogeneity 77
4.2.1 Background: from GLMs to GLMMs 77
4.2.2 Model specification 78
4.2.3 Model estimation 79

4.2.4 Results 80
4.3 Latent space models 83
4.3.1 Modelling networks with latent space models . 83
4.3.2 Estimation of latent space models 84
4.3.3 Application to bill cosponsorship networks . . 85
4.4 Discussion 89
Bibliography 90

Chapter 5: Clustering graphs using mixtures of generalized linear models 93
5.1 Introduction 93
5.2 Model specification 95
5.2.1 Mixtures of generalized linear models 95
5.2.2 Clustering networks with mixtures of generalized linear models 96
5.3 Model estimation with the EM algorithm 98
5.3.1 Implementation of the EM algorithm 98
5.3.2 Simulations 99
5.4 An extension of the EM algorithm based on Simulated Annealing (EMSAGC) 101
5.4.1 Implementation of the EMSAGC algorithm . . 101
5.4.2 Simulations 103
5.5 Example application 104
5.6 Concluding remarks and future work 108
Bibliography 109

Appendix A: Vignettes of the R package neat 113

Appendix B: Manual of the R package neat 119

Appendix C: Curriculum Vitae 133

Abstract 139

Samenvatting 141

Acknowledgements 143

As to Holmes, I observed that he sat frequently for half an hour on end, with knitted brows and an abstracted air, but he swept the matter away with a wave of his hand when I mentioned it. “Data! Data! Data!” he cried impatiently. “I can’t make bricks without clay”.

SIR ARTHUR CONAN DOYLE,
The Adventure of the Copper Beeches

Chapter 0

Preamble

0.1 Overview

The last decades have witnessed a growing interest in the analysis of relational data. Typically, these data come in the form of a network specifying a list of relations between individuals or objects and are represented by means of a graph, which translates objects into nodes and relations into edges connecting the nodes.

Interest in the study of networks started in 1934, when the psychosociologist Jacob Moreno introduced sociograms as a way to represent relations between individuals. For many decades, research on networks was mostly focused on the study of random graph structures on the theoretical side, and on qualitative analyses of sociological networks as concerns applications. In the Eighties, a more quantitative approach to the study of social networks was undertaken and many popular network models (such as the p_1 model, exponential random graphs and stochastic blockmodels) were introduced. Attention, however, was still restricted to the study of small networks with a few nodes because of difficulties in data collection and computational limitations.

Recent technological advances such as the development of sensor-based measurements, next generation sequencing techniques and functional magnetic resonance imaging, as well as the advent and diffusion of social media, have widely simplified the collection of network data, fostering the analysis of larger network datasets. Nowadays, networks are a subject of interest in a varied range of disciplines, including sociology, medicine, biology, neuroscience, finance and engineering. Understanding relations encoded in large graphs, however, still repre-

sents a challenging task, and tools that can help to summarize and simplify complex networks are needed.

In this thesis, we will present some statistical methods which aim at providing substantive help in the interpretation of complex networks. The methods have been tailored so as to take into account features that are relevant to the specific applications considered.

0.2 Main contributions of the thesis

The main contributions of the thesis can be summarized as follows.

- We propose NEAT, a novel and efficient statistical test for the analysis of genetic networks that allows to overcome the limitations of existing network enrichment analysis tests. The test has been implemented in the R package `neat`, which is freely available from CRAN (Chapter 2).
- We propose two extensions of stochastic blockmodels for networks, which allow to model community structure in networks while accounting for observed sources of nodal heterogeneity (Chapter 3) and for both observed and unobserved sources of heterogeneity (Chapter 4), respectively. We implement the proposed extensions in a variable selection framework by making use of penalized inference methods.
- We provide an analysis of collaborations between political parties in the Italian Parliament, by considering bill cosponsorship networks in the Chamber of Deputies from 2001 to 2015 (Chapters 3 and 4).
- We propose a model that allows to detect clusters of graphs within a sequence of graphs, based on mixtures of generalized linear models. We develop two different algorithms (EM and EMSAGC) to estimate the model (Chapter 5).

Chapter 1

Introduction

1.1 Foreword

Interest towards networks can be dated back at least to 1934, when Jacob Levy Moreno wrote the book "Who shall survive?" [Moreno, 1934]. Moreno was a psychiatrist who though that Freud's psychoanalysis artificially isolated individuals from their usual social settings; in contrast to this, he advocated that psychotherapy should reproduce the social settings that an individual faces, and which could be at the origin of their traumas. He therefore invented the psychodrama, a theatrical representation where the patient is encouraged to perform and reproduce events from their past.

In line with this view, Moreno developed an interest in the study of interactions between individuals, and in "Who shall survive?" he introduced the sociogram as a way to represent relations between individuals. Using the current terminology, we could say that Moreno was interested in understanding the process of formation and the features of social networks, and that he started to employ graphs (what he called "sociograms") to represent these networks.

From Moreno's perspective, thus, a graph represented a primarily visual tool that allowed him to gain an insight into a complex tangle of relations between a limited number of individuals. All along the 82 years that separate us from Moreno, however, scholars from many fields have increasingly devised a plentiful of networks of different types and sizes.

Sociologists have long since been interested in the study of the patterns through which relations such as friendship or collaboration can arise between human beings [Sampson, 1969]. More recently, ecolo-

gists have started to employ networks to understand how animals relate to each other [Shizuka et al., 2014]. Networks are also employed to model the spread of infectious diseases, in an attempt to find ways to limit their diffusion [Klovdahl, 1985].

Networks, however, can be used not only to describe relations between living beings, but also between objects or organizations. In genetics, cell biologists soon realized that genes do not work in isolation, but they act in a concerted manner to carry out most cellular functions [Alberts et al., 2004]. Therefore, networks have been employed to represent functional couplings or regulatory mechanism between genes [Barabasi and Oltvai, 2004]. Political scientists have used networks to describe international relations between States [Cranmer et al., 2014]. Engineers, instead, use networks to represent flows of individuals or goods between different points in space [Guimera et al., 2005], and aim at optimizing these flows.

The growing interest in network science that the last decades have witnessed has been fostered by technological advances which have highly facilitated data collection: think, for example, to the development of sensor-based measurements, of next generation sequencing techniques and of functional magnetic resonance imaging, or to the large diffusion of social media such as Facebook or Twitter. These developments have rapidly expanded the focus of network science from networks with a few nodes (typically a few tens) to networks with hundreds, thousands or even millions of nodes.

The amount of information that is encoded in large networks challenges our capacities of understanding: as every node in a graph can be related to any other node, a graph with n nodes (and without self-loops) can consist of at most $n^2 - n$ arrows if directed, and $n(n-1)/2$ edges if undirected. Visualization of large networks typically leads to overly complicated pictures, whence it is hard to gain a synthetic insight.

Often in applications, however, the attention can be shifted from individual nodes to groups of nodes, and one could wonder how these groups, rather than the original nodes, are related to each other. By doing so, one can rephrase the original question on how a large number of individuals or objects interact with each other into the problem of reconstructing the pattern of interactions between these groups. It

goes without saying that some information will get lost in the translation of the original graph into a reduced graph summarizing group-group relations. Nevertheless, the reduced graph can provide a powerful tool for summarizing complex networks. This thesis will provide two examples of this, one of which arises in genetics (see Chapter 2) and the other in political science (see Chapters 3 and 4).

In the final Chapter of this thesis we will shift attention to a different problem: instead of focusing on communities of nodes (genes or individuals) within a network, we will try to summarize a sequence of networks by seeking clusters of networks (see Chapter 5). Although they are still quite uncommon, cross-sectional and temporal sequences of networks have received increasing attention in the last years. When confronted with multiple network instances, one could wish to simplify the problem by seeking for clusters of homogeneous networks, so that attention can then be restricted to the interpretation of the features of each cluster. The method that we propose in Chapter 5 exploits mixtures of generalized linear models to retrieve such clusters.

The remainder of this introduction is organized as follows: in Section 1.2 we will review the concept and some properties of relations, and we will introduce graphs as a convenient way to represent relational data. Then, in Section 1.3 we will shortly make a distinction between the concepts of network and of graph, pointing out how, sometimes, one network can be represented with different types of graphs. Finally, in Section 1.4 we will briefly outline the contents of the upcoming chapters.

1.2 Graphs

1.2.1 Relations and their properties

A *relation* $\mathcal{R}$ from a set $A \neq \emptyset$ to a set $B \neq \emptyset$ is a proposition $r(x, y)$ that is either true or false for any given pair of elements $x \in A$ and $y \in B$. If $r(x, y)$ is true, we say that x is related to y and we write $x\mathcal{R}y$; otherwise, we write $x\not\mathcal{R}y$. A relation is thus a subset of the Cartesian product of A and B, $A \times B$.

A relation can be represented in different ways:

- by listing all the pairs $(x, y) : \ x\mathcal{R}y$ (*extensive* representation);

- with a Euler-Venn diagram, where for every $(x, y) : \ x\mathcal{R}y$ an arrow is drawn from $x \in A$ to $y \in B$;
- with a Cartesian diagram.

A relation can be defined on a single set by letting $A = B$. It is common to classify relations defined on a single set A as

- *reflexive* if $x\mathcal{R}x \ \forall x \in A$;
- *irreflexive* if $x\not\mathcal{R}x \ \forall x \in A$;
- *symmetric* if $x\mathcal{R}y \Rightarrow y\mathcal{R}x \ \forall x, y \in A$;
- *anti-symmetric* if $x\mathcal{R}y \Rightarrow y\not\mathcal{R}x \ \forall x, y \in A$;
- *transitive* if $\ x\mathcal{R}y \wedge y\mathcal{R}z \Rightarrow x\mathcal{R}z \ \forall x, y, z \in A$.

1.2.2 Representing relations with graphs

A *graph* $\mathcal{G}$ is typically[1] defined as a pair (V, E), where V is the set of vertices or nodes and $E \subseteq V \times V$ is the set of edges or links. Thus, it is possible to view a graph as a relation on a single set V, whose extensive representation is nothing but the set E.

Edges in a graph can be directed, or undirected. A directed edge, or arrow, from a node v to another node w indicates that $v\mathcal{R}w$, whereas an undirected edge between nodes v and w denotes that $v\mathcal{R}w \wedge w\mathcal{R}v$.

Graphs can be classified according to the type of edges that they contain. If every edge is undirected, the graph is said to be *undirected*; if, instead, every edge is an arrow, the graph is said to be *directed*. Finally, if both arrows and undirected edges are present, the graph is said to be *mixed* or *partially directed*.

It follows that a symmetric relation can be represented by means of an undirected graph, whereas an anti-symmetric relation is representable as a directed graph. A relation that is not symmetric, nor anti-symmetric can, instead, be represented with a mixed graph.

A *self-loop* is an edge that connects a node to itself. We put a self-loop around $v \in V$ if $v\mathcal{R}v$. Clearly, self-loops are absent in graphs

[1]Note that graphs different from the ones consider here exist. E.g., a *bipartite graph* is a triple (V, W, E) with two sets of vertices V and W and an edge set $E \subseteq V \times W$. Therefore, a bipartite graph is equivalent to a relation on two sets V and W, whose extensive representation corresponds to E.

representing irreflexive relations, whereas they are always present if the graph represents reflexive relations.

1.2.3 The adjacency matrix of a graph

Graphs allow to associate different values to each relation they represent. A distinction, then, can be made between binary graphs, where an edge $e_{ij} = (v_i, v_j)$ can either be present or absent, but every edge has the same intensity, and edge-valued graphs, where edges not only can be present or absent but, if present, they can also have different strength.

Besides Eulero-Venn diagrams, a convenient representation of a graph can be obtained by means of a square matrix called *adjacency matrix*. For a graph with n nodes, the adjacency matrix A is a $n \times n$ matrix whose entries a_{ij} are null if no edge is present from node v_i to node v_j, and non-null otherwise. For binary graphs $a_{ij} \in \{0, 1\}$, whereas for edge-valued graphs $a_{ij} \in \mathbb{R}$.

If no self-loops are present, each diagonal element a_{ii} of A is null.

The adjacency matrix of an undirected graph is symmetric. In this case, attention can be restricted to the upper triangle of A, as the lower one encodes the same information about the graph.

1.3 One network, multiple graphs?

Even though the terms "graph" and "network" are often used interchangeably, they refer to different concepts. A *network* consists of a group of individuals or objects which have relations with each other, whereas a *graph* is the mathematical abstraction that we employ to represent it.

As an example, in Chapter 3 we analyse bill cosponsorship networks: the network, there, is made by the deputies (the members of the Italian Chamber), who relate with each other by cosponsoring bills. It consists of a known set of individuals, the deputies, who interact with each other to discuss and elaborate legislative proposals, and who can eventually decide to cosponsor a bill together. We do not have information on the interactions that take place between the deputies until they cosponsor a bill together. When they do so, they formally state

their agreement on a proposed legislation and, so, we obtain information on their collaboration and joint support to a bill. Thus, we represent the network of cosponsorships with an undirected, edge-valued graph, where the value of an edge is given by the number of cosponsorships that take place between two deputies. Alternatively, one could consider an undirected binary graph to represent the same network, placing an edge between two deputies if they have cosponsored together at least one bill: in this way, the focus is reduced to the presence or absence of a relation between pairs of deputies, and the intensity of relations (when present) is ignored. If specific data on each bill cosponsored were available, two further alternative representations could be considered. First, the process of bill cosponsorship could be represented with a bipartite graph, where one set of nodes V contains the deputies and the other one W consists of the bills that have been subject to cosponsorships. In such a bipartite graph, links e_{ij} would connect a deputy $v_i \in V$ to each of the bills $w_j \in W$ which they have cosponsored. Second, one could represent each cosponsored bill as a clique involving each of the deputies that have cosponsored it. The resulting graph would therefore be defined as a collection of cliques (rather than an edge set as usual) between deputies.

Besides emphasizing the distinction between the network (the set of interactions and relations taking place "in reality") and the graph (or, better, the graphs!), this example also points out that a network can sometimes be represented by more than one graph. Which graph is more suitable for a given statistical analysis depends, clearly, on the scope of each analysis.

Even if they stand for different concepts, nevertheless the words network and graph are often used equivalently without loss of clarity. Awareness of this distinction is however important, as it clarifies that a graph is nothing but a mathematical abstraction that we employ to handle the real phenomenon - making it apparent that the translation of a network into a graph is subject to some simplifications and conventions, and that at the same time a choice between different graph representations is often possible.

1.4 Chapter summaries

1.4.1 Outline of Chapter 2

In Chapter 2 we will present a test that allows to assess the relation between groups of genes in genetic networks. This test is motivated by the need to integrate traditional gene enrichment analysis approaches for the interpretation of microarray experiments with information on known interactions between genes.

Gene enrichment analysis (GEA hereafter) seeks for known sets of genes that can be related to a set of target genes. A known limitation of GEA is that it bases assessment of enrichment on the level of overlap between sets of genes only, ignoring associations and interactions between genes. The role of gene-gene (and protein-protein) interactions in the regulation of cellular processes, however, is at the basis of our current understanding of genetic mechanisms, and should thus be considered as part of enrichment tests. These interactions are typically represented with gene interaction networks, and the integration of genetic networks into GEA, called network enrichment analysis (NEA), has been advocated over the last decade [Shojaie and Michailidis, 2010; Alexeyenko et al., 2012; McCormack et al., 2013].

Existing tests for network enrichment analysis, however, deal only with undirected networks, they can be computationally slow and are based on normality assumptions. In Chapter 2, we propose an alternative Network Enrichment Analysis Test (NEAT) that aims to overcome these limitations. As a matter of fact, NEAT does not require normality assumptions, it is computationally more efficient and it can be applied not only to undirected, but to directed and partially directed networks as well. By means of simulations and real data analyses, we will show that NEAT is considerably faster than alternative resampling-based methods, and that its capacity to detect enrichments is at least as good as the one of alternative tests.

1.4.2 Outline of Chapter 3

In Chapter 3 we will shift our attention from networks in genetics to social networks. In particular, we will study bill cosponsorship networks in the Italian Chamber of Deputies from 2001 to 2015.

The attention of political scientists has traditionally been focused on bill cosponsorship in the US Congress; if compared to it, a distinguishing feature of the Chamber is the presence of a large number of political groups. The primary focus of our analysis will thus be to infer the pattern of collaborations between these groups.

In order to achieve this result, we propose an extension of stochastic blockmodels for the analysis of edge-valued graphs that views bill cosponsorship as the result of a Poisson process, and we derive measures of productivity and collaboration between political parties. We cope with the large number of model parameters by pursuing a penalized likelihood approach, which allows us to infer a sparse reduced graph summarizing collaborations between political parties.

The application of the model allows to point out the evolution from a highly polarized political arena, in which deputies based collaborations on their identification with left or right-wing values, towards an increasingly fragmented Parliament, where a rigid separation of political groups into coalitions does not hold any more, and collaborations beyond the perimeter of coalitions have become possible.

1.4.3 Outline of Chapter 4

In Chapter 4 we will tackle the issue of modelling unobserved sources of nodal heterogeneity within the framework of stochastic blockmodels.

Besides displaying community structures, social networks typically feature also a strong heterogeneity among their actors. For example, in friendship networks it is common to observe that a few individuals are highly popular, whereas most individuals in the network have a smaller number of friends.

Despite their capacity to handle networks with community structure, a major limitation of stochastic blockmodels is that they are based on information on group membership of nodes only and, thus, they fail to model nodal heterogeneity consistently. The extension of stochastic blockmodels which we consider in Chapter 3 already allows to model directly this heterogeneity by including nodal or edge-related covariates. However, sometimes such covariates might not be available, or they could be insufficient to account for all of the observed hetero-

geneity. Therefore, in Chapter 4 we will propose a further extension to the model proposed in Chapter 3, which allows to model possible unobserved sources of heterogeneity by adding a set of nodal random effects to the model. We will also consider latent space models, which are an alternative class of models that allow to model nodal heterogeneity.

1.4.4 Outline of Chapter 5

Whereas in Chapters 2, 3 and 4 the interest lies in inference of relations between communities within a graph, in Chapter 5 we aim at modelling a sequence of graphs, and at providing an efficient strategy to detect cluster of graphs in that sequence.

Although statistical analysis of networks has traditionally focused on modelling relations in a single network, we expect that the collection of multiple instances (either in a cross-sectional or a longitudinal sense) of a network will become common in the near future [Durante et al., 2016; Matias and Miele, 2017].

Even though one could tackle the study of a sequence of graphs by modelling each graph separately, this would result in a cumbersome exercise. As we foresee the possibility that graphs in the sequence could be similar to a certain degree, it seems reasonable to model them jointly. By doing this, one can model the sequence in a more parsimonious way, and at the same time they can borrow information among similar graphs.

Building on the fact that many network models can be implemented within the framework of generalized linear models, in Chapter 5 we will propose to jointly model all the graphs in the sequence by using a mixture of generalized linear models, where each component in the mixture is given by a network model of interest for a given subpopulation of graphs. The proposed model allows to estimate the probability that each graph belongs to a certain subpopulation, and it can thus be employed to cluster the graphs within the sequence. Moreover, it allows to characterize each subpopulation by means of the model estimates of the corresponding component.

We will initially tackle model estimation by implementing the EM (Expectation-Maximization) algorithm, showing that this can some-

times result in a low clustering accuracy. Therefore, we will then propose EMSAGC, an alternative algorithm where we integrate the EM with Simulated Annealing. EMSAGC allows a wider exploration of the likelihood surface than the simple EM, thus resulting in a highly accurate clustering strategy even in the cases where the EM alone fails to retrieve the correct clusters.

Bibliography

Alberts, B., Bray, D., Hopkin, K., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2004). *Essential Cell Biology*. Garland Science.

Alexeyenko, A., Lee, W., Pernemalm, M., Guegan, J., Dessen, P., Lazar, V., Lehtiö, J., and Pawitan, Y. (2012). Network enrichment analysis: extension of gene-set enrichment analysis to gene networks. *BMC Bioinformatics*, 13(1):226.

Barabasi, A.-L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113.

Cranmer, S. J., Heinrich, T., and Desmarais, B. A. (2014). Reciprocity and the structural determinants of the international sanctions network. *Social Networks*, 36:5–22.

Durante, D., Paganin, S., Scarpa, B., and Dunson, D. B. (2016). Bayesian modeling of networks in complex business intelligence problems. *Journal of the Royal Statistical Society. Series C*.

Guimera, R., Mossa, S., Turtschi, A., and Amaral, L. N. (2005). The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences*, 102(22):7794–7799.

Klovdahl, A. S. (1985). Social networks and the spread of infectious diseases: the AIDS example. *Social Science & Medicine*, 21(11):1203–1216.

Matias, C. and Miele, V. (2017). Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society. Series B*.

McCormack, T., Frings, O., Alexeyenko, A., and Sonnhammer, E. (2013). Statistical assessment of crosstalk enrichment between gene groups in biological networks. *PLOS ONE*, 8(1):e54945.

Moreno, J. L. (1934). *Who shall survive?*, volume 58. Nervous and mental disease monograph series.

Sampson, S. F. (1969). *Crisis in a cloister*. PhD thesis, Cornell University, Ithaca.

Shizuka, D., Chaine, A. S., Anderson, J., Johnson, O., Laursen, I. M., and Lyon, B. E. (2014). Across-year social stability shapes network structure in wintering migrant sparrows. *Ecology Letters*, 17(8):998–1007.

Shojaie, A. and Michailidis, G. (2010). Network enrichment analysis in complex experiments. *Statistical Applications in Genetics and Molecular Biology*, 9(1).

Chapter 2

NEAT: an efficient network enrichment analysis test

2.1 Background

The advent of high throughput technologies has driven the development of cell biology over the last decades. The diffusion of microarrays and next generation sequencing techniques has made available a large amount of data that can be used to increase our understanding of gene expression. The need to analyse and interpret these data has led to the development of new methods to infer relationships between genes, which require a combination of biological knowledge, statistical modelling and computational techniques.

When the first data on gene expression became available, they were usually analysed considering each gene separately. However, researchers soon realized that genes act in a concerted manner, and that cellular processes are the result of complex interactions between different genes and molecules. Nowadays, sets of genes that are responsible for many cellular functions have been identified, and are collected in publicly available databases [Ashburner et al., 2000; Kanehisa and Goto, 2000].

One of the advantages of these sets of genes, whose function is already known, is that they can be used to interpret the results of new

Published as:

Signorelli, M., Vinciotti, V., and Wit, E. C. (2016). NEAT: an efficient network enrichment analysis test. *BMC Bioinformatics*, 17:352, DOI: 10.1186/s12859-016-1203-6.

A shorter version of this Chapter was also published in the Proceedings of the 31st International Workshop on Statistical Modelling (vol. 1, pp. 289–294).

experiments: this has led to the implementation of a large number of methods for *gene enrichment analysis* [Huang et al., 2009]. Their aim is to compare gene expression levels under two different conditions (experimental vs control), and to detect which sets of genes are differentially expressed (enriched) in the experimental condition. To this end, genes are ordered in a list L in decreasing order of differential expression, and enrichment is then tested in different ways. *Singular enrichment analysis* [Robinson et al., 2002; Beißbarth and Speed, 2004] tests the over or under-representation of functional gene sets within the set of genes defined by the first k top genes in L. The major limitations of this approach lie in the fact that the choice of k is arbitrary, and that the test does not take into account gene expression levels. *Gene set enrichment analysis* [Subramanian et al., 2005; Kim and Volsky, 2005] overcomes these limitations, by making use of the whole list L of genes, and testing the tendency of genes belonging to a functional set to occupy positions at the top (or at the bottom) of L. A limitation that is common to both single and gene set enrichment analysis, however, is that these methods base computations on the level of overlap between sets of genes only, without considering associations and interactions between genes.

Gene networks are an established tool to represent these interactions. In *network inference* [De Smet and Marchal, 2010; Marbach et al., 2010], genes or molecules are represented as nodes of a graph and their interactions are modelled as links between the nodes. These links can be represented as either a directed or an undirected edge, and a graph is called directed if all edges are directed, undirected if every edge is undirected and partially directed (or mixed) otherwise [Lauritzen, 1996]. An undirected edge displays association between two genes, while a directed edge posits a direction in the relationship between them. Network estimation represents a difficult task, and many different estimation methods have been proposed [Friedman et al., 2008; Abegaz and Wit, 2013]. Marbach et al. [2012] classified them into six groups and pointed out that their predictive performance can vary a lot within each group and according to the structure of the network. In order to integrate evidence on gene associations unveiled by a number of experimental and computational studies into a single network, curated gene networks for different species have been proposed, in-

cluding *YeastNet* [Kim et al., 2013] and *FunCoup* [Schmitt et al., 2014]. In an attempt to integrate the information on interactions between genes provided by gene networks into enrichment analyses, researchers have recently developed methods for *network enrichment analysis* [Shojaie and Michailidis, 2010; Glaab et al., 2012; Alexeyenko et al., 2012; McCormack et al., 2013]. The idea, here, is to test enrichment between sets of genes in a network. Shojaie and Michailidis [2010] focus mainly on network inference, proposing to represent the gene network with a linear mixed model, so that enrichment tests can be then computed by testing a system of linear hypotheses on the fixed effect parameters of the model. Glaab et al. [2012], Alexeyenko et al. [2012] and McCormack et al. [2013], instead, assume that a gene network is already available (either from the literature or as the result of a tailored inferential process) and focus their attention on the strategy that can be used to assess enrichment between sets of nodes. In particular, Glaab et al. [2012] propose a network enrichment score based on a suitably defined network distance between two sets of nodes, alongside an empirical method for setting a cut-off on this distance. In contrast to this, Alexeyenko et al. [2012] and McCormack et al. [2013] derive network enrichment scores on the basis of statistical tests against the null distribution of no enrichment. The advantage of the approach proposed by Alexeyenko et al. and McCormack et al. is that the assessment of enrichment is based on a significance testing procedure.

The idea of Alexeyenko et al. [2012] and McCormack et al. [2013] is that the presence of enrichment between two sets of genes, say A and B, can be assessed by comparing the number of links connecting nodes in A and B with a reference distribution, which models the number of links between the same two sets in the absence of enrichment. Both Alexeyenko et al. [2012] and McCormack et al. [2013] assume that the reference distribution is approximately normal, and they obtain its mean and variance by means of permutations, i.e., computing the mean and variance of the number of links between A and B in a sequence of random replications of the network. Their tests rely on algorithms that permute the network, and mainly differ between themselves for the fact that each algorithm aims to preserve different topological properties of the original network in the generation of network replicates. These methods, however, suffer from three limi-

tations. First of all, they require the simulation of a large number of permuted networks, an activity that can be computationally intensive and highly time consuming (especially for big networks). Furthermore, they base the computation of the test on a normal approximation for the reference distribution, whose nature is discrete. McCormack et al. [2013] show that such an approximation is inaccurate when the expected number of links between A and B is small. A further drawback of these methods is that they have been implemented so far only for undirected networks.
In this work we build upon the approach of Alexeyenko et al. [2012] and McCormack et al. [2013] and propose an alternative test which we call NEAT (Network Enrichment Analysis Test). The main idea behind this test is that, under the null hypothesis of no enrichment, the number of links between two gene sets A and B follows an hypergeometric distribution. This enables us to model the reference distribution directly via a discrete distribution, without having to resort to a normal approximation. NEAT does not require network permutations to compute mean and variance under the null hypothesis, and is therefore faster than the existing resampling-based methods. Moreover, we develop NEAT not only for undirected, but also for directed and partially directed networks, thus providing a common framework for the analysis of different types of networks.

2.2 Methods

The starting point of enrichment analyses is the identification of one or more gene sets of interest. These target gene sets are typically groups of genes that are differentially expressed between experimental conditions, but they can also be different types of gene sets: e.g., clusters of genes that are functionally similar in a given time course, or genes that are bound by a particular protein in a ChIP-chip or ChIP-seq experiment. Enrichment analysis provides a characterization of each target gene set by testing whether some known functional gene sets can be related to it. Methods for gene enrichment analysis assess the relationship between a target gene set and each functional gene set simply by considering the overlap of these two groups. In contrast to this, network enrichment analysis incorporates an evaluation of the

level of association between genes in the target set and genes in the functional gene set into the test.
Information on associations and dependences between genes is represented by a network, which consists of a set of N nodes $V = \{v_1, ..., v_N\}$ that are connected by edges (links). Each gene is thus represented as a node v_i of the network, and a link between two nodes is drawn to signify interaction between the corresponding genes. Examples of genome-wide curated networks that collect known gene associations are *YeastNet* [Kim et al., 2013] and *FunCoup* [Schmitt et al., 2014].
A natural way to study the relation between two sets of genes A and B in a network is to consider the presence or absence of links connecting nodes in the two groups [Alexeyenko et al., 2012; McCormack et al., 2013]. In the inferred network, we expect that individual links may be slightly unstable and noisy. However, we do expect that the inferred links contain a sign of the relationships between gene sets. So, although links between individual genes in sets A and B may be noisy, if there is a functional relationship between functions described by sets A and B we expect the number of links between the two groups to be larger (or smaller) than expected by chance. If this is the case, we say that there is enrichment between A and B.
Links between two nodes of a network can be either directed (arrows) or undirected. The presence of an arrow between two genes implies a directionality in the relation between them, whereas an undirected edge does not provide information on the direction of the relation. The upcoming subsection considers directed networks. In this case, one can distinguish two cases: whether genes in the target set regulate genes of the functional set, or genes in the functional gene set regulate genes in the target set (enrichment from A to B, or from B to A). This distinction does not occur for undirected networks, which are the subject of the next subsection: in this case, A and B are exchangeable, and we simply talk of enrichment "between" A and B. A workflow diagram summarizing the input and the output of NEAT is shown in Figure 2.1.

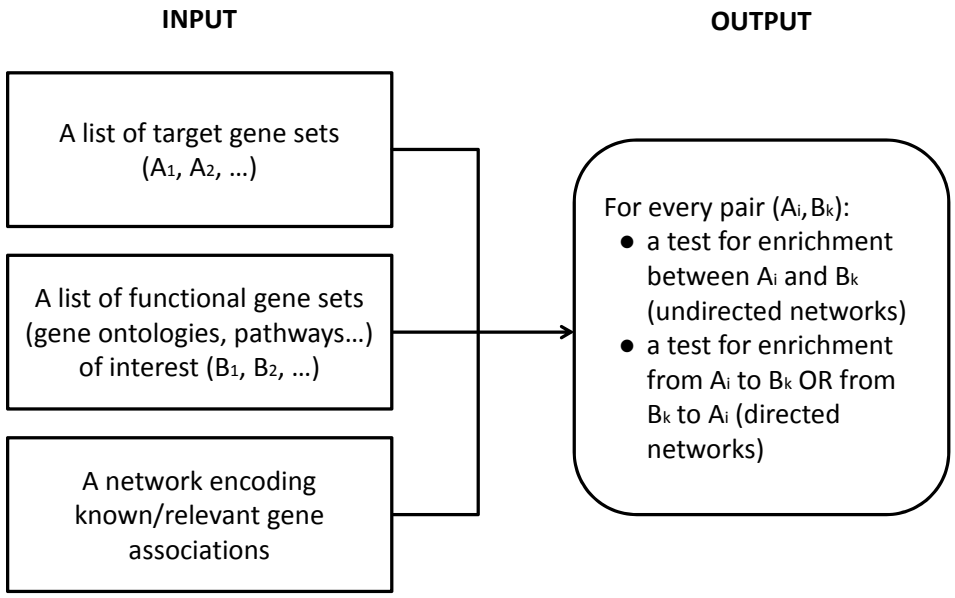


Figure 2.1: **Workflow diagram of a typical network enrichment analysis with NEAT.**

2.2.1 Enrichment test for directed networks

In a directed network, we assess the presence of enrichment from A to B by considering the number of arrows going from genes in A to genes belonging to B. We denote this by n_{AB}. The observed n_{AB} can be thought of as a realization from a random variable N_{AB}, with expected value μ_{AB}. To assess the relation from A to B, we compare μ_{AB} with the number of arrows that we would expect to observe from A to B by chance, which we denote as μ_0. We say that there is enrichment from A to B if μ_{AB} is different from μ_0. Furthermore, we say that there is over-enrichment from A to B if μ_{AB} is higher than μ_0, and under-enrichment (or depletion) if μ_{AB} is lower than μ_0.
We propose a test based on the hypergeometric distribution to assess the significance of this difference. The motivation behind this choice is the following. The hypergeometric distribution models the number of successes in a random sample without replacement: in our case, we can mark arrows in the network that reach genes in B as "successful", and the remaining ones as "unsuccessful". Then, we can view the arrows that go out from genes in A as a random sample without re-

placement from the population of arrows present in the graph: if there is no relation (i.e., no enrichment) between A and B, then the distribution of N_{AB} (the number of successes in the sample) is

$$N_{AB} \sim \text{hypergeom}(n = o_A, K = i_B, N = i_V), \tag{2.1}$$

where the sample size o_A is the outdegree of A (the total number of arrows going out from genes that belong to A), the number of successful cases in the population i_B is the indegree (number of incoming arrows) of B and the population size i_V is the total indegree of the network (which is equal to the total number of arrows).
It is certainly possible to imagine alternative choices for the null distribution of N_{AB}. Alexeyenko et al. [2012] and McCormack et al. [2013] assume that N_{AB} is normal with mean μ_0 and variance σ_0^2, and they use network permutations to estimate μ_0 and σ_0^2. However, the normal distribution is continuous and symmetric, so that their choice implies somehow that the behaviour of N_{AB} should be roughly symmetric, and could be well approximated with a continuous random variable. In addition, estimation of μ_0 and σ_0^2 by means of network permutations can be highly time consuming. Alternatively, one could consider for N_{AB} an hypergeometric distribution with different parameters, defined for example, by considering all possible edges in the network (instead of the edges that are actually present in the network) as a population. We prefer model (2.1) over this alternative, because the choice of the parameters therein allows to condition on two quantities that we consider crucial, which are the outdegree of A and the indegree of B. Moreover, in our experience so far, we have observed that tests based on alternative parametrizations often result in poor performances.
The null mean and variance of N_{AB} can be immediately derived from model (2.1). In particular, in the absence of enrichment we expect to observe, on average, $\mu_0 = o_A \frac{i_B}{i_V}$ arrows from nodes in A to nodes in B. Thus, we expect μ_0 to increase as the number of arrows leaving A, or reaching B, increases. Biological assessment of enrichment can therefore be carried out by testing the null hypothesis of no enrichment

$$H_0 : \mu_{AB} = \mu_0$$

against the alternative hypothesis of enrichment

$$H_1 : \mu_{AB} \neq \mu_0.$$

In a test with a discrete test statistic and two-sided alternative, such as the one that we propose, the p-value can be computed in different ways [Gibbons and Pratt, 1975; Blaker, 2000; Agresti, 2013]. Let T be a discrete test statistic and t be the observed value of T. A first possibility is to compute the p-value for the two-tailed test by doubling the one-tailed p-value, $p_1 = 2\min[P_0(T \leq t), P_0(T \geq t)]$, where P_0 denotes the distribution of T under the null hypothesis. An evident drawback of this formula, however, is that p_1 can exceed 1, and therefore p_1 does not represent a probability. Even though a simple modification $p_2 = \min(p_1, 1)$ could avoid the problem, we prefer to subtract $P_0(T = t)$ from p_1 ($P_0(T = t)$ is non-null for discrete T, and this is the reason why p_1 can exceed 1) and to compute the p-value using

$$\begin{aligned} p &= 2\min[P_0(T < t), P_0(T > t)] + P_0(T = t) \\ &= 2\min\left[P_0(N_{AB} > n_{AB}), P_0(N_{AB} < n_{AB})\right] + P_0(N_{AB} = n_{AB}), \end{aligned} \tag{2.2}$$

which always lies within the interval $[0, 1]$ and differs from p_1 by a factor equal to $P_0(T = t)$. A p-value close to 0 can be regarded as evidence of enrichment, because it entails that the number of links from A to B is significantly smaller or higher than we would expect it to be in the absence of enrichment. Therefore, for a given type I error probability α, we conclude that there is evidence of enrichment from A to B if $p < \alpha$, while if $p \geq \alpha$ there is not enough evidence of enrichment. As an example, consider the network in Figure 2.2. Suppose that we are interested to test whether there is enrichment from the set $A = \{1, 4\}$ to the set $B = \{3, 5, 7\}$. It can be observed that there are 5 arrows going out from A, and 2 of them reach B. The whole network consists of 15 arrows, of which 4 reach B. Thus, $n_{AB} = 2$, $o_A = 5$, $i_B = 4$ and $i_V = 15$. The idea behind (2.1) is that, if the 5 arrows that are going out from A are a random sample (without replacement) from the 15 arrows that are present in the network, then the proportion of arrows reaching B from A should be close to the proportion of arrows reaching B in the whole network, and in the absence of enrichment we should observe on average $\mu_0 = 1.33$ edges. In this case, it seems that

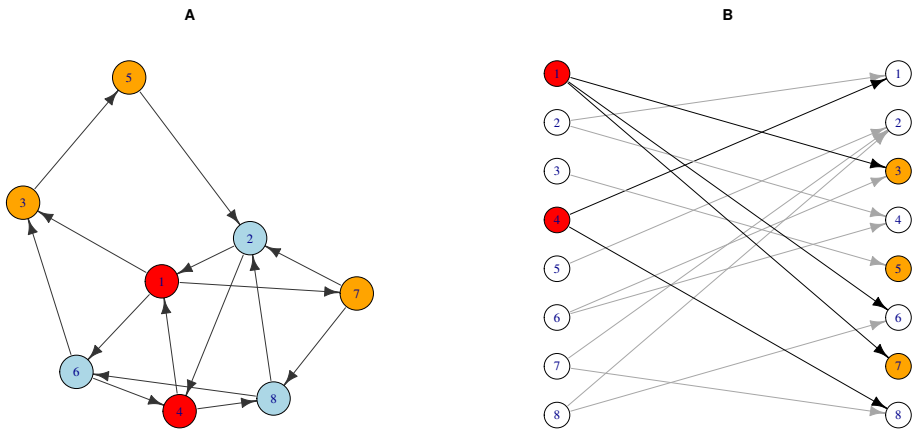


Figure 2.2: **Example: NEAT in directed networks.** *Left:* directed network consisting of 8 nodes connected by 15 arrows. Set A contains nodes 1 and 4 (red) and set B nodes 3, 5 and 7 (orange). *Right:* bipartite representation of the same network: it can be observed that $n_{AB} = 2$, $o_A = 5$, $i_B = 4$ and $i_V = 15$. It follows that $\mu_0 = 1.07$ and $p = 0.48$.

arrows going out from A tend to reach B more frequently (40%) than other arrows do (27% of the 15 arrows in the network reach B). However, the computation of the p-value leads to $p = 0.48$: the observed $n_{AB} = 2$ does not provide enough evidence to reject the null hypothesis, so that the conclusion of the test is that there is no enrichment from A to B.

We can also consider sets $B = \{3, 5, 7\}$ and $C = \{2, 5\}$ (note that the two groups share gene 5), and test enrichment from B to C. In this case, $n_{BC} = 3$ arrows out of $o_B = 4$ (75%) reach C from B, whereas in the whole network $i_C = 4$ arrows out of $d_V = 15$ (27%) reach C. The null expectation is here $\mu_0 = 1.07$; if we fix the type I error probability equal to $\alpha = 5\%$, the p-value $p = 0.03$ leads to the conclusion that there is enrichment from B to C.

2.2.2 Enrichment test for undirected networks

When dealing with undirected networks, the presence of enrichment between A and B is assessed considering the number of edges that connect genes in A to genes in B. We denote this by n_{AB}. Given the undirected nature of the links in the network, there is no distinc-

tion between indegree and outdegree of a node, and it only makes sense to consider the degree of a node, which is the number of vertices that are linked to that node. The null distribution (2.1) should thus be adapted accordingly. Let us define the total degree d_S of a set S as the sum of the degrees of nodes that belong to it: then, in the absence of enrichment we can view n_{AB} as the number of successes in a random sample of size d_A, drawn from a population of size d_V. The null distribution of N_{AB} for undirected networks is thus

$$N_{AB} \sim \text{hypergeom}(n = d_A, K = d_B, N = d_V),$$

where d_A, d_B and d_V are the total degrees of sets A, B and V.
The null hypothesis is then that $\mu_{AB} = \mu_0 = d_A \frac{d_B}{d_V}$, the alternative that $\mu_{AB} \neq \mu_0$. The p-value is computed using formula (2.2).
As an example, consider the network in Figure 2.3A and suppose that we are interested to test the presence of enrichment between the pairs of sets (A, B), (A, C) and (B, C). Sets A and B are linked by $n_{AB} = 4$ edges, and their degrees are $d_A = 4$ and $d_B = 15$, while $d_V = 36$. Thus, $\mu_0 = 1.67$ and $p^{AB} = 0.023$. In the same way, it is possible to compute $p^{AC} = 0.465$ and $p^{BC} = 0.038$. Figure 2.3B shows the relation between the three sets fixing $\alpha = 5\%$: enrichment is present between the pairs (A, B) and (B, C), but not between sets A and C.

2.2.3 Enrichment test for partially directed networks

A partially directed network (or "mixed" network) is a network where both directed and undirected edges are present. It is possible to view such a network as a directed network, where every undirected edge connecting two nodes v and w represents in fact a pair of arrows, the former going from v to w and the latter from w to v. If such an adaptation is adopted, model (2.1) can be applied and partially directed networks can be analysed within `neat` as directed networks.

2.2.4 Software

NEAT is implemented in the R package `neat` [Signorelli et al., 2016], which can be freely downloaded from CRAN: https://cran.r-project.org/package=neat. The manual and a vignette illustrating the package are also available from the same URL. A copy of the vignette

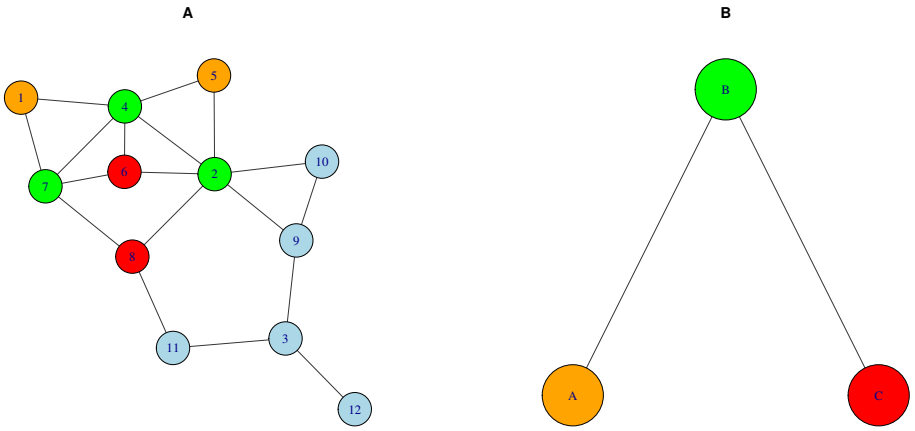


Figure 2.3: **Example: NEAT in undirected networks.** *Left:* undirected network with 12 nodes. We are interested to infer the relation between sets A (nodes 1 and 5), B (2, 4 and 7) and C (6 and 8). *Right:* representation of the relations between sets: enrichment is detected between sets A and B ($p = 0.023$) and between sets B and C ($p = 0.038$), but not between sets A and C ($p = 0.465$).

and manual for version 1.0 of the package can be found in Appendices A and B. The package allows users to specify the network in different formats, it includes functions to plot and summarize the results of the analysis and is accompanied by a set of data and examples, including the enrichment analysis of the ESR gene sets that we discuss in Section 2.4.

2.3 Performance evaluation

We assess the performance of NEAT by means of simulations. Table 2.1 summarizes some aspects of these simulations, which are the subject of the next two subsections. The R scripts and data files for each simulation can be found at https://github.com/m-signo/neat.

We first consider directed networks, and check whether the performance of NEAT is influenced by the degree distribution of the network, or by the level of overlap between sets of nodes. We then consider undirected networks, and carry out a comparison of NEAT with the NEA test of Alexeyenko et al. [2012] and with the LP, LA, LA+S and NP tests of McCormack et al. [2013].

Table 2.1: **An overview of simulations S1-S5.** In Simulations S1 and S2, we compare the performance of NEAT in two directed networks with different degree distribution. In simulation S3, we check the performance of the test for different levels of overlap, ranging from 0% to 100%. In Simulations S4 and S5, we compare NEAT to alternative tests in two undirected networks with different degree distribution.

				Overlap:	
Simulation	Network type	Degree distribution	Graph density	mean	maximum
S1	Directed	Power law	3%	4%	11.3%
S2	Directed	Mixture of 2 Poisson	4%	3.6%	9.5%
S3	Directed	Mixture of 2 Poisson	4%	-	-
S4	Undirected	Power law	3%	3.8%	12%
S5	Undirected	Mixture of 2 Poisson	4%	3.6%	11%

We compare the performance of the methods under the null hypothesis by checking whether the empirical distribution of p-values in the absence of enrichment is uniform using the Kolmogorov-Smirnov test, and by computing the following ratios:

$$R_1 = \frac{\text{Number of enrichments at } 1\% \text{ level}}{0.01 \times \text{Number of tests where } H_0 \text{ is true}}$$

and

$$R_5 = \frac{\text{Number of enrichments at } 5\% \text{ level}}{0.05 \times \text{Number of tests where } H_0 \text{ is true}}.$$

The idea behind R_1 and R_5 is that if the null hypothesis H_0 is true, we expect a good test to reject it with a frequency that is close to α. So, the target value for R_1 and R_5 is 1.
Furthermore, we compare the capacity of different tests to correctly detect enrichments and non-enrichments by computing specificity and sensitivity at $\alpha = 5\%$ level, and the area under the ROC curve (AUC). The specificity is the proportion of correctly detected non-enrichments, and we expect it to be as close as possible to $1 - \alpha$. The sensitivity indicates the proportion of correctly detected enrichments, whereas the AUC is a measure of the overall capacity of a test to discriminate enrichments and non-enrichments across all values of α. Therefore, a test will show a good performance whenever it achieves a specificity close to $1 - \alpha$, and values of sensitivity and AUC as high as possible (ideally 1).

Table 2.2: **Performance of NEAT in simulations S1 and S2.** p^{KS} denotes the p-value of the Kolmogorov-Smirnov test for uniform distribution, AUC is an abbreviation for "area under the ROC curve". In both simulations, the distribution of p-values under H_0 is uniform and the specificity is close to the expected 95% value. Sensitivity and AUC are higher in simulation S2.

Simulation	p^{KS}	R_1	R_5	Sensitivity	Specificity	AUC
S1	0.510	1.56	1.17	73%	94%	0.894
S2	0.125	1.20	1.12	78%	94%	0.927

2.3.1 Simulation with directed networks

In simulations S1 and S2, we generate two random networks with 1000 nodes and with fixed indegree and outdegree distributions using the algorithm implemented by Csardi and Nepusz [2006]. The indegree and outdegree distributions of nodes are power law with exponent 4 and minimum degree 20 in simulation S1, and a mixture of two Poisson distributions, with parameters $\lambda_1 = 40$ and $\lambda_2 = 100$ and weights $q_1 = 99\%$ and $q_2 = 1\%$, in simulation S2.
We consider 50 sets of nodes whose size ranges between 50 and 100, and we test enrichment from A to B and from B to A for every pair of sets: this means that, in total, we compute $50 \times 49 = 2450$ tests. In the original networks, no preferential attachment (i.e., no enrichment) between any couple of these sets is present; we generate enrichments by increasing or reducing the number of arrows for 200 pairs of sets. In each case, enrichment is created by adding or removing arrows randomly from one group to the other, in such a way that n_{AB} increases or reduces by a proportion uniformly ranging from 10% to 50%.
Table 2.2 shows that the empirical distribution of p-values in absence of enrichment is approximately uniform both in simulation S1 and S2. The sensitivity is higher in simulation S2, whereas the specificity is close to the target value (95%) in both cases. As a result, the area under the ROC curve is slightly higher in simulation S2. Overall, the test shows in both cases a good capacity to discriminate enrichments and non-enrichments.

In simulation S3 we check whether the proportion of overlap between sets A and B, that we measure with the Jaccard index

$$J_{AB} = |A \cap B|/|A \cup B|,$$

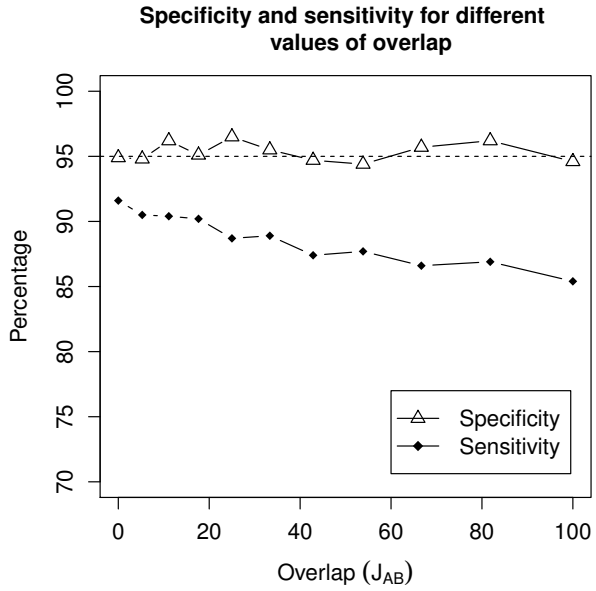


Figure 2.4: **Specificity and sensitivity in simulation S3.** The plot shows the values of specificity and sensitivity for different levels of overlap (every point in the plot is computed on the basis of 1000 tests). We observe that the specificity of the test does not vary substantially for different levels of overlap, and is always close to 95% as expected. The sensitivity, instead, slightly reduces as the percentage of overlap increases.

could have an effect on specificity and sensitivity. We consider the same network used in simulation S2, and we test enrichment between pairs of sets with fixed size $|A| = |B| = 50$, but with increasing overlap (we consider $|A \cap B| \in \{0, 5, 10, 15, ..., 50\}$). Under H_0 we do not modify the network, whereas under H_1 we introduce enrichments adding 35 arrows going from genes in A to genes in B. For every value of overlap, we consider 2000 test (H_0 is true in 1000 cases, and false in the remaining 1000). Figure 2.4 shows that the specificity remains constant and close to 95% for any level of overlap; the sensitivity, on the other hand, is slightly higher when the level of overlap is moderate.

2.3.2 Simulation with undirected networks

As alternative methods for network enrichment analysis are available for undirected networks only, we compare NEAT with them in two simulations where we consider undirected networks with 1000 nodes. We generate two random networks with fixed degree distribution, using the algorithm implemented by Csardi and Nepusz [2006];

Table 2.3: **Results of simulation S4.** The best results for each indicator are in **bold**. p^{KS} denotes the p-value of the Kolmogorov-Smirnov test for uniform distribution, AUC is an abbreviation for "area under the ROC curve". The distribution of p-values under H_0 is evidently not uniform for NEA and LP. NEAT shows the highest values of sensitivity and AUC, and its specificity is close to the target value (95%).

Test	p^{KS}	R_1	R_5	Sensitivity	Specificity	AUC
NEAT	**0.399**	1.33	**1.14**	**69%**	**94%**	**0.920**
NEA	0.001	0	**0.87**	**68%**	**96%**	**0.918**
LP	0	2.13	1.51	**68%**	92%	0.908
LA	**0.255**	1.60	1.17	60%	**94%**	0.897
LA+S	**0.409**	1.87	1.17	63%	**94%**	0.913
NP	0.037	**1.24**	1.28	58%	**94%**	0.884

the degree distribution follows a power law in simulation S4 and a mixture of Poisson distributions in simulation S5, with the same parameters used in simulations S1 and S2. Likewise, we consider 50 sets of nodes, whose sizes vary between 50 and 100 nodes. We test enrichment between every pair of sets A and B, so that the total number of comparisons is here $50 \times 49/2 = 1225$. We introduce enrichments for 100 pairs of sets by adding or removing edges randomly between them, in such a way that n_{AB} is increased or reduced by a proportion uniformly ranging from 10% to 50%.

Tables 2.3 and 2.4 show the results for simulations S4 and S5, respectively. As concerns the behaviour under the null hypothesis, the distribution of p-values is uniform in both cases for NEAT and LA, and in one case for LA+S (simulation S4) and NP (S5). NEA and LP, instead, do not produce uniform distributions: as it can be observed from Figure 2.5, the reason is that the distribution is strongly left-skewed for NEA, whereas for LP the distribution is right-skewed (the same patterns occur also in simulation S5). In both simulations, most of the methods achieve a specificity close to 95% as expected; comparison with the other tests shows that the sensitivity and AUC of NEAT are overall good.

Table 2.5 compares the speed of computation for the different methods. NEAT turns out to be the fastest method by far, being 22 times faster than NP (the fastest alternative) and more than 3000 times faster than NEA (the slowest alternative). This result is mostly due to the fact that NEAT does not require the generation of a large number of permuted

Table 2.4: **Results of simulation S5.** The best results for each indicator are in **bold**. p^{KS} denotes the p-value of the Kolmogorov-Smirnov test for uniform distribution, AUC is an abbreviation for "area under the ROC curve". The distribution of p-values under H_0 can be considered uniform for NEAT, LA and NP, and is questionable for LA+S. NEAT shows the highest values of sensitivity and AUC, and its specificity is exactly equal to the target value (95%).

Test	p^{KS}	R_1	R_5	Sensitivity	Specificity	AUC
NEAT	**0.343**	0.62	**0.98**	**79**%	**95**%	**0.925**
NEA	0.024	0	0.82	73%	96%	0.912
LP	0	1.33	1.51	**78**%	92%	0.904
LA	**0.111**	**1.16**	1.33	73%	93%	0.908
LA+S	0.024	**1.16**	1.13	76%	94%	0.910
NP	**0.323**	1.42	1.16	70%	94%	0.908

Table 2.5: **Speed comparison.** The table compares the time (in seconds) that each method required to compute 1225 tests for enrichment in simulations S4 and S5, using a processor with 2.5 GhZ CPU frequency. NEAT turns out to be by far the fastest method.

Test	Software	Simulation S4	Simulation S5
NEAT	R package `neat`	0.6	0.7
NEA	R package `neaGUI`	2125.4	2151.5
LP	CrossTalkZ	28.6	44.7
LA	CrossTalkZ	14.4	18.0
LA+S	CrossTalkZ	21.8	27.6
NP	CrossTalkZ	12.9	15.8

networks to compute the test.

2.4 Network enrichment analysis: an application to yeast

The budding yeast *Saccharomyces cerevisiae* is a unicellular eukaryote organism that can be easily grown in laboratory. Because of these features, it represents a model organism that has been extensively studied, and it was the first eukaryote whose genome was completely sequenced [Goffeau et al., 1996]. Since then, a large number of studies has aimed to detect associations between genes. In an attempt to collect these results into a unique source, Kim et al. [2013] developed *YeastNet*, an undirected gene network that aims to integrate the results

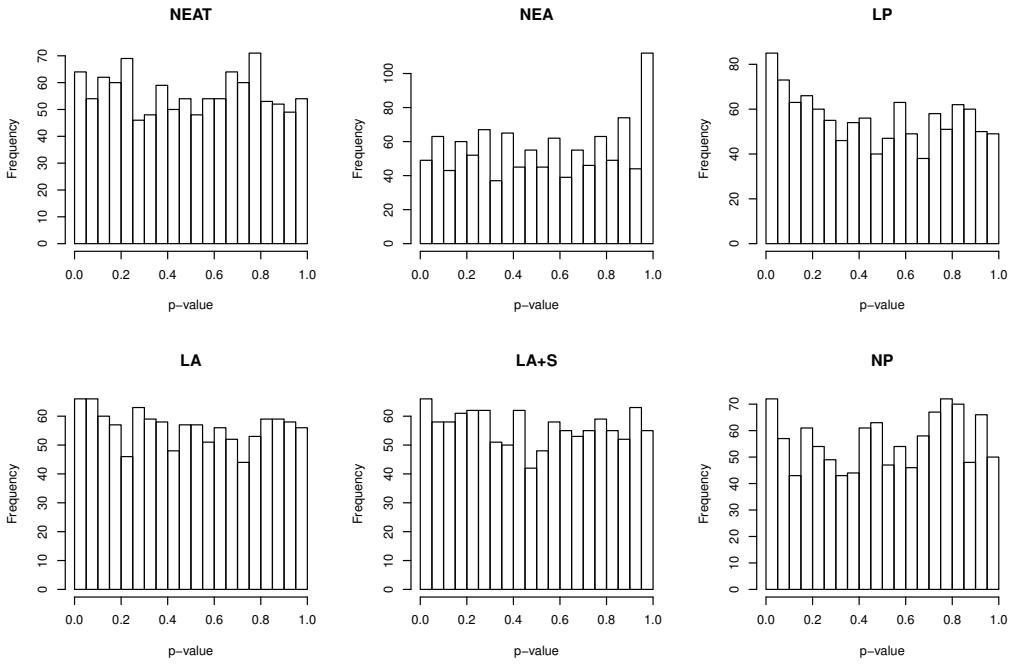


Figure 2.5: **Histogram of p-values in absence of enrichment in simulation S4.** The test of Kolmogorov-Smirnov indicates that the distribution is uniform for NEAT ($p = 0.34$), LA ($p = 0.11$) and NP ($p = 0.32$). The distribution of p-values is highly left-skewed for NEA, and right-skewed for LP.

of a large number of high-throughput studies on Saccharomyces cerevisiae. In its most recent version (v3), YeastNet comprises 362512 edges connecting 5808 genes. We use this network of known associations in the following analyses.

2.4.1 Network enrichment analysis of environmental stress response in yeast

After analysing gene expression patterns of yeast Saccharomyces cerevisiae in response to different stressful stimuli, Gasch et al. [2000] inferred the existence of a set of 868 genes that reacted in a similar way to different, hostile environmental changes. This set of genes, called *Environmental Stress Response* (ESR), is believed to constitute a coordinated, initial reaction to the emergence of any hostile condition in the cell. It consists of two subgroups of genes, containing genes that are repressed and induced under stressful conditions, respectively.
We take these two gene sets as target sets, and for each of them we test enrichment with the following functional gene sets: 99 gene sets

that are part of the GO Slim biological process ontology (we do not consider the groups "biological process" and "other" in the analysis) and 106 known KEGG pathways.
At $\alpha = 1\%$ level, NEAT detects over-enrichment between 23 GO Slim sets and the set of repressed genes, and between 25 GO Slim sets and the set of induced genes. Furthermore, 15 KEGG pathways are found to be over-enriched with the set of repressed ESR genes, and 47 with the set of induced genes.
Gasch et al. [2000] reports that genes that are repressed in the ESR are involved in growth related processes, various aspects of RNA metabolism, nucleotide biosynthesis, secretion, encoding of ribosomal proteins and other metabolic processes. These results are in strong agreement with the list of over-enrichments detected by NEAT, shown in Table 2.6. As a matter of fact, most of the over-enrichments detected by NEAT are related to RNA transcription, nucleotide secretion and translation of ribosomal proteins (rows 1-18 and 24-35 in Table 2.6), growth-related processes (row 22) and further metabolic processes (rows 23 and 33-35).
Gasch et al. [2000] observed that inference for the set of genes that are induced by the ESR is more complicated, because most of the genes in this group lack functional annotations. It is worthwhile to observe that NEAT detects a large number of enriched KEGG pathways (47 out of 106). This preliminary observation points out a major feature of the Environmental Stress Response: the cell reacts to the emergence of different hostile conditions by activating a number of known cellular pathways that involve energy production, metabolic reactions and molecular transportation (see Table 2.8).
Our results for this gene set do not only match the ones of the original study - identifying many processes and pathways that are related to carbohydrate metabolism (rows 1-3 in Table 2.7 and 1-9 in Table 2.8), fatty acid metabolism (rows 4-6 in Table 2.7 and 10-18 in Table 2.8), mitochondrial functions and cellular redox reactions (rows 5-9 in Table 2.7 and 19-21 in Table 2.8), protein folding and degradation (10 in Table 2.7 and 22 in Table 2.8) and cellular protection during stressful conditions (rows 11-13 in Table 2.7 and 23 in Table 2.8) - but they also unveil further enrichments that involve molecular transportation (rows 3, 6, 14-18 in Table 2.7) and amino-acid metabolism (rows 24-36

in Table 2.8).
Tables 2.9, 2.10 and 2.11 compare the p-values obtained with NEAT with those obtained with LA+S [McCormack et al., 2013], which, according to the conclusions of McCormack et al. [2013] and to our own simulations, can be considered as the main competitor of NEAT. The tables show a large overlap between the over-enrichments detected by the two methods at a 1% significance level: the two methods jointly detect 34 over-enrichments (19 GO Slim sets and 15 KEGG pathways) for the set of repressed ESR genes, and 67 (24 GO Slim sets and 43 KEGG pathways) for the set of induced ESR genes. There is only a small number of discrepancies between the two methods and these are mostly borderline cases. In particular, LA+S detects 4 over-enrichments that are not detected by NEAT (rows 39 in Table 2.9, 26-27 in Table 2.10 and 48 in Table 2.11), whereas NEAT detects 9 over-enrichments that are not detected by LA+S (rows 19-22 in Table 2.9, 25 in Table 2.10 and 43-46 in Table 2.11). As concerns computing time, NEAT computed the required task (410 tests in total) in 23 seconds, whereas the same computation with LA+S required 1171 seconds. In summary, the two methods lead to very similar conclusions, but NEAT is considerably more efficient.

2.4.2 Network enrichment analysis of GO Slim sets: overlap does not imply enrichment

Gene ontologies [Ashburner et al., 2000] consist of a large number of gene sets, which are involved in different cellular functions or biological processes, or that are active in a specific component of the cell. These sets of genes are typically employed to enrich sets of differentially expressed genes that have been experimentally detected (the analysis of the ESR gene sets in the previous subsection provides an example of this). However, network enrichment analysis is a more general instrument, which allows to assess the relation between pairs of gene sets in a network. One might wonder, for instance, whether gene sets within an ontology tend to be strongly related to each other, or whether there is a strong separation between them.
We consider gene sets in the GO Slim biological process ontology for *Saccharomyces cerevisiae* (we once more exclude the two general

groups "biological process" and "other" from the analysis). As a result of the hierarchical structure of Gene Ontologies, 12 gene sets are nested within another group. We exclude these 12 sets from the analysis: the remaining 87 gene sets do not have hierarchical relations with each other, and pairs of these sets display overall a low overlap (1.7 % on average), which is null in most cases (62% of pairs of sets do not share genes). If overlapping of sets was taken by itself as evidence of a relation between two gene sets, one would therefore conclude that most of these gene sets are unrelated.
If, however, we do not limit our attention to the overlap between pairs of sets, but consider also known associations between genes in the two sets as represented in YeastNet [Kim et al., 2013], we obtain a different conclusion. We have used NEAT to test whether there is enrichment between each pair of sets. In a random network where no relations between the sets are present, we would expect to detect 37 enrichments (on average) out of 3741 tests for $\alpha = 1\%$; instead, we detect 1409 enrichments, 38 times more than expected. Out of these, 710 are under-enrichments, and 699 are over-enrichments. An under-enrichment, here, indicates that two GO Slim sets are poorly connected to each other: the high number of under-enrichments, therefore, might be not particularly surprising or interesting, as we do expect that unrelated gene sets within the ontology are poorly connected. The high number of over-enrichments, on the other hand, is striking: this indicates that many groups within the ontology are highly connected to each other - something that would occur rather rarely, if there was no relation between the sets.
This result points out a major difference between gene enrichment analysis and network enrichment analysis: whereas in the first case the extent of overlapping between two gene sets is taken by itself as evidence of enrichment, network enrichment analysis bases the evaluation of enrichment on the level of connectivity that exists between the two sets in a network. Of course, the two facts are not completely unrelated. Figure 2.6 shows that there is a certain correlation between overlap of gene sets (Jaccard index) and network enrichment, so that we tend to find network enrichment in the presence of higher levels of overlap. This correlation is, however, low (the Pearson correlation coefficient between J_{AB} and p^{AB} is -0.15), pointing out that there does

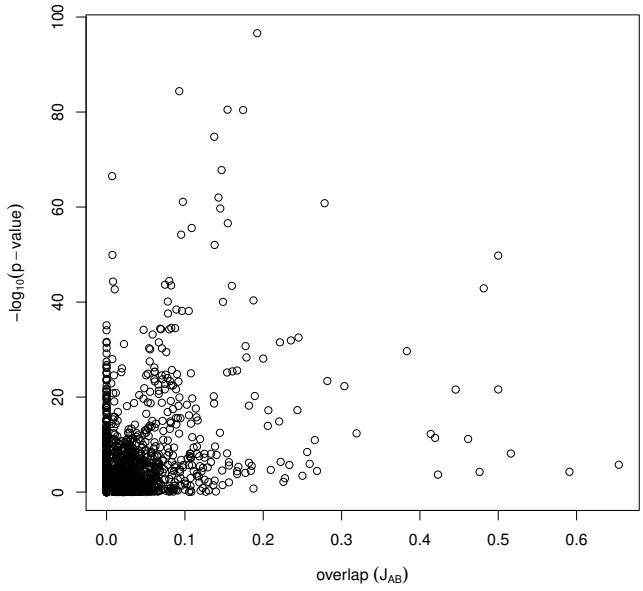


Figure 2.6: **Relation between overlap (J_{AB}) and p-values.** Note that p-values are represented on a negative log-scale to enhance readability.

not necessarily have to be enrichment for highly overlapping gene sets, and vice versa. As an example, the GO Slim sets "cytokinesis" and "nuclear organization" do not share genes, but are detected as enriched ($p = 0.0003$) in YeastNet. This result can be explained by the fact that "nuclear organization" includes genes involved in the assembly and disassembly of the nucleus, which is a preliminary step in cell cytokinesis.

2.5 Conclusion

Network enrichment analysis is a powerful extension of traditional methods of gene enrichment analysis, that allows to integrate them with the information on connectivity between genes provided by genetic networks. Whereas gene enrichment analysis bases the test for enrichment solely on the overlap between two gene sets and ignores the relationships between individual genes, network enrichment analysis exploits information on gene-gene interactions by making use of gene networks, and it is thus capable to detect enrichment even between two gene sets that do not share genes.

In this Chapter, we have presented a Network Enrichment Analysis Test (NEAT) that aims to overcome some limitations which affect the network enrichment tests of Alexeyenko et al. [2012] and McCormack et al. [2013]. First of all, we believe that a normal approximation does not make justice to the discrete nature of N_{AB}. We have shown that this approximation can be avoided if one models N_{AB} directly, using a hypergeometric distribution with suitably specified parameters. In addition, the normal approximation employed by Alexeyenko et al. [2012] and McCormack et al. [2013] requires the computation of a large number of network permutations to obtain the mean and variance under H_0: this operation can be very time consuming for big networks and it makes the computation of the test rather slow. The use of the hypergeometric distribution, instead, allows to specify the null distribution of N_{AB} without resorting to permutations, thus speeding up computations considerably. A further drawback of existing methods for network enrichment analysis [Shojaie and Michailidis, 2010; Glaab et al., 2012; Alexeyenko et al., 2012; McCormack et al., 2013] is that they have been implemented only for undirected networks. We address this problem by considering different types of networks (directed, undirected and partially directed) and by proposing two different parametrizations, which take into account the different nature of directed and undirected links.

We believe that NEAT could constitute a flexible and computationally efficient test for network enrichment analysis. Our simulations show that NEAT has a good capacity to correctly classify enrichments and non-enrichments. Comparison of NEAT with other methods points out an overall good performance in terms of sensitivity and of specificity, as well as the computational efficiency of the proposed method. The examples illustrated in the previous Section show that NEAT can retrieve enrichments that were detected with gene enrichment analysis, but it can also unveil further enrichments that would be overlooked, if known associations between genes were ignored. Furthermore, the comparison with the LA+S test of McCormack et al. [2013], which we take as gold standard among pre-existing tests for network enrichment analysis, points out that NEAT and LA+S yield almost identical conclusions, but NEAT is considerably faster (23 seconds vs 19.5 minutes) in producing them.

Even though the focus of this work is on gene regulatory networks, we remark that NEAT is a rather general test: it can be applied to networks that arise in different contexts and disciplines, whenever the interest is to infer the relationship between groups of vertices. This can include, for example, other types of biological networks, as well as social, economic or technological networks.

Bibliography

Abegaz, F. and Wit, E. (2013). Sparse time series chain graphical models for reconstructing genetic networks. *Biostatistics*, 14(3):586–599.

Agresti, A. (2013). *Categorical Data Analysis*. Wiley, Hobocken.

Alexeyenko, A., Lee, W., Pernemalm, M., Guegan, J., Dessen, P., Lazar, V., Lehtiö, J., and Pawitan, Y. (2012). Network enrichment analysis: extension of gene-set enrichment analysis to gene networks. *BMC Bioinf.*, 13(1):226.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene Ontology: tool for the unification of biology. *Nat. Genet.*, 25(1):25–29.

Beißbarth, T. and Speed, T. P. (2004). Gostat: find statistically overrepresented gene ontologies within a group of genes. *Bioinformatics*, 20(9):1464–1465.

Blaker, H. (2000). Confidence curves and improved exact confidence intervals for discrete distributions. *Can. J. Stat.*, 28(4):783–798.

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695:1–9.

De Smet, R. and Marchal, K. (2010). Advantages and limitations of current network inference methods. *Nat. Rev. Microbiol.*, 8(10):717–729.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., and Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, 11(12):4241–4257.

Gibbons, J. D. and Pratt, J. W. (1975). P-values: interpretation and methodology. *Am. Stat.*, 29(1):20–25.

Glaab, E., Baudot, A., Krasnogor, N., Schneider, R., and Valencia, A. (2012). Enrichnet: network-based gene set enrichment analysis. *Bioinformatics*, 28(18):i451–i457.

Goffeau, A., Barrell, B., Bussey, H., Davis, R., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J., Jacq, C., Johnston, M., et al. (1996). Life with 6000 genes. *Science*, 274(5287):546–567.

Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2009). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.*, 37(1):1–13.

Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, 28(1):27–30.

Kim, H., Shin, J., Kim, E., Kim, H., Hwang, S., Shim, J. E., and Lee, I. (2013). YeastNet v3: a public database of data-specific and integrated functional gene networks for Saccharomyces cerevisiae. *Nucleic Acids Res.*, 42(D1):D731–D736.

Kim, S.-Y. and Volsky, D. J. (2005). Page: parametric analysis of gene set enrichment. *BMC Bioinf.*, 6(1):144.

Lauritzen, S. L. (1996). *Graphical Models*. Oxford Univ Press, Oxford.

Marbach, D., Costello, J. C., Küffner, R., Vega, N. M., Prill, R. J., Camacho, D. M., Allison, K. R., Kellis, M., Collins, J. J., Stolovitzky, G., et al. (2012). Wisdom of crowds for robust gene network inference. *Nat. Methods*, 9(8):796–804.

Marbach, D., Prill, R. J., Schaffter, T., Mattiussi, C., Floreano, D., and Stolovitzky, G. (2010). Revealing strengths and weaknesses of methods for gene network inference. *Proc. Natl. Acad. Sci. U. S. A.*, 107(14):6286–6291.

McCormack, T., Frings, O., Alexeyenko, A., and Sonnhammer, E. (2013). Statistical assessment of crosstalk enrichment between gene groups in biological networks. *PLoS One*, 8(1):e54945.

Robinson, M. D., Grigull, J., Mohammad, N., and Hughes, T. R. (2002). Funspec: a web-based cluster interpreter for yeast. *BMC Bioinf.*, 3(1):35.

Schmitt, T., Ogris, C., and Sonnhammer, E. L. (2014). Funcoup 3.0: database of genome-wide functional coupling networks. *Nucleic Acids Res.*, 42(D1):D380–D388.

Shojaie, A. and Michailidis, G. (2010). Network enrichment analysis in complex experiments. *Stat. Appl. Genet. Mol. Biol.*, 9(1).

Signorelli, M., Vinciotti, V., and Wit, E. C. (2016). neat: efficient Network Enrichment Analysis Test (R package).

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci.*, 102(43):15545–15550.

Table 2.6: **Network enrichment analysis of the repressed ESR gene set.** The table lists the 23 Go Slim BP gene sets and the 15 KEGG pathways which the set of repressed ESR genes is found to be over-enriched with at 1% significance level.

	Gene set	n_{AB}	μ_0	$log_{10}p$
	Go Slim BP sets:			
1	cytoplasmic translation	6878	2641.9	<-300
2	ribosomal large subunit biogenesis	3408	1097.8	<-300
3	ribosomal small subunit biogenesis	5861	2073.7	<-300
4	ribosome assembly	1782	621.9	<-300
5	RNA modification	2944	1062.0	<-300
6	rRNA processing	9187	3290.2	<-300
7	tRNA processing	2037	901.0	<-300
8	translational elongation	1786	782.3	-283.8
9	ribosomal subunit export from nucleus	1420	561.4	-281.8
10	translational initiation	939	462.5	-112.1
11	transcription from RNA polymerase III promoter	565	228.4	-107.7
12	snoRNA processing	634	303.3	-82.0
13	regulation of translation	1952	1328.6	-73.5
14	DNA-dependent transcription, termination	774	447.0	-57.5
15	transcription from RNA polymerase I promoter	1005	646.4	-49.5
16	protein alkylation	1063	759.4	-31.4
17	tRNA aminoacylation for protein translation	400	233.1	-29.4
18	peptidyl-amino acid modification	1088	883.0	-13.2
19	nuclear transport	3154	2003.5	-162.4
20	organelle assembly	2090	1362.7	-96.1
21	nucleobase-containing compound transport	1453	1155.4	-20.8
22	cytokinesis	1024	806.9	-16.0
23	vitamin metabolic process	325	274.0	-3.1
	KEGG pathways:			
24	Ribosome biogenesis in eukaryotes	9824	3661.0	<-300
25	Ribosome	18640	8731.7	<-300
26	RNA polymerase	3057	1541.2	<-300
27	RNA transport	4341	2906.4	-177.6
28	Aminoacyl-tRNA biosynthesis	1433	960.9	-58.2
29	RNA degradation	2560	1939.3	-51.9
30	mRNA surveillance pathway	1768	1413.5	-24.0
31	Pentose phosphate pathway	1126	947.1	-9.7
32	Spliceosome	2649	2523.6	-2.3
33	Purine metabolism	5579	3623.0	-263.6
34	Pyrimidine metabolism	4541	2884.5	-234.9
35	Cyanoamino acid metabolism	218	158.8	-6.3
36	One carbon pool by folate	541	392.5	-15.0
37	Sulfur relay system	238	196.5	-2.9
38	Carbapenem biosynthesis	117	89.8	-2.7

Table 2.7: **Network enrichment analysis of the induced ESR gene set (GO Slim sets).** The table lists the 25 Go Slim BP gene sets which the set of induced ESR genes is found to be over-enriched with at 1% significance level.

	GO Slim BP gene set	n_{AB}	μ_0	$log_{10}p$
1	carbohydrate metabolic process	1296	671.2	-110.9
2	oligosaccharide metabolic process	442	165.3	-77.3
3	carbohydrate transport	202	65.8	-45.0
4	lipid metabolic process	693	484.4	-19.9
5	peroxisome organization	181	124.8	-6.0
6	lipid transport	120	79.7	-4.9
7	generation of precursor metabolites and energy	585	294.8	-54.0
8	cellular respiration	210	118.4	-14.5
9	proteolysis involved in cellular protein catabolic proc.	639	488.5	-10.9
10	protein folding	476	296.9	-22.7
11	response to oxidative stress	813	242.2	-202.7
12	response to chemical stimulus	1489	885.1	-83.4
13	response to starvation	459	331.4	-11.2
14	transmembrane transport	910	644.4	-24.2
15	endocytosis	395	245.5	-19.3
16	protein targeting	628	478.8	-10.9
17	ion transport	464	380.2	-4.8
18	amino acid transport	137	109.4	-2.1
19	cofactor metabolic process	523	219.0	-73.7
20	nucleobase-containing small molecule metabolic proc.	722	404.5	-49.2
21	membrane invagination	278	120.6	-37.0
22	vacuole organization	335	200.2	-18.9
23	protein maturation	49	27.7	-3.9
24	cell morphogenesis	113	79.4	-3.6
25	sporulation	352	306.4	-2.1

Table 2.8: **Network enrichment analysis of the induced ESR gene set (KEGG pathways)**.The table lists the 47 KEGG pathways which the set of induced ESR genes is found to be over-enriched with at 1% significance level.

	KEGG pathway	n_{AB}	μ_0	$log_{10}p$
1	Starch and sucrose metabolism	1436	394.2	<-300
2	Pentose and glucuronate interconversions	414	110.7	-119.9
3	Glycolysis / Gluconeogenesis	1235	616.3	-116.5
4	Fructose and mannose metabolism	562	200.0	-106.7
5	Galactose metabolism	511	173.9	-104.5
6	Amino sugar and nucleotide sugar metabolism	567	264.2	-63.4
7	Other glycan degradation	79	11.7	-44.2
8	Pyruvate metabolism	633	355.9	-42.8
9	Propanoate metabolism	189	107.3	-12.9
10	Glycerolipid metabolism	444	172.1	-72.7
11	Peroxisome	633	313.3	-61.2
12	Fatty acid degradation	419	215.0	-37.2
13	Arachidonic acid metabolism	117	36.7	-28.1
14	Sphingolipid metabolism	227	103.6	-27.3
15	Glycerophospholipid metabolism	450	270.9	-24.5
16	alpha-Linolenic acid metabolism	69	27.1	-11.7
17	Fatty acid elongation	138	75.3	-10.8
18	Biosynthesis of unsaturated fatty acids	134	103.9	-2.5
19	Glutathione metabolism	467	204.8	-59.9
20	Citrate cycle (TCA cycle)	487	267.3	-35.6
21	Ubiquinone and other terpenoid-quinone biosynthesis	96	41.8	-13.1
22	Protein processing in endoplasmic reticulum	1121	866.0	-17.4
23	Longevity regulating pathway	987	544.0	-70.6
24	beta-Alanine metabolism	397	104.0	-118.0
25	Taurine and hypotaurine metabolism	132	24.3	-59.4
26	Tyrosine metabolism	382	163.5	-51.8
27	Tryptophan metabolism	292	113.3	-48.2
28	Valine, leucine and isoleucine degradation	276	107.5	-45.3
29	Alanine, aspartate and glutamate metabolism	488	262.2	-38.0
30	Histidine metabolism	267	127.4	-28.8
31	Arginine and proline metabolism	301	154.3	-27.0
32	Lysine degradation	294	150.4	-26.6
33	Phenylalanine metabolism	171	71.4	-25.0
34	Glycine, serine and threonine metabolism	350	264.3	-6.7
35	Cysteine and methionine metabolism	338	285.3	-2.8
36	Arginine biosynthesis	167	134.0	-2.4
37	Butanoate metabolism	460	84.8	-202.8
38	Pentose phosphate pathway	604	288.0	-64.0
39	Regulation of autophagy	303	126.7	-43.3
40	Insulin resistance	337	172.8	-30.1
41	Glyoxylate and dicarboxylate metabolism	368	201.6	-27.3
42	Methane metabolism	435	254.2	-26.2
43	Nicotinate and nicotinamide metabolism	154	99.8	-6.7
44	Nitrogen metabolism	88	52.8	-5.4
45	Thiamine metabolism	57	32.9	-4.1
46	Selenocompound metabolism	122	89.3	-3.2
47	Sulfur metabolism	133	105.3	-2.2

Table 2.9: **Repressed ESR gene set: comparison between NEAT and LA+S.** The table reports the gene sets that are found to be over-enriched ($\alpha = 1\%$) by at least one of the two methods. μ_0 denotes the expected value of N_{AB} in the absence of enrichment. The last two columns report log_{10} p-values for the proposed NEAT and the LA+S test of McCormack et al. [2013], respectively.

		μ_0		$\log_{10} p$	
	Gene set	NEAT	LA+S	NEAT	LA+S
	GO Slim BP sets:				
1	cytoplasmic translation	2641.9	3583.5	<-300	-290.9
2	ribosomal large subunit biogenesis	1097.8	1602.4	<-300	-269.2
3	ribosomal small subunit biogenesis	2073.7	3013.2	<-300	-236.8
4	ribosome assembly	621.9	872.1	<-300	-95.9
5	RNA modification	1062.0	1422.7	<-300	-213.7
6	rRNA processing	3290.2	4623.2	<-300	<-300
7	tRNA processing	901.0	1137.6	<-300	-103.3
8	translational elongation	782.3	1019.5	-283.8	-71.2
9	ribosomal subunit export from nucleus	561.4	693.4	-281.8	-151.2
10	nuclear transport	2003.5	2452.5	-162.4	-33.0
11	translational initiation	462.5	594.8	-112.1	-33.6
12	transcription from RNA polymerase III promoter	228.4	281.6	-107.7	-43.6
13	organelle assembly	1362.7	1719.2	-96.1	-8.0
14	snoRNA processing	303.3	349.8	-82.0	-26.5
15	regulation of translation	1328.6	1577.5	-73.5	-12.9
16	DNA-dependent transcription, termination	447.0	575.2	-57.5	-11.7
17	transcription from RNA polymerase I promoter	646.4	874.2	-49.5	-5.2
18	tRNA aminoacylation for protein translation	233.1	256.7	-29.4	-11.2
19	protein alkylation	759.4	1000.0	-31.4	-1.2
20	nucleobase-containing compound transport	1155.4	1445.1	-20.8	-0.1
21	cytokinesis	806.9	925.9	-16.0	-1.8
22	peptidyl-amino acid modification	883.0	1102.4	-13.2	-0.1
23	vitamin metabolic process	274.0	245.8	-3.1	-5.5
	KEGG pathways:				
24	Ribosome biogenesis in eukaryotes	3661.0	5212.5	<-300	<-300
25	Ribosome	8731.7	11954.0	<-300	-283.3
26	RNA polymerase	1541.2	2058.0	<-300	-76.1
27	Purine metabolism	3623.0	4136.9	-263.6	-66.9
28	Pyrimidine metabolism	2884.5	3402.5	-234.9	-61.0
29	RNA transport	2906.4	3193.2	-177.6	-75.4
30	Aminoacyl-tRNA biosynthesis	960.9	934.2	-58.2	-49.8
31	RNA degradation	1939.3	2051.3	-51.9	-19.9
32	mRNA surveillance pathway	1413.5	1477.3	-24.0	-12.7
33	One carbon pool by folate	392.5	344.2	-15.0	-19.5
34	Pentose phosphate pathway	947.1	979.2	-9.7	-4.6
35	Cyanoamino acid metabolism	158.8	132.2	-6.3	-7.2
36	Sulfur relay system	196.5	172.7	-2.9	-3.9
37	Carbapenem biosynthesis	89.8	75.1	-2.7	-4.1
38	Spliceosome	2523.6	2432.2	-2.3	-4.1
39	Synthesis and degradation of ketone bodies	39.8	29.8	-0.3	-2.2

Table 2.10: **Induced ESR gene set: comparison between NEAT and LA+S (GO Slim sets)**. The table reports the gene sets that are found to be over-enriched ($\alpha = 1\%$) by at least one of the two methods. μ_0 denotes the expected value of N_{AB} in the absence of enrichment. The last two columns report log_{10} p-values for the proposed NEAT and the LA+S test of McCormack et al. [2013], respectively.

		μ_0		$\log_{10} p$	
	GO Slim BP set	NEAT	LA+S	NEAT	LA+S
1	response to oxidative stress	242.2	248.5	-202.7	-253.7
2	carbohydrate metabolic process	671.2	663.9	-110.9	-123.3
3	response to chemical stimulus	885.1	912.4	-83.4	-92.8
4	oligosaccharide metabolic process	165.3	158.1	-77.3	-104.5
5	cofactor metabolic process	219.0	225.6	-73.7	-76.2
6	generation of precursor metabolites and energy	294.8	293.4	-54.0	-56.1
7	nucleobase-containing small molecule metabolic proc.	404.5	417.4	-49.2	-41.0
8	carbohydrate transport	65.8	77.7	-45.0	-52.8
9	membrane invagination	120.6	118.3	-37.0	-51.7
10	transmembrane transport	644.4	684.7	-24.2	-16.2
11	protein folding	296.9	296.3	-22.7	-26.6
12	lipid metabolic process	484.4	495.7	-19.9	-23.3
13	endocytosis	245.5	248.7	-19.3	-19.3
14	vacuole organization	200.2	199.7	-18.9	-22.4
15	cellular respiration	118.4	125.2	-14.5	-14.1
16	response to starvation	331.4	318.4	-11.2	-15.8
17	protein targeting	478.8	485.1	-10.9	-15.8
18	proteolysis involved in cellular protein catabolic proc.	488.5	494.1	-10.9	-9.8
19	peroxisome organization	124.8	123.5	-6.0	-6.0
20	lipid transport	79.7	90.4	-4.9	-2.8
21	ion transport	380.2	410.7	-4.8	-2.1
22	protein maturation	27.7	30.9	-3.9	-3.0
23	cell morphogenesis	79.4	80.8	-3.6	-3.7
24	sporulation	306.4	301.7	-2.1	-2.5
25	amino acid transport	109.4	113.0	-2.1	-1.6
26	response to osmotic stress	181.8	178.3	-1.6	-2.1
27	protein phosphorylation	587.6	564.3	-1.4	-2.7

Table 2.11: **Induced ESR gene set: comparison between NEAT and LA+S (KEGG pathways)**.

		μ_0		$\log_{10} p$	
	KEGG pathway	NEAT	LA+S	NEAT	LA+S
1	Starch and sucrose metabolism	394.2	400.6	<-300	<-300
2	Butanoate metabolism	84.8	98.0	-202.8	<-300
3	Pentose and glucuronate interconversions	110.7	127.5	-119.9	-185.7
4	beta-Alanine metabolism	104.0	122.9	-118.0	-209.8
5	Glycolysis / Gluconeogenesis	616.3	618.7	-116.5	-149.3
6	Fructose and mannose metabolism	200.0	206.2	-106.7	-160.7
7	Galactose metabolism	173.9	193.2	-104.5	-126.4
8	Glycerolipid metabolism	172.1	193.2	-72.7	-103.2
9	Longevity regulating pathway - multiple species	544.0	508.2	-70.6	-79.1
10	Pentose phosphate pathway	288.0	284.2	-64.0	-105.8
11	Amino sugar and nucleotide sugar metabolism	264.2	277.6	-63.4	-66.7
12	Peroxisome	313.3	332.9	-61.2	-55.8
13	Glutathione metabolism	204.8	221.6	-59.9	-77.8
14	Taurine and hypotaurine metabolism	24.3	28.5	-59.4	-92.8
15	Tyrosine metabolism	163.5	169.9	-51.8	-62.6
16	Tryptophan metabolism	113.3	130.9	-48.2	-59.4
17	Valine, leucine and isoleucine degradation	107.5	124.8	-45.3	-56.8
18	Other glycan degradation	11.7	12.9	-44.2	-66.3
19	Regulation of autophagy	126.7	135.2	-43.3	-45.5
20	Pyruvate metabolism	355.9	388.8	-42.8	-41.6
21	Alanine, aspartate and glutamate metabolism	262.2	284.5	-38.0	-36.7
22	Fatty acid degradation	215.0	225.0	-37.2	-43.7
23	Citrate cycle (TCA cycle)	267.3	299.5	-35.6	-32.9
24	Insulin resistance	172.8	176.5	-30.1	-30.4
25	Histidine metabolism	127.4	147.8	-28.8	-25.8
26	Arachidonic acid metabolism	36.7	44.1	-28.1	-40.6
27	Glyoxylate and dicarboxylate metabolism	201.6	224.8	-27.3	-23.7
28	Sphingolipid metabolism	103.6	116.3	-27.3	-26.2
29	Arginine and proline metabolism	154.3	180.2	-27.0	-24.8
30	Lysine degradation	150.4	160.2	-26.6	-31.5
31	Methane metabolism	254.2	262.7	-26.2	-23.7
32	Phenylalanine metabolism	71.4	81.5	-25.0	-26.4
33	Glycerophospholipid metabolism	270.9	285.1	-24.5	-22.3
34	Protein processing in endoplasmic reticulum	866.0	857.1	-17.4	-20.7
35	Ubiquinone and other terpenoid-quinone biosynth.	41.8	47.1	-13.1	-12.3
36	Propanoate metabolism	107.3	122.9	-12.9	-9.9
37	alpha-Linolenic acid metabolism	27.1	30.5	-11.7	-11.2
38	Fatty acid elongation	75.3	76.1	-10.8	-12.9
39	Glycine, serine and threonine metabolism	264.3	281.1	-6.7	-3.5
40	Nicotinate and nicotinamide metabolism	99.8	111.9	-6.7	-4.7
41	Nitrogen metabolism	52.8	60.7	-5.4	-4.0
42	Thiamine metabolism	32.9	36.8	-4.1	-3.2
43	Selenocompound metabolism	89.3	97.0	-3.2	-1.9
44	Cysteine and methionine metabolism	285.3	310.6	-2.8	-1.0
45	Arginine biosynthesis	134.0	154.2	-2.4	-0.6
46	Sulfur metabolism	105.3	121.9	-2.2	-0.5
47	Biosynthesis of unsaturated fatty acids	103.9	102.1	-2.5	-3.1
48	Regulation of mitophagy - yeast	554.4	510.4	-1.6	-5.1

Chapter 3

A penalized inference approach to stochastic blockmodelling of community structure in the Italian Parliament

3.1 Introduction

The legislative process in modern democracies typically involves three fundamental steps: the proposal of a bill, a discussion on its contents and a final vote on it. Throughout this process, many interactions and collaborations can arise between different political actors, who join their efforts to support, change or oppose a proposed legislation. The analysis of these interactions can, then, provide insight into the features and the mode of operation of different parliaments, and on the way and the extent to which these interactions can influence the legislative process.

Two types of data are often considered in this context. The first is represented by bill cosponsorships networks [Fowler, 2006; Rocca and Sanchez, 2007; Parigi and Sartori, 2014]. A parliamentarian can sponsor a bill individually, or cosponsor it together with other parliamentarians. In the latter case, bill cosponsorship implies a formal collaboration between its proponents, who officially state their agreement and support of the proposed legislation. The second type of legislative data is given by roll-call votes [Kirkland, 2014; Dal Maso et al., 2014], in which parliamentarians express their final decision on a bill.

In this Chapter we study bill cosponsorship in the Italian Chamber of Deputies over the last four legislative cycles, covering the period

2001-2015. We represent bill cosponsorships by means of a undirected graph, where a weighted edge displays the number of bills that two deputies have cosigned together. Compared to other parliaments, such as the American Congress or the German Bundestag, a distinguishing feature in the history of the Italian Parliament is the presence of a large number of political factions. Our aim is to infer a network that summarizes collaborations within and between parties from the network of bill cosponsorships, whose actors are the deputies.

We tackle this issue by viewing edges e_{ij} in the graph as a result of a Poisson process that explicitly depends on group memberships of nodes i and j. The resulting model that we propose builds on the stochastic blockmodels that have been developed for the analysis of unweighted digraphs in social network analysis (see Section 3.1.1 for a review). We resort to generalized linear models and derive measures of group relevance and of attraction or repulsion between groups. Finally, we propose a penalized inference approach for sparse estimation. We show that with the use of penalized likelihood methods, a sparse reduced graph representing collaborations (and repulsions) between political parties can be obtained directly from the signs of the model parameters.

3.1.1 Stochastic blockmodels

Community membership can play an important role in shaping social interactions. Social networks are often featured by the presence of clusters of units that are strongly linked between themselves and weakly connected to individuals that fall outside their cluster, so that ignoring the preferential attachment of units based on community memberships can lead to misleading interpretations of the determinants of network ties. Thus, cluster identification and assessment of the relation between groups of nodes in a network have been active topics of research in the analysis of social networks.

Stochastic blockmodels were first introduced as a modification of the p_1 class of models for unweighted digraphs proposed by Holland and Leinhardt [1981]. Let X_{ij} denote a Bernoulli random variable that takes value 1 if an arrow from node i to node j is present, and is 0 otherwise. The p_1 model assumes that pairs of edges or dyads $Y_{ij} =$

(X_{ij}, X_{ji}) are stochastically independent, and expresses the probability to observe the arrow $X_{ij} = 1$ as a function of four parameters, representing the density of the graph (θ), the tendency of arrows to be reciprocated (ρ), expansiveness (α_i) and popularity (β_j) of nodes i and j. Fienberg and Wasserman [1981] considered a situation in which a partition of units into p groups, also called *blocks*, is available, proposing a more parsimonious representation where α_i and β_j are replaced by p expansiveness group effects α_r, such that $\alpha_i = \alpha_{i'}$ for every i, i' belonging to block B_r, and p popularity group effects β_s.

The definition of *stochastic blockmodel* was proposed by Holland et al. [1983]. According to their definition, a probability distribution for a graph defines a stochastic blockmodel if the random variables X_{ij} are independent, and the random vectors X_{ij} and X_{kl} are identically distributed if nodes i and k are members of the same block B_r, and j and l are in the same block B_s. Stochastic blockmodels imply that nodes within a block are stochastically equivalent, in the sense that if nodes i and k belong to the same block B_r, any probability statement on the graph is left unchanged by interchanging them. Holland et al. [1983] criticized the model proposed by Fienberg and Wasserman [1981] deeming it too restrictive, and advocated that the parameters θ, α_r and β_s should be replaced by block parameters θ_{rs}.

Later on, Wang and Wong [1987] proposed a network model that retains the original formulation of the p_1 model with individual effects α_i and β_j, but also includes a set of parameters ϕ_{rs} associated to each pair of blocks (B_r, B_s).

Anderson et al. [1992] elaborated on the idea of stochastic blockmodels, viewing them as "a mapping of approximately equivalent actors into blocks or positions and a statement regarding the relations between the positions". They considered the p_1 class of models, and they proposed to represent relational ties between blocks of units by means of a reduced graph. They obtained such a graph setting a cutoff c on the predicted probability to observe an arrow from nodes in block B_r to nodes in block B_s, $\hat{\pi}_{rs}$, and drawing an arrow from B_r to B_s if $\hat{\pi}_{rs} > c$.

Hoff et al. [2002] proposed a latent space model that mainly differs from the aforementioned models for the fact that it assumes independence of dyads conditionally on the unobserved position of nodes in

a latent social space, rather than on the observed block-membership. Airoldi et al. [2008] introduced a mixed membership stochastic blockmodel, which can accomodate multiple group membership of units. Finally, stochastic blockmodels have been recently considered as useful tools for graphon inference [Airoldi et al., 2013; Wolfe and Olhede, 2013].

3.2 Bill cosponsorship in the Italian Parliament

The Italian Parliament is based on a bicameral system in which two separate assemblies, the Chamber of Deputies and the Senate, play similar roles in the legislative process. Legislations can be proposed by different actors (including deputies, senators, the government, regions and groups of electors); here, we focus on the legislations proposed by deputies. Each bill can be proposed by a single deputy, or cosponsored by a group of deputies. In the second case, bill cosponsorship defines a symmetric relation between deputies, who formally state their agreement on the content of the proposed legislation by cosponsoring it. Thus, cosponsorship can be taken as a measure of proximity or collaboration between deputies.

Bill cosponsorships can be represented as an undirected network where nodes represent parliamentarians, and the presence of an edge e_{ij} indicates that parliamentarians i and j have cosponsored at least one legislation. We associate to each edge a weight equal to the number of bills that the two parliamentarians have sponsored together in a given time course (typically, one legislative cycle).

In the Italian Chamber, each deputy is required to express their affiliation to one and only one parliamentary group, which typically corresponds to a political party or to a coalition of parties. As a consequence, membership of parliamentary groups generates a partition of deputies into political groups, which we use to assess the patterns of collaboration between political parties.

Data on bill cosponsorship in 27 parliamentary chambers of 20 European countries have been recently collected by Briatte [2016], who has created and published the bill cosponsorship networks aggregated over the span of legislatures. Here we consider the cosponsorship networks for the Italian Chamber of Deputies between the XIV and the

XVII legislature (2001-2015) and we integrate these data with personal details on deputies retrieved from the website of the Chamber of Deputies (*http://dati.camera.it*).

3.3 Poisson process model of bill cosponsorship

A graph is a pair $\mathcal{G} = (V, E)$, which consists of a set of nodes $V = \{1, ..., n\}$ connected by a set of edges $E \subseteq V \times V$. Edges represent relations between nodes, and they can be directed or undirected, as well as weighted or unweighted. In bill cosponsorship networks, each node represents a parliamentarian and a weighted undirected edge between two parliamentarians displays the number of bills that they have cosponsored together. Thus, hereafter we consider the case of an undirected graph, where a discrete weight is associated to each edge. Such a graph can be conveniently represented by means of a symmetric adjacency matrix A, where we set $a_{ij} = 0$ if deputies i and j are not connected, and a_{ij} equal to the number of cosponsorships between deputies i and j otherwise. We assume absence of self-loops, i.e., $a_{ii} = 0$.

We emphasize that alternative representations of bill cosponsorship could be considered as well, as already discussed in Section 1.3. The choice of the representation with an edge-valued graph is motivated by the availability of data aggregated by legislature. This prevents the possibility to consider both a bipartite graph with links connecting deputies to bill, and a graph where a clique is added for each bill subject to cosponsorship. Although we could consider a binary graph in its place, this would imply a loss of information on the frequency of collaborations between the deputies.

3.3.1 Data generating process

We view such a graph as the result of the action of a multivariate Poisson process in a given time course T. Let $N(t)$ be a counting process that denotes the number of events that have occurred until time t. We say that $\{N(t),\ t \in [0, +\infty)\}$ is a univariate Poisson process if $N(0) = 0$, $N(t)$ has independent increments (i.e., $N(t+s)$ - $N(t)$ is independent from $N(t)\ \forall s > 0$) and $N(t)$ follows a Poisson distribution with mean λt.

We can associate a Poisson process $N_{ij}(t)$ with rate λ_{ij} to each pair of deputies (i, j) in the graph. At the beginning of the legislature, i.e. $t = 0$, no cosponsorship as occurred yet, so that $N_{ij}(0) = 0$. If after some time t_1 a first cosponsorship takes place between deputies i and j, we set $N_{ij}(t_1) = 1$. If a second interaction occurs at t_2, we set $N_{ij}(t_2) = 2$, and so on. Thus, $N_{ij}(t)$ denotes the number of bill cosponsorships that have occurred between i and j at a given time point t. If we stop the process at $t = T$, the number of cosponsorships $N_{ij}(T)$ observed until T between each pair (i, j) of deputies is a realization from a Poisson distribution with mean $\mu_{ij} = \lambda_{ij}T$ and it defines a weighted graph, where $a_{ij} = N_{ij}(T)$.

Now, suppose that a partition $\mathcal{P}$ of deputies into p groups or blocks is available, and that block membership determines the rates of each Poisson process, so that we can assume that the interaction rates λ_{ij} are homogeneous within each pair of blocks (B_r, B_s):

$$\lambda_{ij} = \zeta_{rs} \; \forall i \in \text{group } B_r, \forall j \in \text{group } B_s, \text{ with } r, s \in \{1, ..., p\}. \quad (3.1)$$

Under the assumption of independence between the univariate processes, Equation (3.1) defines a stochastic blockmodel, because $N_{ij}(t)$ and $N_{kj}(t)$ are independent, and they are also identically distributed if i and k belong to the same block. Here, the probability that a randomly drawn interaction involves any two deputies in groups B_r and B_s is

$$\pi_{rs} = \frac{n_{rs}\zeta_{rs}}{\sum_{u \leq v=1}^{p} n_{uv}\zeta_{uv}},$$

where $n_{uv} = n_u n_v$ if $u \neq v$, $n_{vv} = n_v(n_v - 1)/2$ and n_v denotes the number of deputies that belong to group B_v.

Our primary interest is to understand which groups are more active in the network, and how members from different groups interact with each other. Thus, we would like to decompose $\mu_{rs} = \zeta_{rs}T$ into a baseline parameter θ_0 that controls the overall bill cosponsorship activity of the network, two effects α_r and α_s that account for the relative importance (productivity or popularity) of political parties r and s, and a further effect ϕ_{rs} that accounts for attraction or repulsion between pairs of parties.

Since a linear relation between μ_{rs} and θ_0, α_r, α_s, ϕ_{rs} is impossible

for the range $\mathbb{R}^+$ of μ_{rs}, we consider a monotone transformation $g : \mathbb{R}^+ \to \mathbb{R}$ of μ_{rs} to be linear in the parameters, i.e.

$$g(\mu_{rs}) = \theta_0 + \alpha_r + \alpha_s + \phi_{rs}. \tag{3.2}$$

This idea is the workhorse of generalized linear models. A convenient choice for g is represented by the logarithm, but others can be considered as well.

Model (3.2) assumes that the cosponsorship behaviour is affected by party membership only, and it may thus be too restrictive [Wang and Wong, 1987]. For example, we can imagine a data generating process where, besides party membership, attributes such as age or gender difference between deputies play a role in the process of bill cosponsorship. If this is the case, a pure stochastic blockmodel would disregard these effects on network formation. In order to cope with such situations, we can consider the following model:

$$\begin{aligned} a_{ij}|(i \in B_r, j \in B_s, x_{ij}) &\sim \text{Poi}(\mu_{ij} = \lambda_{ij}T) \\ g(\mu_{ij}) &= \theta_0 + \alpha_r + \alpha_s + \phi_{rs} + x_{ij}\beta, \end{aligned} \tag{3.3}$$

where x_{ij} is a vector of covariates associated to the couple (i, j) and β is the vector of parameters related to those covariates.

Similar to the model of Wang and Wong [1987], model (3.3) is not a proper stochastic blockmodel, because it allows $\mu_{ij} \neq \mu_{kj}$ for two units $i,\ k$ belonging to the same group B_r. Nevertheless, it retains its focus on the role played by blocks in shaping the network, including specific sets of parameters α_r for block relevance and ϕ_{rs} for interactions within and between blocks. Note that the stochastic blockmodel in (3.2) can be derived as a particular case of (3.3) by setting $\beta = 0$.

3.3.2 Identifiability

Generalized linear models [Nelder and Wedderburn, 1972; McCullagh and Nelder, 1989] relate the mean of the response $\mu \in M$ to a linear combination of variables by means of a link function $g : M \to \mathbb{R}$, which transforms $\mu \in M$ into $\eta = g(\mu) \in \mathbb{R}$.

We model the data generating process in equation (3.2) with

$$\log(\mu_{ij}) = \theta_0 + \sum_{r=1}^{p} \alpha_r D_r(i) + \sum_{r=1}^{p} \alpha_r D_r(j) + \sum_{r \leq s}^{p} \phi_{rs} D_{rs}(i,j), \quad (3.4)$$

where $D_r(i) = I(i \in B_r)$ and $D_{rs}(i,j) = I(i \in B_r, j \in B_s \vee i \in B_s, j \in B_r)$ for $r \leq s = 1, ..., p$ are dummy variables that indicate whether a unit i belongs to group B_r, or whether the pair of nodes (i,j) implies an interaction between blocks B_r and B_s. However, (3.4) is not identifiable without further contraints. Typically the way in which identifiability constraints are specified is not particularly important, as each parametrization is equivalent; however, as we will be penalizing some parameters in later sections, the parametrization will be important. Thus, we introduce the following set of $p+1$ identifiability conditions:

$$\sum_{r=1}^{p} \alpha_r = 0 \text{ and } \sum_{s=1}^{p} \phi_{rs} = 0 \ \ \forall r = 1, ..., p, \quad (3.5)$$

where for ease of notation we write $\phi_{sr} = \phi_{rs}$.

If we incorporate these constraints into (3.4) by letting $\alpha_1 = -\sum_{r=2}^{p} \alpha_r$ and $\phi_{rr} = -\sum_{s \neq r} \phi_{rs}, \ \forall r = 1, ..., p$, (3.4) can be rewritten as

$$\log(\mu_{ij}) = \theta_0 + \sum_{r=2}^{p} \alpha_r T_r(i) + \sum_{r=2}^{p} \alpha_r T_r(j) + \sum_{r<s}^{p} \phi_{rs} T_{rs}(i,j), \quad (3.6)$$

where $T_r(i) = D_r(i) - D_1(i), \ r \neq 1$ and

$$T_{rs}(i,j) = D_{rs}(i,j) - D_{rr}(i,j) - D_{ss}(i,j), \ r \neq s.$$

Likewise, it is possible to represent the data generating process in (3.3) with the following generalized linear model:

$$\begin{aligned} \log(\mu_{ij}) = \theta_0 &+ \sum_{r=2}^{p} \alpha_r T_r(i) + \sum_{r=2}^{p} \alpha_r T_r(j) \\ &+ \sum_{r<s}^{p} \phi_{rs} T_{rs}(i,j) + x_{ij}\beta. \end{aligned} \quad (3.7)$$

3.3.3 Extendibility

The model that we propose differs from traditional models, where the outcome variable refers to a single statistical unit. An edge e_{ij} involves, in fact, two statistical units, i and j. This, in turn, implies that covariates that measure individual features ought to be transformed into edge attributes before they can be included into (3.6). As an example, the sex (F/M) of two nodes gives rise to three possible edges: edges involving two males (MM), two females (FF) or one male and one female individual (FM). The ages of two individuals could be transformed into their absolute difference, or some other transformation such as their average, minimum, maximum, etc.

The unusual nature of this model makes us examine its relevant invariance properties. Wit and McCullagh [2001] introduced the concept of extendibility of a statistical model, arguing that a sensible model is the one that, depending on the particular circumstances, can accommodate further treatments, fewer covariate levels or changes of measurement scale than the ones actually observed. They advocate that invariance under selection of treatments, merging of covariate levels and changes of measurement scale should be explicitly discussed when a new statistical model is introduced, and they showed that some commonly used models fail in this respect.

In our context, one could wonder whether it is sensible to require invariance with respect to group selection (introduction or elimination of a party), group merging (union of two existing parties) or changes of the measurement scale for a_{ij}. The answer to the first two points is strictly connected to what we consider to be a group: in the context of bill cosponsorship networks, each deputy joins a parliamentary group, so that a block is a group of deputies who share similar political views and come together to promote the same political agenda. We therefore would like our model to retain its structure irrespective of the fact that certain groups of individuals have been included or excluded from the analysis. On the other hand, if two parliamentary groups were to be merged this would produce a new political group, whose features would be different from any of the two original groups. For these reasons, we require model (3.6) to be invariant under selection of groups, whereas we do not require invariance under group merging.

Invariance under selection of groups requires that, if one group - say B_p - is excluded from model (3.6) and the new model

$$\log(\mu'_{ij}) = \theta'_0 + \sum_{r=2}^{p-1} \alpha'_r T_r(i) + \sum_{r=2}^{p-1} \alpha'_r T_r(j) + \sum_{r<s}^{p-1} \phi'_{rs} T_{rs}(i,j), \quad (3.8)$$

$$\text{s.t. } \sum_{r=1}^{p-1} \alpha'_r = 0 \text{ and } \sum_{s=1}^{p-1} \phi'_{rs} = 0 \ \ \forall r = 1, ..., p-1,$$

is considered, then it is possible to derive the parameters of (3.8) as a function of the parameters of (3.6). Indeed, this can be achieved by imposing $\mu'_{rs} = \mu_{rs}, \ \ r \leq s = 1, ..., p-1$ (selection requirement), and solving the resulting system of linear equations.

Finally, one might wonder whether it would be sensible to require invariance with respect to changes of measurement scale. Since the edge weights a_{ij} are counts, it does not make sense to apply translations or dilatations to a_{ij}. However, we can consider changes of time scale and ask how this affects the block-means μ_{rs}. Let's consider a change of time scale from a system A with time expressed as T_A and rates as ζ^A_{rs} to a system B with time T_B and rates ζ^B_{rs}. E.g., system A could consider days and system B hours as time unit, so that $T_A = T_B/24$ and $\zeta^A_{rs} = 24\zeta^B_{rs}$. More generally, we can let $\zeta^A_{rs} = k\zeta^B_{rs}, \ k > 0$. Since $T^A = k^{-1}T^B$, the block-means μ_{rs} are not affected by the change of time system:

$$\mu^A_{rs} = T^A \zeta^A_{rs} = k^{-1} T^B k \zeta^B_{rs} = T^B \zeta^B_{rs} = \mu^B_{rs}.$$

This result implies that the parameters θ_0, α_r and ϕ_{rs} in (3.2) are left unchanged, so that the model is invariant with respect to changes of time scale measurement.

3.4 Inference

3.4.1 Parameter estimation

The sufficient statistics associated to model (3.6) consists of the sum of weights a_{ij} and the corresponding number of node pairs involved

for every pair of blocks (B_r, B_s), i.e.,

$$\left(\sum_{i<j, i\in B_r, j\in B_s} a_{ij},\ n_{rs} \right),\ r \leq s \in \{1, ..., p\},$$

where $n_{rs} = n_r n_s$ if $r \neq s$, $n_{rr} = n_r(n_r - 1)/2$ and n_r denotes the number of nodes that belong to group B_r.

As concerns the extended blockmodel in (3.7), denote by

$$\theta = (\theta_0, \alpha_2, ..., \alpha_p, \phi_{12}, \phi_{13}, ..., \phi_{p-1,p}, \beta)$$

the parameter vector of length $q = dim(\theta) = p(p+1)/2 + dim(\beta)$ and let

$$X = (1,\ T_2(i)+T_2(j),\ ...,\ T_p(i)+T_p(j),\ T_{12}(i,j),\ ...,\ T_{p-1,p}(i,j),\ x_{ij})_{i<j}$$

and $y = (a_{ij})_{i<j}$ be the corresponding design matrix and response vector. Hence, the sufficient statistic is given by $X^T y$, as usual in a generalized linear model.

Model estimation can be performed with maximum likelihood. However, since the number of parameters q included in the model increases quadratically with the number of groups p, maximum likelihood estimation could lead to solutions with an extremely large number of parameters, making interpretation cumbersome. Thus, we propose the use of penalized likelihood methods so as to achieve a parsimonious solution.

Besides enhancing model interpretability, penalized likelihood methods enable us to detect potentially sparse blockmodel generating mechanisms: as an example, one could imagine that no preferential attachment or repulsion exists between some pairs of blocks, i.e., that $\phi_{rs} = 0$ for some pairs (B_r, B_s) in (3.3).

Since the introduction of the Lasso [Tibshirani, 1996], penalized inference has become a popular choice for variable selection and the solution of high dimensional problems. Many methods in this field have been introduced (see Bühlmann and van de Geer [2011] and Fan and Li [2001] for an overiew). In this paper we use the adaptive Lasso [Zou, 2006], which is a weighted extension of the Least Absolute Shrinkage and Selection Operator (Lasso) introduced by Tibshirani [1996],

because it has good consistency properties.

The adaptive Lasso aims for a sparse model solution by maximizing a penalized likelihood that incorporates the loglikelihood of the model, and a weighted ℓ_1 penalty on the parameters included in the model. This penalty is multiplied by a tuning parameter $\delta \geq 0$, which determines the amount of regularization that is imposed on the parameters. The adaptive Lasso problem for (3.7) is

$$\max_{\theta} \ \log L(\theta) - \delta \sum_{j=1}^{q} w_j |\theta_j|, \tag{3.9}$$

where $L(\theta)$ denotes the likelihood of the model and w_j is the weight associated to the jth element θ_j of θ. The tuning parameter δ is typically chosen either by cross-validation, or by minimizing a suitably defined information criterion. We discuss this issue in more detail in Section 3.4.2.

Denote by θ^* a consistent estimator of θ and by $N = n(n-1)/2$ the total number of pairs of nodes in the network. The attractive feature of the adaptive Lasso is that if the weight vector is defined as $w = 1/|\theta^*|^\gamma$, and if $\delta/\sqrt{N} \to 0$ and $\delta N^{(\gamma-1)/2} \to \infty$, then the adaptive lasso estimator $\hat{\theta}$ is consistent in variable selection (see theorem 4 in Zou, 2006).

The choice of the parameters that are subject to the ℓ_1 penalty mostly depends on the role and the meaning that we associate to them. In our view, the parameter ϕ_{rs} expresses the presence of a preferential attachment or repulsion between units in groups B_r and B_s after we have accounted both for the overall density of the network (θ_0), and the relevance of the groups (α_r and α_s). In order to retain this interpretation, we do not penalize θ_0 nor $\alpha_r,\ r = 1, ..., p$, i.e., we set $w_j = 0$ if $j \in \{1, ..., p\}$.

On the other hand, we would like to achieve some sparsity in the representation of relations betweeen groups by penalizing the ϕ_{rs} coefficients ($r \neq s$), as well as β. For the penalty weights, we compute the maximum likelihood estimate $\hat{\theta}$ and set $w_j = 1/|\hat{\theta}_j|^\gamma$, with $\gamma = 2$, for $j > p$.

3.4.2 Model selection

In a penalized likelihood framework, the tuning parameter δ determines the amount of regularization that it is imposed on the parameters and, eventually, the level of sparsity of the solution. Two main approaches are typically employed for the selection of an optimal tuning parameter δ^*: cross-validation, or minimization of model information criteria. In the latter case, one seeks for

$$\delta^* = \underset{\delta}{\operatorname{argmin}} \left(-2 \log L_\delta(\hat{\theta}) + a_m \cdot h_\delta \right), \tag{3.10}$$

where m denotes the number of observations and h_δ the dimensionality of the model. Different choices have been proposed for a_m. Alongside Akaike's information criterion (AIC), which sets $a_m = 2$, and the Bayesian information criterion (BIC), which takes $a_m = \log m$, recent proposals include the generalized information criterion ("GIC" hereafter) of Fan and Tang [2013], where $a_m = \log(\log m) \log h_\delta$, and the modified BIC ("MBIC" hereafter) of Chand [2012], where $a_m = \sqrt{m}/h_\delta$.

Here, we consider five simulations to assess the performance of these criteria in the selection of δ. In each simulation, we generate a sequence of networks with increasing number of nodes $n = 50, 100, 150, \ldots, 500$, following the blockmodel defined by (3.2). We set $\theta_0 = 0.7$ and draw $\alpha_r \in U(-0.3, 0.3), r > 1$. Moreover, we set some $\phi_{rs},\ r \neq s$ coefficients equal to 0, and draw the remaining ones in such a way that $|\phi_{rs}| \sim U(c_{\min}, c_{\max})$, with $c_{\max} = 0.5$. Coefficients α_1 and $\phi_{rr},\ r = 1, \ldots, p$ are subsequently derived from Equation (3.5). The simulations differ for the number h of null ϕ_{rs} coefficients ($r \neq s$) and for the betamin condition ($|\phi_{rs}| \geq c_{\min}$) imposed on the non-null ϕ_{rs} coefficients; Table 3.1 summarizes the different settings in each simulation.

We perform model selection over a grid of 100 δ values. Each selection criterion leads to an optimal δ and corresponding model estimates. In order to compare the performance of each criterion in the selection of models capable to correctly distinguish signals ($\phi_{rs} \neq 0$) and non-signals ($\phi_{rs} = 0$), we compute the accuracy of each solution, i.e.

$$\text{Accuracy} = \frac{\text{True positives + True negatives}}{p(p-1)/2},$$

Table 3.1: **An overview of Simulations A-D.** In Simulation A, we consider a dense model (i.e., with high dimensionality h) with a moderate betamin condition imposed on the non-null ϕ_{rs} coefficients ($|\phi_{rs}| \geq c_{\min}$). We progressively increase the sparsity of the model in Simulations B and C. In Simulation D we consider a model with medium sparsity level (like the one in Simulation B), but we make signal detection harder by imposing a milder betamin condition.

Simulation	# ($\phi_{rs} = 0$)	h	Betamin condition
A	10	45 (dense)	$c_{\min} = 0.2$ (moderate)
B	20	35 (medium)	$c_{\min} = 0.2$ (moderate)
C	30	25 (sparse)	$c_{\min} = 0.2$ (moderate)
D	20	35 (medium)	$c_{\min} = 0.1$ (mild)

and we compare it to the maximum achievable accuracy for the set of 100 models considered. As shown in Figure 3.1, every criterion quickly achieves the maximum accuracy when a dense model is considered (Simulation A), but the accuracy of cross-validation, AIC and MBIC is often lower when sparser models are considered (Simulations B and C), or when signal detection is complicated by the imposition of a milder betamin condition (Simulation D). Overall, BIC and GIC outperform the competing methods and, thus, appear to be the best information criteria in terms of selection accuracy.

3.4.3 Reduced graph

A focal aspect of stochastic blockmodels is the description of the relations between blocks of individuals. Anderson et al. [1992] proposed to represent relational ties between blocks of units by means of a reduced graph, whose nodes are the blocks. The idea behind this reduced graph is rather simple: summarize the original graph by visualizing relations between blocks directly, so as to achieve a simpler and clearer representation.

As an example, consider the graph in the left box of Figure 3.2. Three groups of nodes (sets 1, 4 and 5) appear to be featured by a strong internal connectivity; besides, nodes within each group tend to be preferentially linked to nodes belonging to one or two other groups; e.g., it appears that nodes in set 3 tend to prefer nodes in sets 1 and 2 to nodes in sets 4 and 5. Based on similar observations, we can attempt to draw a graph that summarizes our intuition: the graph in the right

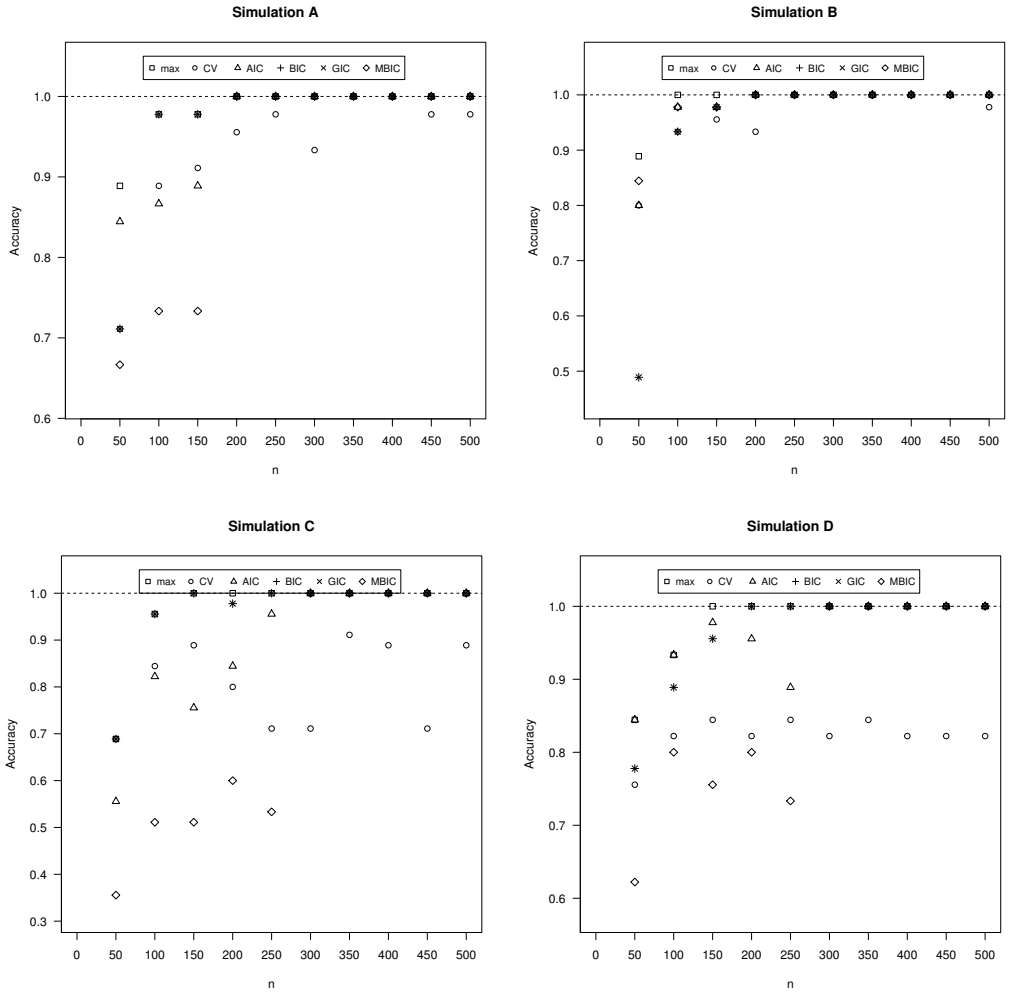


Figure 3.1: **Results of Simulations A-D.** Comparison of the accuracy of models chosen by 10-fold cross-validation (CV), Akaike's Information Criterion (AIC), Bayesian Information Criterion (BIC), the Generalized Information Criterion (GIC) of Fan and Tang [2013] and the modified BIC (MBIC) of Chand [2012] with the maximum achievable accuracy (MAX). Every criterion quickly achieves the maximum accuracy in Simulation B, where we consider a model with few null ϕ_{rs}. In Simulations A, C and D, instead, BIC and GIC outperform CV, AIC and MBIC: this is particularly apparent when a sparser model is considered (Simulation C), or when signal detection is made harder by the imposition of a milder betamin condition (Simulation D).

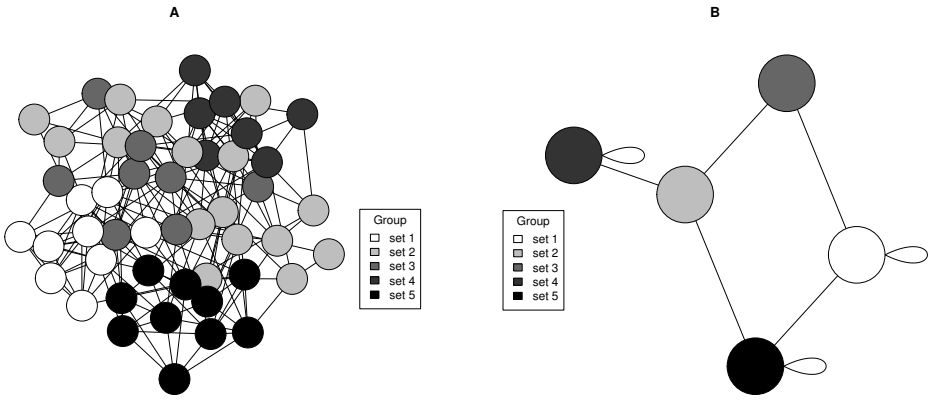


Figure 3.2: An unweighted graph with 50 nodes, partitioned into 5 groups (A) and a simplified representation of relations between groups (B).

box of Figure 3.2 provides an example.

Different strategies to derive a reduced graph from a statistical model can be considered. Anderson et al. [1992] obtained such a graph setting a cutoff c on the predicted probability to observe an arrow from nodes in a group B_r to nodes in a group B_s, $\hat{\pi}_{rs}$, and drawing an arrow from B_r to B_s if $\hat{\pi}_{rs} > c$. The resulting reduced graph links blocks that are highly connected, but edges therein do not necessarily display preferential attachments between groups. For example, nodes in a group B_r could have overall higher degrees: if this is the case, block B_r would be connected to any block, just as a result of the high average degree of nodes in B_r.

Instead, we propose an alternative strategy to derive the reduced graph, which is based on the parameter estimates $\hat{\phi}_{rs}$ in models (3.6) and (3.7) rather than on $\hat{\mu}_{rs}$ (or $\hat{\pi}_{rs}$). By doing so, we control for the average degree of blocks B_r and B_s, because an estimate $\hat{\phi}_{rs} > 0$ entails preferential attachment between nodes in blocks B_r and B_s. Thus, we draw an edge between two blocks B_r and B_s if $\hat{\phi}_{rs} > 0$. We can also derive a reduced graph that displays preferential repulsions by connecting blocks such that $\hat{\phi}_{rs} < 0$.

3.5 Analysis of bill cosponsorship networks of the Italian Chamber of Deputies

We consider now the networks representing bill cosponsorship in the Italian Chamber of Deputies, which we introduced in Section 3.2. We focus our attention on the cosponsorship networks of the four legislative cycles XIV-XVII, covering the period 2001-2015.

During this period, the number of parliamentary groups has ranged from 8 (XIV and XVI legislative cycles) to 10 (XVII) and 13 (XV legislative cycle); in each legislative cycle, a mixed group has always been present, gathering deputies from small political groups with different political orientation, which did not meet the requirements (defined in the Chamber's regulations) for the creation of a parliamentary group.

We study the dependency between bill cosponsorship and parliamentary groups, controlling for individual features such as gender, age and the electoral constituency in which the deputy has been elected. Gender can give rise to edges involving two male (MM), two female (FF) and a female and a male (FM) deputies; we take MM as reference. Besides, we consider the age difference of the two deputies, and an indicator function indicating whether the two deputies have been elected in the same electoral constituency.

Then, for each legislative cycle we estimate (3.7) with the adaptive LASSO, using BIC to select the tuning parameter δ. Table 3.2 shows the estimates of θ_0 and β (standard errors are not shown because their computation and usefullness is still controversial in penalized inference settings). Note that the intercept θ_0 is lower for the XV and XVII cycles. This is coherent with the fact that whereas legislatures XIV and XVI lasted 5 years, the XV legislative cycle lasted 2 years only, and that the data for the current (XVII) legislature refer to a period of less than 3 years (until the end of 2015)[I]. Furthermore, bill cosponsorships turn out to be more frequent between female deputies (FF) and, in general, they are more likely to take place if at least one of the deputies involved is female (FM). The effect of age difference on bill cosponsorship is small and negligible, whereas the positive coefficient associated to pairs of deputies elected in the same electoral constituency provides evidence that deputies tend to collaborate also

[I] Assuming a fixed rate ζ across legislatures, we would expect $\mu = T\zeta$ and θ_0 to increase with T.

Table 3.2: **Size effects of gender, age and electoral constituency on bill cosponsorship.** The table displays the estimates of θ_0 (unpenalized) and β (penalized) in model 3.7 for the following legislative cycles: XIV (2001-2006), XV (2006-2008), XVI (2008-2013) and XVII (2013-2015).

Covariate	Legislative cycle			
	XIV	XV	XVI	XVII
Intercept (θ_0)	-2.49	-3.05	-2.53	-3.60
Female-Male (FM)	0.251	0.170	0.174	0.198
Female-Female (FF)	0.998	1.00	0.662	0.606
Age difference	0	0	-0.010	-0.002
Same electoral constituency	0.522	0.490	0.514	0.553

on the basis of geographic proximity.

Whereas the effect of covariates in Table 3.2 appears to be qualitatively the same over time, the pattern of collaboration between parties changes substantially. The reduced graphs in Figure 3.3 display preferential attachments between parliamentary groups; a self loop indicates that there is a tendency of deputies to cosign with deputies from the same parliamentary group, and node size is proportional to the relative frequency of cosponsorship (α_r) of deputies in each group. A first, interesting conclusion is that cosponsorships during the XIV and XV legislative cycles reflects collaborations within each party, and between parties that belonged to the same political coalition. In fact, both legislatures featured strong competition between two coalitions, one of which (the right-wing in the first case, and the left-wing in the latter) held the majority in Parliament and could thus govern on its own. This situation seems to have generated a strong ideological polarization, which is evident from the pattern of collaborations between the parliamentary groups.

The division of the Chamber into two coalitions ended with the XVI legislature, as a centrist party (UDC) that was not part of any coalition entered the Chamber. The majority was in the hand of the right-wing coalition, whereas UDC and the left-wing coalition were at the opposition. Three years later, a group of right-wing deputies formed FLI, a new political group that abandoned the right-wing coalition and entered a centrist coalition with UDC. One year later, the right-wing government resigned and a coalition government, supported by a heterogeneous coalition of parties, took its place. Besides cospon-

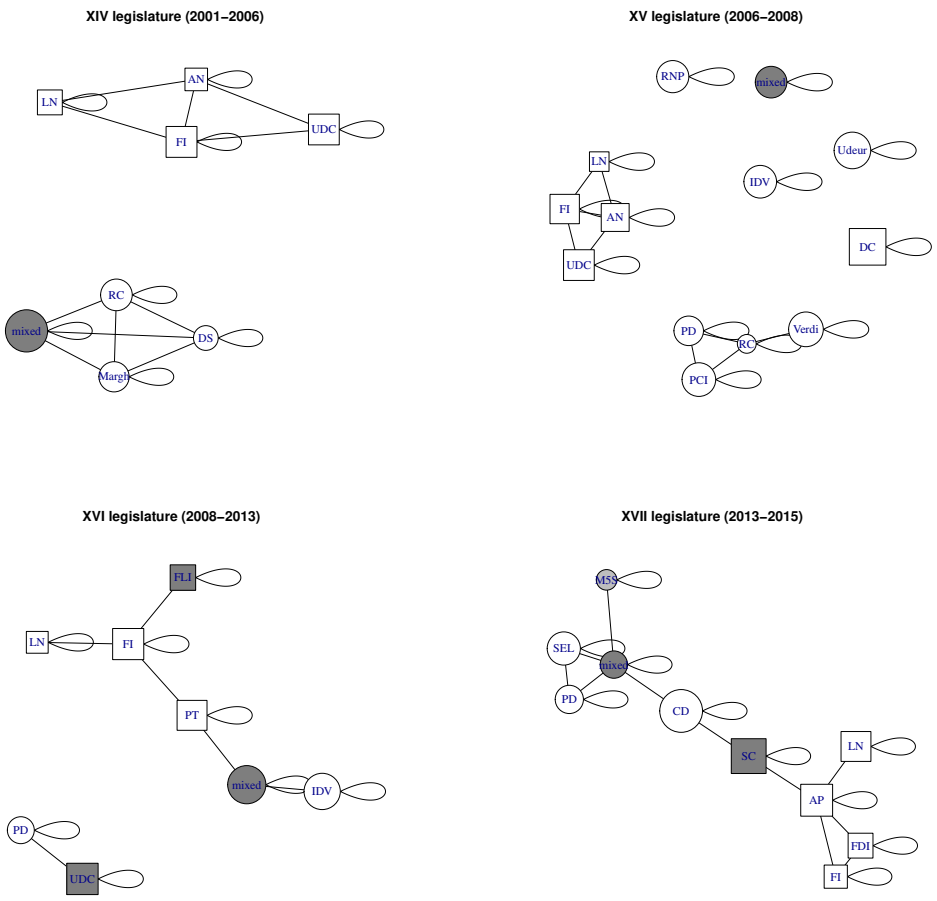


Figure 3.3: **Reduced graphs representing collaborations between parliamentary groups based on bill cosponsorship.** The graphs display preferential attachments based on model 3.7 (i.e., $\hat{\phi}_{rs} > 0$). White squares denote right-wing parliamentary groups, white circles left-wing groups and darkgrey squares centrist groups. A darkgrey circle denotes the mixed group, whereas a lightgrey circle the Movimento 5 Stelle. Node size is proportional to the productivity of each parliamentary group ($\hat{\alpha}_r$).

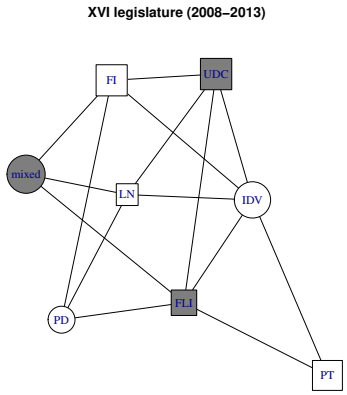


Figure 3.4: **Reduced graphs representing preferential repulsions in the XVI legislative cycle.** The graph displays preferential repulsions based on model 3.7 (i.e., $\hat{\phi}_{rs} < 0$). White squares denote right-wing parliamentary groups, white circles left-wing groups and darkgrey squares centrist groups. A darkgrey circle denotes the mixed group, whereas a lightgrey circle the Movimento 5 Stelle. Node size is proportional to the productivity of each parliamentary group ($\hat{\alpha}_r$).

sorships within parliamentary groups, our model detects preferential attachments between the main right-wing party (PDL) and each of the smaller parties from the same coalition (including FLI), between two opposition parties (PD and UDC) and a couple of further preferential attachments involving the mixed group. It is also interesting to consider the reduced graph displaying preferential repulsions ($\hat{\phi}_{rs} < 0$) shown in Figure 3.4: most of the edges indicate (not surprisingly) that there is few collaboration between parties in different coalitions, but also between UDC and FLI, which allied towards the end of the legislative cycle. In short, the pattern of bill cosponsorships reflects the division between the right-wing majority (FLI, LN, PDL and PT) and the opposition (PD, IDV, UDC) of the first half of the legislature quite clearly, despite the fact that the analysis considers cosponsorship over the whole legislature span. A possible explanation for this result is that cosponsorship events are more likely to take place in the first years of each legislature: as a matter of fact, owing to the long time that is typically necessary for a bill of parliamentary initiative to be discussed and approved, a bill proposed towards the end of the legislature is extremely unlikely to be approved, and this can in turn discourage deputies from proposing bills in the last years of their mandate.

The fragmentation in the composition of the Chamber has become even stronger in the current (XVII) legislative cycle. Since none of the 4 coalitions now represented in the Parliament (left-wing, right-wing, the centrist Scelta Civica (SC) and the Movimento 5 Stelle (M5S)) could form a government alone, alliances between parties belonging to different coalitions have arisen, giving rise to heterogeneous parliamentary majorities. In this case, the reduced graph in Figure 3.3 shows that besides self-loops accounting for a tendency towards within-group cosponsorship, deputies from different right-wing parties collaborate with each other. Moreover, deputies from the centrist party SC collaborate with deputies belonging to two centrist parties (CD and AP) which are ideologically alike and are all part of the majority, but belong to different political coalitions (left and right-wing, respectively). Further collaborations are detected between two left-wing parties (PD and SEL) and between the mixed group and various parties. Apart from a preferential attachment with the mixed group, deputies from M5S do not seem to collaborate with any other party.

In short, our analysis of bill cosponsorship networks indicates the evolution from a highly polarized political arena, in which deputies based collaborations on their identification with left or right-wing values, towards an increasingly fragmented Parliament, where a rigid separation of political groups into coalitions does not seem to hold any more, and collaborations beyond the perimeter of coalitions have now become now possible.

3.6 Conclusion and discussion

Community affiliation can deeply affect social behaviour and the formation of relations between individuals. In social network analysis, stochastic blockmodels represent a popular approach to assess community structure in the presence of known community memberships.

In this Chapter, we have developed an extended stochastic blockmodel for the analysis of bill cosponsorships in the Italian Parliament. This model retains the focus on relations between pairs of blocks that characterizes pure stochastic blockmodels by including parameters for group productivity (α_r) and interactions between pairs of groups (ϕ_{rs}), but it also allows heterogeneity of units within a block. Because the

number of parameters increases quadratically with the number of groups, we advocate the use of a penalized estimation approach so as to select a parsimonious model that displays relevant preferential attachments and repulsions between pairs of blocks only. We represent these preferential relations by means of a reduced graph, which summarizes the relations that exist between blocks.

Our analysis of bill cosponsorship in the Italian Chamber of Deputies from 2001 to 2015 points out the evolution from a political system strongly polarized into a left and a right-wing coalition, in which bill cosponsorship takes place almost exclusively between deputies belonging to the same coalition, towards an increasingly fragmented political arena, with more than two coalitions of parties and in which collaborations beyond the perimeter of coalitions are now possible.

We remark that our data analysis relies on bill cosponsorship networks that are aggregated over the span of each legislature. This does not allow us to take into account possible changes in membership of parliamentary groups within a legislature, a practice - known as *trasformismo* - that is rather frequent in the Italian Parliament. For this reason, we have relied on the group memberships of each deputy as reported by the website *http://dati.camera.it*. In principle, our model is capable to handle this situation. If, for example, deputy i has been member of party B_q for a time span equal to t_1 and of party B_r for t_2, the number of bills that they have cosponsored with deputy $j \in B_s$ is still a Poisson process:

$$N_{ij}(t_1 + t_2) = N_{ij}(t_1) + N_{ij}(t_2) \sim Poi(\lambda_{qs} t_1 + \lambda_{rs} t_2).$$

Thus, availability of data disaggregated over time would allow us to cope with these changes in group membership, providing a more realistic account of this phenomenon. Furthermore, this would also entitle us to model directly the interaction rates λ_{ij} between deputies, which (as we pointed out in our comment to the results for the XVI legislature) is unlikely to be constant across the legislature (both because of procedural issues, and of the changing political environment). In particular, it would make it possible to verify the hypothesis that most cosponsorships take place at the beginning of the legislature.

Even though here we have considered networks where edges are undirected and weighted, with weights in the set of natural numbers,

the models that we propose can be easily generalized in two directions.

Directed edges can be handled by introducing a new set of parameters so as to distinguish sender and receiver nodes, as well as a parameter ρ that indicates the tendency of arrows to be reciprocated. As an example, we can rewrite (3.2) as follows

$$\begin{aligned} a_{ij}|(i \in B_r, j \in B_s) &\sim \text{Poi}(\mu_{rs}), \\ \log(\mu_{rs}) &= \theta_0 + \rho + \alpha_r + \beta_s + \phi_{rs}. \end{aligned}$$

Here, a new set of parameters β_s ought to be introduced: whereas α_r is now a measure of productivity of group B_r (which the sender node i belongs to), β_s is a measure of popularity of group B_s (which the receiver node j belongs to). Furthermore, note that here $\phi_{rs} \neq \phi_{sr}$, and that a positive ϕ_{rs} denotes now a preferential attachment from nodes in group B_r towards nodes in group B_s.

Furthermore, the use of generalized linear models allows to extend easily model (3.3) beyond Poisson processes. E.g., if the network is unweighted (i.e., $a_{ij} \in \{0, 1\}$) it suffices to replace the Poisson with a Bernoulli distribution, and the log-link with a logit or a probit link function; if a weighted network with weights in the set of real numbers is at hand, the Poisson distribution can be replaced with any continuous distribution, and the identity function becomes a natural choice for g.

Note that the models that we have considered here fit inside the p_1 class of models, where independence of edges (undirected graphs) or dyads (digraphs) is assumed. In terms of the network generating process, this implies that the univariate Poisson processes $N_{ij}(t)$, that are responsible of the final value $a_{ij} = N_{ij}(t)$, are assumed to be independent of each other. An extension of the model could go in the direction of allowing dependence between these Poisson processes, relaxing this independence assumption. A further extension, which we will pursue in Chapter 4, is the inclusion of nodal random effects in the model, so as to model possible unobserved sources of nodal heterogeneity.

Bibliography

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014.

Airoldi, E. M., Costa, T. B., and Chan, S. H. (2013). Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. *arXiv preprint arXiv:1311.1731*.

Anderson, C. J., Wasserman, S., and Faust, K. (1992). Building stochastic blockmodels. *Social Networks*, 14(1):137–161.

Briatte, F. (2016). Network patterns of legislative collaboration in twenty parliaments. *Network Science*, 4(2):266–271.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer.

Chand, S. (2012). On tuning parameter selection of lasso-type methods - A Monte Carlo study. In *IBCAST 2012 Conference Proceedings*, pages 120–129.

Dal Maso, C., Pompa, G., Puliga, M., Riotta, G., and Chessa, A. (2014). Voting behavior, coalitions and government strength through a complex network analysis. *PLOS ONE*, 9(12):e116046.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, Y. and Tang, C. Y. (2013). Tuning parameter selection in high dimensional penalized likelihood. *Journal of the Royal Statistical Society. Series B*, 75(3):531–552.

Fienberg, S. E. and Wasserman, S. (1981). Categorical data analysis of single sociometric relations. *Sociological Methodology*, 12:156–192.

Fowler, J. H. (2006). Connecting the Congress: A study of cosponsorship networks. *Political Analysis*, 14(4):456–487.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137.

Holland, P. W. and Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50.

Kirkland, J. H. (2014). Ideological heterogeneity and legislative polarization in the United States. *Political Research Quarterly*, 67(3):533–546.

McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models*, volume 37. CRC press.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A*, 135(3):370–384.

Parigi, P. and Sartori, L. (2014). The political party as a network of cleavages: disclosing the inner structure of Italian political parties in the seventies. *Social Networks*, 36:54–65.

Rocca, M. S. and Sanchez, G. R. (2007). The effect of race and ethnicity on bill sponsorship and cosponsorship in Congress. *American Politics Research*, 36(1):130–152.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288.

Wang, Y. J. and Wong, G. Y. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19.

White, H. C., Boorman, S. A., and Breiger, R. L. (1976). Social structure from multiple networks. i. blockmodels of roles and positions. *American Journal of Sociology*, 81(4):730–780.

Wit, E. and McCullagh, P. (2001). The extendibility of statistical models. *Contemporary Mathematics*, 287:327–340.

Wolfe, P. J. and Olhede, S. C. (2013). Nonparametric graphon estimation. *arXiv preprint arXiv:1309.5936*.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Chapter 4

Joint modelling of community structure and nodal heterogeneity in networks

4.1 Introduction

In Chapter 3 we have discussed how stochastic blockmodels can provide a useful insight of the relations between communities of nodes in a network. We have also observed that traditional stochastic blockmodels suffer from a strong limitation, which is the assumption that nodes in a group have a homogeneous behaviour. Therefore, we have proposed an extension of stochastic blockmodels that allows to model heterogeneity between nodes within the same block on the basis of a set of observed covariates, and we have advocated the use of a penalized inference approach to estimate that model.

Social networks typically feature a strong heterogeneity among their actors, which is apparent from the fact that their degree distribution is usually strongly skewed. In friendship networks, it is common to observe that a few individuals are highly popular, whereas most individuals in the network have a smaller number of friends. In bill cosponsorship networks, often a few, highly collaborative parliamentarians tend to cosponsor a large number of bills, whereas their colleagues usually cosponsor just a few, selected legislations. In order to be of practical utility in the analysis of real networks, it is therefore important that stochastic blockmodels can handle this characteristic feature of social networks consistently.

Stochastic blockmodels, however, are based on information on group

membership of nodes only. A first source of information which allows to model directly this heterogeneity is given by any other individual covariate besides group membership. As discussed in Section 3.4.1, the inclusion of covariates in stochastic blockmodels sensitively increases the computational complexity of the model at hand, but estimation of the extended blockmodel with covariate information proposed in Chapter 3 can still be performed on a standard computer with limited temporary memory (RAM) using the R package `glmnet` [Friedman et al., 2010].

There are two reasons, however, that suggest that considering further sources of heterogeneity could prove valuable. The foremost is that in some cases, only information on group membership might be available, without any further nodal covariates. But even if a limited number of covariates is available, such as in the case of the bill cosponsorship networks for the Italian Parliament that are the subject of Chapter 3, considering the possibility that there might be further, unobserved sources of heterogeneity allows for a better model fit.

The inclusion of random effects to model unobserved sources of heterogeneity is the subject of this Chapter. In Section 4.2 we will discuss how it is possible to extend the model considered in Chapter 3 so as to include nodal random effects. We will propose a generalized linear mixed model that allows to achieve such purpose and consider two alternative inference approaches: a traditional one, based on maximum likelihood estimation, and a penalized inference one, in which (in analogy to Chapter 3) we resort to the adaptive Lasso. Besides comparing the results from those two approaches, we will discuss the considerable increase in computational complexity that the inclusion of random effects in a penalized inference framework currently implies - a computational burden that, for the time being, seems to prevent (or, at least, to limit to very small networks) the possibility to carry out the estimation of an extended stochastic blockmodel with random effects in a penalized inference setting using a personal computer.

In Section 4.3 we will consider an alternative approach to modelling unobserved sources of heterogeneity in networks based on latent space models for networks [Hoff et al., 2002; Handcock et al., 2007; Krivitsky et al., 2009]. Latent space models differ in nature from the stochastic blockmodels discussed in Chapter 3 and Section 4.2, mainly

because they do not incorporate information on known group membership of units. Nevertheless, group membership can be used after model estimation to inspect the position of nodes in the latent space, and so the latent space model can provide an interesting alternative to inspect the presence of community structure while accounting for unobserved sources of heterogeneity across nodes.

4.2 Joint modelling of community structure and nodal heterogeneity

4.2.1 Background: from GLMs to GLMMs

A well known feature of generalized linear models [McCullagh and Nelder, 1989] is the fact that they relate the linear predictor $\eta = X\beta$ to the expectation of the response $\mu = E(Y)$ by means of a (monotone continuous and differentiable) link function g: $\eta = g(\mu)$. It is important to observe that differently from the linear model, generalized linear models (GLMs) do not include an error component in the model.

The inclusion of a random component in GLMs can be achieved with an extension of GLMs, known as generalized linear mixed models (GLMMs, McCulloch et al. 2008). GLMMs relate the conditional expectation of the response Y given the unobserved random component U, $\mu = E(Y|U = u)$, to the sum of the linear predictor $X\beta$ and of Zu:

$$g(\mu) = X\beta + Zu,$$

where $u \sim f_U(u)$ and Z is a known model matrix associated to the random effects.

Different approaches to the estimation of GLMMs have been proposed. In principle, one would like to maximize the log-likelihood function

$$\ell = \sum_i \log f(y_i) = \sum_i \log \int_u f_{Y_i|U}(y_i|u) f_U(u) du. \tag{4.1}$$

However, optimization of (4.1) is often complicated by the integration of the random component. Therefore, a host of alternative strategies for its maximization have been proposed, including penalized quasi-likelihood, restricted maximum likelihood and the EM algorithm; we

refer to Dean and Nielsen [2007] and McCulloch et al. [2008] for an overview.

An alternative approach to the estimation of GLMMs was proposed by Lee and Nelder [1996]. It consists of the maximization of the hierarchical likelihood h, which is the joint log-likelihood of y and u (or, equivalently, the sum of the conditional log-likelihood of $y|u$ and of the log-density function of u):

$$h = \sum_i \log f(y_i, u) = \sum_i \{\log f(y_i|u) + \log f(u)\}. \tag{4.2}$$

4.2.2 Model specification

In Section 3.3.1 we have introduced the stochastic blockmodel for edge-valued graphs defined by Equation (3.2). Such a model assumes exchangeability between those node pairs (i, j) which involve the same pair of groups (B_r, B_s), so that $\mu_{ij} = \mu_{i'j'}$ for $i, i' \in B_r$ and $j, j' \in B_s$. In order to make the model more flexible, we have allowed the expected number of cosponsorships between two deputies to depend on a set of covariates x_{ij} (Equation (3.3)). Such a model allows for heterogeneity of nodes within the same block, which is modelled on the basis of observed information at the level of nodes or of edges.

Further flexibility can be achieved by considering potential unobserved sources of heterogeneity. Let u_i be a normally distributed random effect that refers to node i, such that $u_i \sim N\left(0, \sigma^2\right)$, and let $u_i \perp\!\!\!\perp u_j$ if $i \neq j$. Like before, we can specify a Poisson GLM with the logarithm as link function g; besides, we can include nodal random effects $u_i,\ i \in \{1, ..., n\}$, which we associate to each node. A GLMM that extends models (3.2) and (3.3) can thus be defined by the following data generating process

$$\begin{aligned} a_{ij}|(i \in B_r, j \in B_s, x_{ij}, u_i, u_j) &\sim \text{Poi}(\mu_{ij}) \\ g(\mu_{ij}) &= \theta_0 + \alpha_r + \alpha_s + \phi_{rs} + x_{ij}\beta + u_i + u_j \end{aligned} \tag{4.3}$$

and, after imposing the identifiability constraints in Equation (3.5), it

can be specified (cfr. Equations (3.6) and (3.7)) as

$$\begin{aligned}\log(\mu_{ij}) = \theta_0 + \sum_{r=2}^{p} \alpha_r \left[T_r(i) + T_r(j)\right] + \sum_{r<s}^{p} \phi_{rs} T_{rs}(i,j) \\ + \sum_{k=1}^{n} u_k \left[I_k(i) + I_k(j)\right],\end{aligned} \tag{4.4}$$

where $I_k(i) = 1$ if and only if $k = i$, and $I_k(i) = 0$ otherwise.

4.2.3 Model estimation

We consider two alternative approaches to the estimation of model (4.4).

The first one is based on the maximization of the hierarchical likelihood of the model [Lee and Nelder, 1996]. We resort to the R package `hglm` [Ronnegard et al., 2010] for model fitting.

The second approach that we consider is a penalized likelihood approach, in analogy to what we have done in Chapter 3. For the same reasons outlined in Section 3.4, we employ the adaptive Lasso [Zou, 2006] to estimate the model. Because of the random component that has now been included in the model, the optimization problem takes form

$$\max_{(\theta,\sigma)} \ \log L(\theta, \sigma; y, u) - \delta \sum_{j=1}^{q} w_j |\theta_j|, \tag{4.5}$$

where $L(\theta, \sigma; y, u)$ denotes the likelihood of the model, $u = (u_1, ..., u_n)$ is the vector of random effects and θ and w_j are the same as in Equation (3.9). The weights w_j are defined like in Chapter 3 as well: we do not penalize θ_0 nor $\alpha_r,\ r = 1, ..., p$, whereas we set $w = 1/|\theta^*|^\gamma$ for β and ϕ, choosing the maximum likelihood estimator as consistent estimator θ^* of θ and $\gamma = 2$.

Optimization of problem (4.5) can be performed with the R package `glmmLasso` [Groll, 2016], which implements the estimation algorithm presented in Groll and Tutz [2014].

4.2.4 Results

In this Section we discuss the results of the application of model (4.4) to data on the bill cosponsorship network for the XVI legislature of the Italian Chamber, which we already analyzed in Chapter 3.

Maximum likelihood

We begin by estimating the unpenalized generalized linear mixed model with the R package `hglm` [Ronnegard et al., 2010], which maximizes the hierarchical likelihood (4.2) associated to model (4.4).

Table 4.1 shows the results for the covariates and the random effects variance. Comparison with the results from the penalized GLM from Chapter 3 (Table 3.2) shows that the same conclusions can be derived with respect to the facts that female deputies are more active in bill cosponsorship than their male colleagues, and that age difference is substantially irrelevant whereas geographical proximity accounts for many collaborations between the deputies. The variance of the random effects seems relatively small but not negligible.

In Figure 4.1 we display a reduced graph which differs from the ones described in Sections 3.4.3 and 3.5. As we are now considering unpenalized ϕ_{rs} coefficients, it is not possible to distinguish relevant collaborations from weak or irrelevant collaborations by shrinking some ϕ_{rs} to zero. However, we can compare each ϕ_{rs} to its standard error and distinguish positive coefficients that can be thought to be significantly different from zero (solid edges) from positive coefficients that are not significant (dashed edges). It is remarkable to observe that all the edges in the reduced graph in Figure 3.3 are also present, either as solid or dashed edges, in the reduced graph which we derive here. Moreover, the reduced graph for the unpenalized model features some further edges - a result that is however not surprising, as it is easy to imagine that some of those coefficients that were shrunk to zero in the penalized GLM might turn out to be positive (although marginally or not significant) in the unpenalized GLMM, where no penalty is imposed on them.

Overall, inclusion of random effects in the model does not seem to alter substantially the conclusions which we derived in Chapter 3 for the bill cosponsorship network of the XVI legislature.

Table 4.1: **Estimates of covariate effects and random effects variance for the unpenalized GLMM.** The table displays the estimates of θ_0, β and σ^2 for the bill cosponsorship network of the XVI legislature (2008-2013). * denotes estimates that are significant at 5% level and ** estimates significant at 1% level.

Parameter	Estimate
Intercept (θ_0)	-3.298**
Female-Male (FM)	0.258**
Female-Female (FF)	0.800*
Age difference	-0.008
Same electoral constituency	0.562**
σ^2	0.217

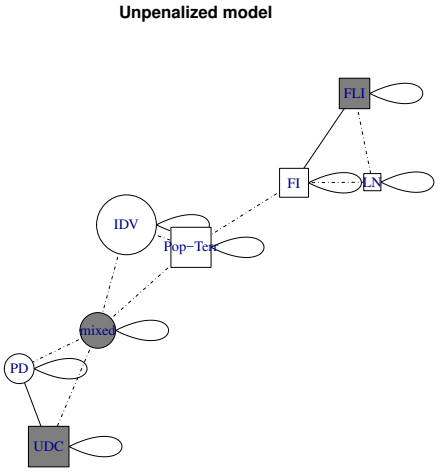


Figure 4.1: **Reduced graph representing collaborations between parliamentary groups based on bill cosponsorship (unpenalized GLMM).** The graph displays preferential attachments based on model 3.7 (i.e., $\hat{\phi}_{rs} > 0$). Solid edges correspond to those parameter estimates that are significant at 5% level, and dashed edges to those that are not. Node size is proportional to the productivity of each parliamentary group ($\hat{\alpha}_r$).

Penalized likelihood

Estimation of the penalized GLMM described in Section 4.2.3, which features both random effects and an ℓ_1 penalty imposed on some of the fixed effects, turns out to be rather challenging.

Fitting model (4.4) with the adaptive Lasso currently represents an extremely expensive computational task, which cannot be performed on a personal computer: as its implementation, based on version 1.4.4 of the R package `glmmLasso`, required an amount of temporary memory approximately equal to 425 GB, we had to resort to some computers with high memory that are part of *Peregrine*, the High Performance Computing cluster of the University of Groningen[I]. This has allowed us to overcome the large memory requirement for the optimization (nevertheless, we emphasize the fact that computers with such an availability of temporary memory are currently rare).

While implementing the computations, we also come across some numerical issues that seem to indicate the possibility of convergence to local maxima of the penalized likelihood of the model which we consider, rather than to the global maximum. In particular, the maximized likelihood of the model is not a monotone function of the tuning parameter, although we would expect the maximized likelihood to increase (or, at least, not to decrease) whenever a smaller delta value is considered. Therefore, we remark that the results reported hereafter should be considered with care.

As concerns the covariates, the conclusions are in accordance with the results from the penalized GLM and the unpenalized GLMM for age difference and electoral constituency. Instead, they appear to be different for FM and FF, for which the estimates of the corresponding fixed effects seem to indicate the irrelevance of sex for the productivity of deputies in bill cosponsorship. However, note that the average of random effects for male deputies is equal to $\bar{z}_M = -0.04$, whereas it is $\bar{z}_F = 0.163$ for female deputies. Consideration of this substantial difference in the random effects allows to conclude once more that female deputies cosponsor bills more frequently than their male colleagues do.

With respect to the inferred collaborations between parties, the

[I]Information on Peregrine can be obtained consulting the following URL: https://redmine.hpc.rug.nl/redmine/projects/peregrine/wiki.

Table 4.2: **Estimates of covariate effects and random effects variance for the penalized GLMM.** The table displays the estimates of θ_0, β and σ^2 for the bill cosponsorship network of the XVI legislature (2008-2013).

Parameter	Estimate
Intercept (θ_0)	-2.38
Female-Male (FM)	-0.004
Female-Female (FF)	-0.043
Age difference	-0.014
Same electoral constituency	0.699
σ^2	0.684

only positive estimates obtained with this approach are related to collaborations within the same party. No positive coefficients are instead detected with respect to collaborations between parties. A possible explanation for this result is that most of the cosponsorships that take place between different parties might be due to the most productive deputies, whose large positive random effects might account for most of these cosponsorships between parties.

4.3 Latent space models

4.3.1 Modelling networks with latent space models

Latent space models for social networks assume that each individual i in a network has an unknown position z_i in a d-dimensional latent social space, and that edges are conditionally independent given the position of individuals in the latent space.

Latent space models were introduced by Hoff et al. [2002], who considered the case of a binary graph. For an undirected graph with $Y = (Y_{12}, ..., Y_{n-1,n})$, where $Y_{ij} \in \{0, 1\}$ denotes presence or absence of an edge between nodes i and j and x_{ij} is a vector of covariates related to the pair (i, j), they assume that each tie is conditionally independent from the other ones given its positions z_i, z_j in the latent space

$$P(Y|Z, X, \theta) = \prod_{i<j=2}^{n} P(Y_{ij}|z_i, z_j, x_{ij}, \theta) \tag{4.6}$$

and they model the conditional probability of $Y_{ij} = 1$ with the follow-

ing logistic model:

$$\text{logit}P(Y_{ij}|z_i, z_j, x_{ij}, \theta) = \beta_0 + x_{ij}\beta - |z_i - z_j|, \tag{4.7}$$

where $z_i \sim \text{MVN}_d(0, \sigma_Z^2 I_d)$ and $|\cdot|$ is the Euclidean distance (but it could be any other suitable distance).

Handcock et al. [2007] introduced an extension of this model that accounts for the presence of community structure in networks, by allowing the nodes to belong to G different clusters and by letting the latent positions follow a mixture of G multivariate normal distributions

$$z_i \sim \sum_{g=1}^{G} \lambda_g \text{MVN}_d(\mu_g, \sigma_g^2 I_d), \tag{4.8}$$

where λ_g denotes the weight of component g in the mixture, $\lambda_g \geq 0$ $\forall g$ and $\sum_{g=1}^{G} \lambda_g = 1$.

Krivitsky et al. [2009] extended the latter model by introducing a set of sociality effects γ_i, which allow to account for degree heterogeneity across nodes. Likewise the models Hoff et al. [2002] and Handcock et al. [2007], their model is discussed for the case of a binary graph, but it can be generalized to weighted networks as well. For an undirected graph with edge weights represented as $Y = (Y_{12}, ..., Y_{n-1,n}) \in \mathbb{R}^{n(n-1)/2}$, their model takes form

$$P(Y|Z, X, \theta) = \prod_{i<j=2}^{n} f(Y_{ij}|z_i, z_j, x_{ij}, \theta), \tag{4.9}$$

where

$$\eta_{ij} = g^{-1}\left[E(Y_{ij}|z_i, z_j, x_{ij}, \theta)\right] = \beta_0 + x_{ij}\beta - |z_i - z_j| + \gamma_i + \gamma_j, \tag{4.10}$$

and $\gamma_i \sim N(0, \sigma_\gamma^2)$, $i = 1, ..., n$ denote the (independent) sociality effects.

4.3.2 Estimation of latent space models

Estimation of latent space models is typically performed in a Bayesian framework. Hoff et al. [2002] proposed to estimate the model defined by Equations (4.6) and (4.7) following a Bayesian approach that requires the specification of prior distributions both for β_0, β and for

σ^2, and an approximate computation of the posterior distribution by means of Markov Chain Monte Carlo (MCMC) sampling. Handcock et al. [2007], instead, consider two alternative approaches: a two-stage maximum likelihood estimation which relies on the Expectation-Maximization (EM) algorithm, and a Bayesian estimation approach which also relies on MCMC. Finally, Krivitsky et al. [2009] proposed to estimate the model in (4.9) and (4.10) in a Bayesian setting, by making use of MCMC.

The latent space models proposed in Hoff et al. [2002]; Handcock et al. [2007]; Krivitsky et al. [2009] can be estimated with the `latentnet` R package [Krivitsky and Handcock, 2015].

4.3.3 Application to bill cosponsorship networks

Hereinafter, we analyse the bill cosponsorship networks introduced in Chapter 3 by considering a latent space model for undirected graphs defined by Equations (4.9) and (4.10), where we take $\eta_{ij} = \log\left[E(Y_{ij}|z_i, z_j, x_{ij}, \theta)\right]$ and $z_i \sim \text{MVN}_d(0, \sigma_Z^2 I_d)$. Since such a model does not exploit information on memberships of deputies to parliamentary groups, we will assess the presence of a community structure with respect to party membership by observing the position of deputies from different parties in the latent space.

The results of the application of the latent space model are reported in Table 4.3 and Figures 4.2 and 4.3.

Similarly to the models presented in Chapter 3, also here the effect of covariates (Table 4.3) is relatively stable over time. Once more, we find strong evidence that cosponsorships are more frequent between female deputies, but based on this model it is instead questionable whether female-male (FM) cosponsorships are more frequent than male-male (MM) ones. Also the irrelevance of age difference and the tendency to collaborate more with deputies elected in the same electoral constituency are coherent with the results from Chapter 3. The variance of the nodal random effects is close to 1 over the whole period.

With respect to the collaborations between parties, the representation of deputies in the latent social space (Figures 4.2 and 4.3) can be compared with the reduced graphs obtained from the extended

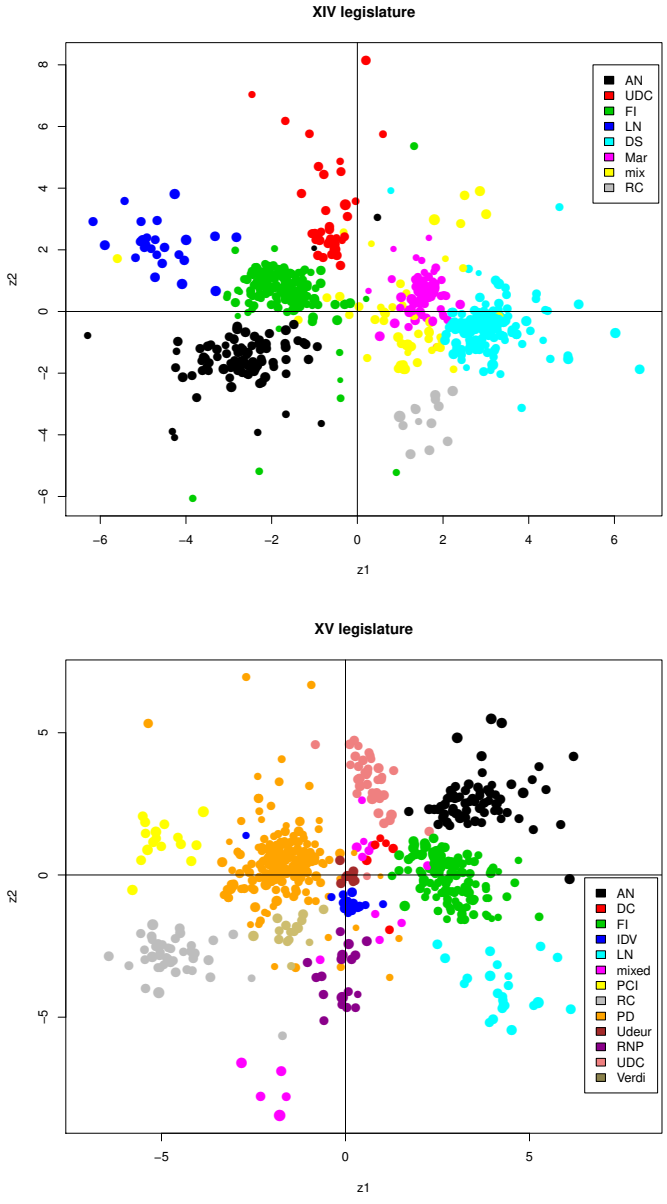


Figure 4.2: **Estimates of the latent position of deputies in legislatures XIV and XV.** Colors denote affiliation to political parties, node sizes are proportional to the estimates of the sociality effects $\hat{\gamma}_i$.

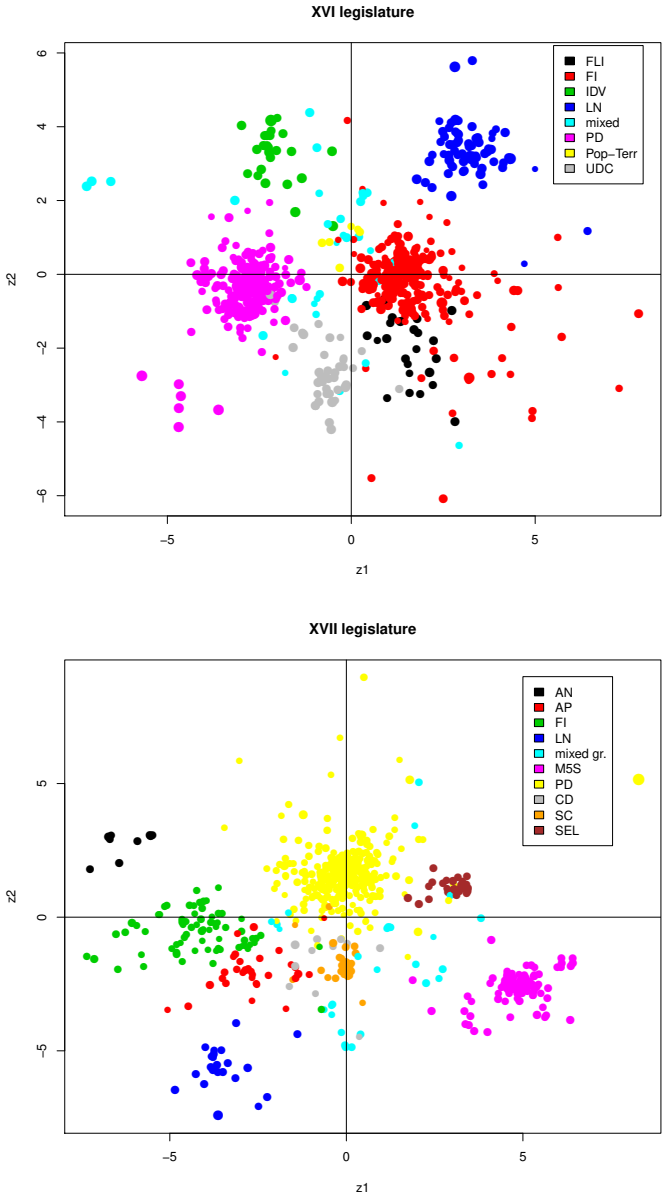


Figure 4.3: **Estimates of the latent position of deputies in legislatures XVI and XVII.** Colors denote affiliation to political parties, node sizes are proportional to the estimates of the sociality effects $\hat{\gamma}_i$.

Table 4.3: **Size effects of gender, age and electoral constituency on bill cosignature.** The table displays the estimates of θ_0, β and of the random effects variance σ^2_γ obtained from the latent space model for the following legislative cycles: XIV (2001-2006), XV (2006-2008), XVI (2008-2013) and XVII (2013-2015). * denotes estimates that are significant at 5% level and ** estimates significant at 1% level.

	Legislative cycle			
Covariates	XIV	XV	XVI	XVII
Intercept (θ_0)	-0.009	-0.151**	-0.067*	-0.143**
Female-Male (FM)	-0.059**	-0.058**	0.034**	-0.010**
Female-Female (FF)	0.426**	0.421**	0.259**	0.195**
Age difference	-0.005**	-0.010	-0.008	0.018
Same electoral constituency	0.529**	0.495**	0.544**	0.587**
σ^2_γ	0.831	1.008	1.063	1.013

stochastic blockmodel (Figure 3.3).

A first conclusion that can be drawn from the analysis of Figures 4.2 and 4.3 is that deputies tend to form clusters according to their party membership. This supports the evidence on strong within-party collaborations already reported in Chapter 3.

For the first two (XIV and XV) legislatures, the polarization between left-wing and right-wing parties is apparent also from the positions of deputies in the latent space (Figure 4.2). In particular, for the XIV legislature almost all deputies from the right-wing coalition have $z_1 < 0$, and those from the left-wing $z_1 > 0$. For the XV legislature, most left-wing deputies have $z_1 < 0$, whereas $z_1 > 0$ for right-wing deputies. Interestingly, members from those parties (DC, IDV, Udeur, RNP and mixed group) that according to the reduced graph in Figure 3.3 do not seem to collaborate with other parties, have z_1 values close to 0.

The reduced graph for the XVI legislature reported three groups of collaborations:

- between the main right-wing party (FI) and the other right-wing groups (LN, FLI, PT). Here, almost all members of those parties are on the right in the top plot of Figure 4.3.
- collaborations between the mixed group and IDV and PT. Indeed, deputies from those parties mostly occupy the second quarter of the plot (but note the fact that the mixed group seems to consist of a few separate subgroups);

- a collaboration between PD and UDC: deputies from those parties are positioned in and close to the third quarter.

Finally, also the results for the XVII legislature seem to match the ones from the extended stochastic blockmodel: members from the two main left-wing parties (PD and SEL) have $z_2 > 0$, whereas members from right-wing parties (FI, LN, AN and AP) belong mostly to the third quarter. The collaborations detected between SC and AP and SC and AP also seem to be corroborated by the proximity of deputies from those parties in the latent space. Moreover, members from the Movimento 5 Stelle (M5S) tend to lie isolated from the other deputies also here (fourth quarter).

4.4 Discussion

In this Chapter, we have discussed two alternative strategies to jointly model community structure and nodal heterogeneity in networks. Accounting for both properties is important, as they are concurrently present in many social networks.

We have begun by proposing an extension to the model which we proposed in Chapter 3. Such an extension is based on the inclusion of a set of nodal random effects into the model, so as to account for possible unobserved sources of degree heterogeneity.

We have considered two alternative approaches to the estimation of such model and compared the results with the ones on bill cosponsorship networks in Chapter 3. First, we have shown that estimation of an unpenalized GLMM yields to conclusions that are substantially coherent with the ones obtained from the penalized blockmodel of Chapter 3.

Then, in analogy with the penalized inference approach which we undertook in Chapter 3, we have introduced a penalty on some of the fixed effects in model (4.4). We have discussed the computational issues that arise in the fitting of the resulting penalized GLMM, as well as the fact that this currently limits the potential applicability of such an approach, which so far cannot be carried out on a (standard) personal computer.

We have also considered an alternative class of models, latent space models, which are different in nature from stochastic blockmodels.

The main assumption behind latent space models is that each individual in a network has an unknown position in a d-dimensional latent social space, and that edges are conditionally independent given the position of individuals in the latent space. Although - differently from stochastic blockmodels - latent space models do not directly incorporate information on known group membership of units, this information can be used to evaluate the presence of community structures in the latent space. Application of such a strategy to the bill cosponsorship networks of the Italian parliament has pointed out clearly the presence of community structures induced by party membership, yielding results that are to a good extent similar to the ones obtained from the proposed extended stochastic blockmodels.

Bibliography

Dean, C. and Nielsen, J. D. (2007). Generalized linear mixed models: a review and some extensions. *Lifetime Data Analysis*, 13(4):497–512.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Groll, A. (2016). *glmmLasso: Variable Selection for Generalized Linear Mixed Models by L1-Penalized Estimation.* R package version 1.4.4.

Groll, A. and Tutz, G. (2014). Variable selection for generalized linear mixed models by l 1-penalized estimation. *Statistics and Computing*, 24(2):137–154.

Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society. Series A*, 170(2):301–354.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.

Krivitsky, P. N. and Handcock, M. S. (2015). *latentnet: Latent Position and Cluster Models for Statistical Networks.* The Statnet Project (http://www.statnet.org). R package version 2.7.1.

Krivitsky, P. N., Handcock, M. S., Raftery, A. E., and Hoff, P. D. (2009). Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31(3):204–213.

Lee, Y. and Nelder, J. A. (1996). Hierarchical generalized linear models. *Journal of the Royal Statistical Society. Series B*, pages 619–678.

McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models*, volume 37. CRC press.

McCulloch, C. E., Searle, S. R., and Neuhaus, J. M. (2008). *Generalized, Linear, and Mixed Models*. Wiley.

Ronnegard, L., Shen, X., and Alam, M. (2010). hglm: A package for fitting hierarchical generalized linear models. *The R Journal*, 2(2):20–28.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Chapter 5

Clustering graphs using mixtures of generalized linear models

5.1 Introduction

The last decades have witnessed a growing interest in the analysis of relational data. Typically, these data come in the form of a network [Newman, 2010] specifying a list of relations between individuals or objects, which is then represented by means of a graph.

Networks have been devised and studied in many fields, including sociology [Moreno, 1934; Wasserman and Faust, 1994], biology [Barabasi and Oltvai, 2004; Signorelli et al., 2016], medicine [Klovdahl, 1985] and engineering [Guimera et al., 2005]. Until a few years ago, the generation and collection of network data represented a challenging task that limited the practical applicability of network science to a single network of modest size. Recent technological advances such as the development of sensor-based measurements, next generation sequencing techniques and functional magnetic resonance imaging, as well as the advent of social media, have widely simplified the collection of relational data, fostering the analysis of larger network datasets.

Statistical modelling of networks has been carried out focusing on different network features, such as degree distribution, community structure or network statistics. Different types of models have been proposed, including the p_1 and p_2 models [Holland and Leinhardt, 1981; van Duijn et al., 2004], exponential random graphs [Frank and Strauss, 1986], stochastic blockmodels [Holland et al., 1983; Airoldi et al., 2008] and latent space models [Hoff et al., 2002].

The increasing availability of network data has also encouraged the collection of several instances of the same network. One example is given by longitudinal sequences of networks, where each network in the sequence represents a snapshot of the network at a given time point, the sequence thus representing the evolution of a system over time. Cross-sectional sequences of networks have been considered as well: in this case, each network can be associated to a different statistical unit and one might want to assess the extent of similarities and differences between units therein by comparing their networks.

Most of the research in this field has focused on the dynamic evolution of a network. Snijders [2001] proposed a stochastic actor-oriented model where the decision to create or dissolve an edge is based only on the current state of the network, and not on its previous states. Hanneke et al. [2010] introduced a dynamic extension of ERGMs, known as Temporal Exponential Random Graph Model (TERGM). An extension of the Latent Space Models for dynamic networks has been proposed by Sewell and Chen [2015]. Matias and Miele [2017], instead, developed a dynamic stochastic blockmodel, that allows group membership of units to vary over time.

Statistical modelling of cross-sectional sequences of networks, often referred to as populations of networks, is more recent. Durante et al. [2016a] proposed a non-parametric bayesian approach to characterize the distribution of the population of networks, rather than that of each network instance, and Durante et al. [2016b] applied this approach to the comparison of networks representing cosubscription of services in different agencies of an insurance company.

The availability of network sequences poses new challenges to statisticians. Clearly, modelling each network separately does not appear an effective strategy: irrespective of whether the sequence is temporal or cross-sectional, we expect networks therein to be similar to a certain degree, so that modelling the networks jointly would allow to borrow information among them. Besides, by jointly modelling the network sequence one can achieve a much more parsimonious answer than by repeating separate analyses of each network in the sequence. In particular, it seems reasonable to specify a joint statistical model capable to quantify similarities and differences between graphs.

In this chapter we propose a strategy to cluster networks, which re-

lies on mixtures of generalized linear models. Mixtures of generalized linear models [Grün and Leisch, 2008] combine mixture models, which have been used to perform model based clustering since long, and generalized linear models, which can be exploited to estimate some popular network models (such as, for example, the p_1 and p_2 models and stochastic blockmodels). We begin by introducting mixtures of generalized linear models and showing how they can be applied so as to cluster graphs in Section 5.2. In Section 5.3 we provide an implementation of the EM algorithm that allows to carry out model estimation, and we assess its performance with simulations. In order to improve the performance of the EM algorithm, in Section 5.4 we propose an extension of the EM based on Simulated Annealing (EMSAGC), and we show that this allows to improve the accuracy of clustering in cases where the EM algorithm alone performs poorly. An example application is provided in Section 5.5, where we consider daily interaction networks between employees of the French Institute for Public Health Surveillance.

5.2 Model specification

5.2.1 Mixtures of generalized linear models

Mixture models have been widely employed to cluster units with a model based clustering approach, as well as for density estimation. A finite mixture model postulates that an observation y is derived from a mixture of M probability density functions $f(y|\theta_m),\ m \in \{1, ..., M\}$, which we call "components" of the mixture:

$$y \sim f(y|\Theta) = \sum_{m=1}^{M} \pi_m f(y|\theta_m), \tag{5.1}$$

where π_m denotes the prior probability that y belongs to component $f(y|\theta_m)$ with parameter θ_m, and $\Theta = (\theta_1, ..., \theta_M)$. Clearly, $\pi_m \geq 0\ \forall m \in \{1, ..., M\}$ and $\sum_{m=1}^{M} \pi_m = 1$.

A generalized linear model [McCullagh and Nelder, 1989], on the other hand, assumes that the density of y belongs to an exponential

family, i.e.,

$$y \sim f(y|\theta, \phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi) + c(y, \theta)}\right\} \tag{5.2}$$

for suitable choices of a, b and c, and that the conditional expectation of Y given a vector of covariates x is related to the linear predictor $x\beta$ by a link function g:

$$\eta = g[E(Y|x)] = x\beta.$$

Although they provide two different ways to characterize the distribution of y, mixtures of probability density functions and generalized linear models can be combined by defining mixtures of generalized linear models [Grün and Leisch, 2008]. This can be achieved by assuming that an observation y is derived from a mixture of M densities from an exponential family, and that the mean μ_m of each density can be related to the linear predictor by a link function g:

$$\begin{aligned} y \sim f(y|\Theta) = \sum_{m=1}^{M} \pi_m f(y|\theta_m, \phi_m) = \sum_{m=1}^{M} \pi_m \exp\left\{\frac{y\theta_m - b(\theta_m)}{a(\phi_m) + c(y, \theta_m)}\right\}, \\ \mu_m = g^{-1}(x\beta). \end{aligned} \tag{5.3}$$

5.2.2 Clustering networks with mixtures of generalized linear models

We consider a sequence of K undirected graphs $\mathcal{S} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_K\}$, where each graph $\mathcal{G}_k = (V, E_k),\ k \in \{1, ..., K\}$, defines a specific set of edges E_k between the same set of v vertices V. Each graph $\mathcal{G}_k$ can be represented by its adjacency matrix Y_k, and we represent the sequence $\mathcal{S}$ with an array $\mathbf{Y}$ of dimension $v \times v \times K$, where each horizontal slice Y_k is the adjacency matrix of graph $\mathcal{G}_k$. Therefore, an entry y_{ij}^k in $\mathbf{Y}$ refers to the presence (and intensity) or absence of edge (i, j) in the k-th graph $\mathcal{G}_k$.

In principle, we could imagine that each graph $\mathcal{G}_k$ with adjacency matrix Y_k is drawn from a different distribution $f(Y|\theta_k),\ k \in \{1, ..., K\}$

with parameter vector θ_k:

$$Y_k \sim f\left(Y|\theta_k\right).$$

In the presence of many networks, however, this would result in a cumbersome modelling exercise, yielding K different models obtained from separate analyses of each graph.

In order to avoid that, it seems sensible to consider the existence of clusters of graphs with similar $f\left(Y|\theta_k\right)$: if any such cluster exists, we would like to borrow information among graphs within that cluster, so as to estimate a joint model within the cluster rather than many separate graph models. As a result, we assume that the graph sequence $\mathcal{S}$ consists of $M \leq K$ subpopulations of graphs $\mathcal{S}_1, ..., \mathcal{S}_M$, each with probability density function $f\left(Y|\theta_m\right)$, $m \in \{1, ..., M\}$. We denote by $Z_k \in \{1, ..., M\}$ the identifying label of graph $\mathcal{G}_k$, such that $Z_k = m$ if $\mathcal{G}_k \in \mathcal{S}_m$. Since it is unknown which graph belongs to which subpopulation, the identifying labels $Z = (Z_1, ..., Z_K)$ are latent. Therefore, we view each graph in the sequence as a random draw from a mixture model whose components are the densities $f\left(Y|\theta_m\right)$

$$Y_k \sim \sum_{m=1}^{M} \pi_m f\left(Y|\theta_m\right), \tag{5.4}$$

with mixing proportions $\pi_m = Pr(Z_k = m),\ m \in \{1, ..., M\}$ denoting the prior probabilities that a graph belongs to the mth subpopulation $\mathcal{S}_m$.

If we let $\Theta = (\theta_1, ..., \theta_M)$, the likelihood of the graph sequence $\mathcal{S}$ with adjacency array $\mathbf{Y}$ is thus

$$\begin{aligned} L(\mathbf{Y}, Z|\Theta) = Pr(\mathbf{Y}, Z|\Theta) &= \prod_{k=1}^{K} Pr(Y_k|Z_k, \Theta) Pr(Z_k|\Theta) \\ &= \prod_{k=1}^{K} \pi_{Z_k} f\left(Y_k|\theta_{Z_k}\right). \end{aligned} \tag{5.5}$$

As we have pointed out in Chapters 3 and 4, often the densities $f\left(Y|\theta_m\right)$ in Equations (5.4) and (5.5) can be conveniently characterized by recurring to generalized linear models. This can be done by considering densities f from exponential families, and modelling the

conditional expectation of each edge y_{ij}^k as

$$\eta_{ij}^k = g\left[E\left(y_{ij}^k|x,\theta_m\right)\right] = x_{ij}\beta. \tag{5.6}$$

Clearly, it is assumed that each edge y_{ij}^k in graph $\mathcal{G}_k$ is drawn from the same (unknown) subpopulation $\mathcal{S}_m$; thus, the density of graph $\mathcal{G}_k$ can be obtained as

$$f\left(Y_k|\theta_{Z_k}\right) = \prod_{i<j} f\left(y_{ij}^k|\theta_{Z_k}\right). \tag{5.7}$$

The use of generalized linear models allows to consider different network models. For example, if one is interested in clustering graphs according to their degree distribution, they could consider a p_1 model by letting $\eta_{ij}^k = \alpha_i^m + \alpha_j^m$. If a partition of nodes into groups is known, such as in the case of bill cosponsorship networks, a stochastic block-model could be specified as well. More generally, if one would simply like to cluster graphs without assuming a specific network model, they can specify a model with one parameter for each pair of nodes:

$$\mu_{ij}^k = g^{-1}\left(\eta_{ij}^k\right) = \gamma_{ij}^m. \tag{5.8}$$

5.3 Model estimation with the EM algorithm

5.3.1 Implementation of the EM algorithm

The EM algorithm [Dempster et al., 1977] represents a popular choice for the estimation of mixture models. The algorithm allows to maximize a likelihood $L(y,z|\theta)$ in the presence of missing or latent data z, and it consists of successive iterations of two steps, respectively called expectation (E) step and maximization (M) step. The expectation step requires the computation of the conditional expectation of the likelihood $L(y,z|\theta)$ given the current estimate of θ and the observed data y, whereas the maximization step updates the parameter estimates by maximizing the expected likelihood determined in the E step.

The first algorithm that we consider for the maximization of the likelihood in Equation (5.5) is given by the following implementation of the EM algorithm:

- for $k \in \{1,...,K\}$ and $m \in \{1,...,M\}$, define the initial probabilities $p_{km}^1 = Pr(Z_k = m)$. Denote by P^1 the $K \times M$ matrix

which collects these probabilities;

- for $t = 1, 2, ...$ and until convergence is reached:
  - ☐ **M step**. Given P^t, estimate M network models (specified as GLMs) with weights given by $(p^t_{1m}, ..., p^t_{Km})$ for the m-th component, and obtain $\hat{\Theta}^t$.
  - ☐ **E step**. Given $\tilde{\Theta}^t$, derive P^{t+1} as

$$p^{t+1}_{km} = \frac{Pr(\mathcal{G}_k|\hat{\theta}^t_m)}{\sum_{j=1}^{M} Pr(\mathcal{G}_k|\hat{\theta}^t_j)}. \tag{5.9}$$

5.3.2 Simulations

We assess the performance of the EM algorithm discussed in Section 5.3.1 by considering both binary and edge-valued networks. We consider sequences of K undirected graphs with v nodes, where $K \in \{10, 20, 50\}$ and $v \in \{10, 20, 50, 100\}$, which are generated from a mixture model consisting of two subpopulations $\mathcal{S}_1$ and $\mathcal{S}_2$, with mixing proportions $\pi_1 = 0.6$ and $\pi_2 = 0.4$. We do not assume a specific network model, but use instead the "full" model specified by Equation (5.8), to wit, we associate one parameter to each pair of nodes in every subpopulation. Once the sequence $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2\}$ is generated, we estimate a mixture of GLMs with the EM algorithm of Section 5.3.1. We consider 10 different starting points, each obtained by drawing $p^1_{k1} \in U(0.48, 0.52),\ k \in \{1, ..., K\}$ and deriving $p^1_{k2} = 1 - p^1_{k1}$ accordingly.

We assess the performance of the algorithm by considering two indicators. The first is the average accuracy of solutions obtained from the different starting points, which we denote as $\bar{A}$. The second is given by the number of cases in which the likelihood of the solution is non inferior to the likelihood of the true solution: we take this indicator as a measure of the performance of the optimization procedure and we denote it by OP (Optimization Performance).

As concerns sequences of binary graphs, we draw each γ^m_{ij} (the probability of having an edge between nodes i and j in graphs from subpopulation m) from a uniform distribution ranging from 0.1 to 0.9. We then employ a mixture of binomial GLMs with logistic link function to estimate the model.

Table 5.1: **Average accuracy ($\bar{A}$) over 10 different starting points using the EM algorithm.** As a measure of accuracy of the EM algorithm, we average the accuracy of the solutions obtained from 10 different starting points.

K	v	Bernoulli	Poisson	Negative Binomial
10	10	72 %	100%	67%
10	20	81%	100%	70%
10	50	68%	100%	65%
10	100	65 %	100%	66%
20	10	91.5 %	100%	80.5%
20	20	89 %	100%	91%
20	50	85 %	100%	90.5%
20	100	81 %	100%	83.5%
50	10	100 %	100%	96.4%
50	20	100 %	100%	100%
50	50	100 %	100%	100%
50	100	100 %	100%	100%

With respect to edge-valued graphs, we first consider sequences of networks where each γ_{ij}^m, $m \in \{1, 2\}$ is drawn from a Poisson distribution with mean uniformly ranging in $[0.1, 10]$. We estimate the model using a Poisson GLM with the logarithm as a link function. In order to assess the performance of the algorithm with respect to model mispecifications, we also consider network sequences generated from a negative binomial distribution with dispersion parameter $\phi = 1$, so as to account for scenarios where the degree distribution is strongly overdispersed (but we still employ a Poisson GLM to estimate the clusters).

The results of the simulation are reported in Tables 5.1 and 5.2. The EM algorithm performs very well when either a large number of graphs ($K = 50$) is available, or in the case of edge-valued graphs with Poisson distribution. Both the average accuracy and optimization performance of the algorithm are instead poorer for network sequences with a smaller number of graphs ($K = 10$ or $K = 20$), if the graphs are binary or edge-valued with overdispersed degree distribution.

Table 5.2: **Optimization performance (OP) of the EM algorithm.** OP is an indicator of the performance of the optimization. It is the number of cases in which the likelihood of the solution is non inferior to the likelihood of the true solution. As we consider 10 different starting points, $0 \leq OP \leq 10$.

K	v	Bernoulli	Poisson	Negative Binomial
10	10	3	10	2
10	20	2	10	1
10	50	1	10	1
10	100	1	10	1
20	10	3	10	1
20	20	3	10	8
20	50	4	10	8
20	100	4	10	6
50	10	0	10	10
50	20	10	10	10
50	50	10	10	10
50	100	10	10	10

5.4 An extension of the EM algorithm based on Simulated Annealing (EMSAGC)

5.4.1 Implementation of the EMSAGC algorithm

In order to improve the performance of the EM algorithm presented in Section 5.3.1, we propose a modified version of that algorithm based on Simulated Annealing. We call this algorithm EMSAGC (Expectation Maximization algorithm with Simulated Annealing for Graph Clustering).

Simulated Annealing (SA) is a strategy that has been exploited in order to improve the performance of optimization procedures since long [Eglese, 1990]. Often, in complex optimization problems the risk is that one gets trapped in local maxima of the objective function f. SA attempts to avoid this risk by proposing a move from the current local maximum $\hat{x}$ to a proposal $\tilde{x}$, and by allowing a positive probability to accept the move even when $f(\tilde{x}) < f(\hat{x})$.

The implementation of SA requires the definition of a strategy to propose a move $\tilde{x}$, as well as the choice of an acceptance probability function. Furthermore, many modifications of the basic SA algorithm can be implemented so as to improve the performance of SA (see

Eglese [1990] for an overview); among them, here we mention the possibilities to "store the best solution so far" and to consider more than one neighbour at a time.

Therefore, the implementation of SA within the algorithm of Section 5.3 requires:

- a method to select a proposal $\tilde{P}^t$. Here, we obtain $\tilde{P}^t$ by modifying the vector of probabilities $\tilde{p}_k^t = (\tilde{p}_{k1}^t, ..., \tilde{p}_{kM}^t)$ for one randomly picked graph $\mathcal{G}_k$, $k \in \{1, ..., K\}$ and keeping $\tilde{p}_s^t = p_s^t \ \forall s \neq k$. $\tilde{p}_k^t$ is chosen in such a way that $\tilde{p}_{km}^t \sim U(0,1) \, \forall m \in \{1, ..., M\}$ and $\sum_{m=1}^{M} \tilde{p}_{km}^t = 1$.
- the definition of an acceptance function, which we discuss below;
- the definition of any modification to the basic SA algorithm; in our implementation of the EMSAGC, we modify the algorithm so as to store the optima determined in each iteration and selecting, at the end of the iterations, the solution with the highest likelihood.

With respect to the definition of the acceptance function, two general properties are desirable. The first is that the probability of acceptance should be higher when $f(\tilde{x})$ is closer to $f(\hat{x})$. The second is that the probability of acceptance should be higher in the first iterations and then decrease: this is achieved by considering a positive, decreasing function $T(t)$ of t, called "temperature", which is higher in the first iterations of SA and then rapidly decreases in such a way that $T(t) \to 0$ for $t \to \infty$. The acceptance function that we consider hereafter is

$$a(\hat{x}, \tilde{x}, T(t)) = \left(\frac{f(\tilde{x})}{f(\hat{x})} \right)^{1/T(t)},$$

which clearly satisfies the two required properties. We take $T(t) = \frac{1}{\log t}$.

Keeping this in mind, we define the following Expectation Maximization algorithm with Simulated Annealing for Graph Clustering (EMSAGC):

1. for $k \in \{1, ..., K\}$ and $m \in \{1, ..., M\}$, define the initial probabilities $p_{km}^1 = Pr(z_{km} = 1)$. Denote by P^1 the $K \times M$ matrix which collects these probabilities;

2. for $t = 1, 2, ...$:

   □ **M step**.

   M1. Given P^t, estimate M network models (specified as GLMs) with weights given by $(p^t_{1m}, ..., p^t_{Km})$ for the m-th component and derive $\hat{\Theta}^t$.

   M2. If $t \geq 2$ and $L(\mathbf{Y}, Z|\hat{\Theta}^t) \leq L(\mathbf{Y}, Z|\hat{\Theta}^{t-1})$, consider the alternative state $\tilde{P}^t$ and determine $\tilde{\Theta}^t$:

   ⋆ if $L(\mathbf{Y}, Z|\tilde{\Theta}^t) \geq L(\mathbf{Y}, Z|\hat{\Theta}^t)$, set $\hat{\Theta}^t = \tilde{\Theta}^t$ and $P^t = \tilde{P}^t$.

   ⋆ if $L(\mathbf{Y}, Z|\tilde{\Theta}^t) < L(\mathbf{Y}, Z|\hat{\Theta}^t)$, set $\hat{\Theta}^t = \tilde{\Theta}^t$ and $P^t = \tilde{P}^t$ with probability equal to

$$\left(\frac{\log L(\mathbf{Y}, Z|\tilde{\Theta}^t)}{\log L(\mathbf{Y}, Z|\hat{\Theta}^t)} \right)^{1/T(t)}, \tag{5.10}$$

   where $T(t) = \frac{1}{\log t}$.

   □ **E step**. Given $\hat{\Theta}^t$, derive P^{t+1} as

$$P^{t+1}_{km} = \frac{Pr(\mathcal{G}_k|\hat{\theta}^t_m)}{\sum_{j=1}^{M} Pr(\mathcal{G}_k|\hat{\theta}^t_j)}. \tag{5.11}$$

3. Choose the best solution within the sequence $\{\hat{\Theta}^1, \hat{\Theta}^2, ..., \}$, i.e.

$$\hat{\Theta}^{EMSAGC} = \underset{t=1,2,...}{\operatorname{argmax}} L(\mathbf{Y}, Z|\hat{\Theta}^t). \tag{5.12}$$

5.4.2 Simulations

Here we reconsider the two scenarios which turned out to be problematic in Section 5.3.2: namely, sequences of binary graphs which we cluster with mixtures of logistic binomial models, and sequences of edge-valued graphs with overdispersed degree distribution which we cluster with mixtures of Poisson GLMs. K, v, π_1, π_2 and γ^m_{ij} are the same as in Section 5.3.2.

We consider the same starting points as before, but now we apply EMSAGC instead of the EM. We let the algorithm run for 300 iterations. The results, shown in Tables 5.3 and 5.4, clearly point out that

Table 5.3: **Average accuracy ($\bar{A}$) over 10 different starting points using the EMSAGC algorithm.** As a measure of accuracy of the EM algorithm, we average the accuracy of the solutions obtained from 10 different starting points. The number in brackets denote the variation with respect to the EM algorithm (Table 5.1).

K	v	Bernoulli	Negative Binomial
10	10	74% (+2)	86 % (+19)
10	20	100% (+19)	100% (+30)
10	50	99% (+31)	100% (+35)
10	100	98% (+33)	100% (+34)
20	10	100% (+8.5)	92% (+11.5)
20	20	100% (+11)	100% (+9)
20	50	100% (+15)	100% (+9.5)
20	100	100% (+19)	100% (+16.5)
50	10	100% (=)	98% (+1.6)
50	20	100% (=)	100% (=)
50	50	100% (=)	100% (=)
50	100	100% (=)	100% (=)

EMSAGC improves considerably the accuracy ($\bar{A}$) and the optimization performance (OP) of the EM algorithm, leading to a highly accurate clustering strategy even when the number of graphs K is small ($K = 10, 20$). Note that the case $K = 10,\ v = 10$ still turns out to be rather problematic: this seems to indicate that when only a few small graphs are at hand, even application of the EMSAGC might lead to inaccurate clusters.

5.5 Example application

We consider data on face-to-face contacts in an office building collected by Génois et al. [2015]. In this study, the employees of the French Institute for Public Health Surveillance were asked to wear sensors capable to measure face-to-face interactions that lasted at least 20 seconds. Measurements were collected for two weeks (10 working days) between June 24 and July 3, 2013.

Here, we focus on the comparison between the daily interaction networks. These networks are undirected and edge-valued; the edge weight is the number of interactions occurred between any two employees in a day. The study involved 92 employees, who belong to 5 different departments. However, for some individuals no daily interac-

Table 5.4: **Optimization performance (OP) of the EMSAGC algorithm.** OP is an indicator of the performance of the optimization. It is the number of cases in which the likelihood of the solution is non inferior to the likelihood of the true solution. As we consider 10 different starting points, $0 \leq OP \leq 10$. The number in brackets denote the variation with respect to the EM algorithm (Table 5.2).

K	v	Bernoulli	Negative Binomial
10	10	10 (+7)	9 (+7)
10	20	10 (+8)	10 (+9)
10	50	9 (+8)	10 (+9)
10	100	8 (+7)	10 (+9)
20	10	10 (+7)	8 (+7)
20	20	10 (+7)	10 (+2)
20	50	10 (+6)	10 (+2)
20	100	10 (+6)	10 (+4)
50	10	10 (+10)	10 (=)
50	20	10 (=)	10 (=)
50	50	10 (=)	10 (=)
50	100	10(=)	10 (=)

tions were recorded for several days (this makes us wonder whether they were not present, they did not wear the sensors, their sensors were not working or they simply did not have any interaction). Thus, we focus our attention only on the 68 employees for which interactions were recorded for more than half of the days considered (i.e., at least 6 days). These employees belong to four departments, which are described in Table 5.5. We remark that the results of the analysis do not change substantially if we consider only employees who had at least one interaction in at least 7, 8 or 9 days. They would be different, instead, if we were to restrict our attention to the 15 employees who had at least one interaction every day, because these employees belong to 3 departments only.

In this application, a known partition of employees in departments is available. We do not have any further information on the employees, besides their affiliation to the departments. It is important to take these two facts into account when choosing the specific network model that we specify for each subpopulation. In particular, the availability of a partition a priori of nodes induces us to consider a stochastic blockmodel. However, as discussed in Section 3.1.1, stochastic blockmodels imply a restrictive assumption of stochastic equivalence of employ-

Table 5.5: **Departments considered in our analysis.** Three departments are involved in the scientific production of the Institute, whereas one is responsible for the management of human resources. Two departments are located on the ground floor, and the remaining two on the first floor.

Abbreviation	Department name	Type of Dept.	Floor
DISQ	Scientific and Quality Direction	Scientific	0
DMCT	Dept. of Chronic Diseases and Traumas	Scientific	0
DSE	Dept. of Health and Environment	Scientific	1
SRH	Human Resources	Management	1

ees within each department, which appears to be unrealistic. For this reason, we consider the extended blockmodel with fixed effects proposed by Wang and Wong [1987]. The model was originally introduced for binary directed graphs, but here we adapt it to the case of edge-valued undirected graphs. Denote by (B_1, B_2, B_3, B_4) the four departments in Table 5.5. Then, for any two employees $i \in B_r$ and $j \in B_s$ ($r, s \in \{1, 2, 3, 4\}$) we let $Y_{ij} \sim Poi(\mu_{ij})$, with

$$\log(\mu_{ij}) = \alpha_i + \alpha_j + \phi_{rs}^{\text{I}}, \tag{5.13}$$

where $\sum_{r \leq s} \phi_{rs} = 0$.

We attempt to cluster the daily networks into two subpopulations, and to describe the difference between them. We remark that the aim of this example is to illustrate the proposed clustering strategy, rather than that of providing a detailed description of the interaction networks at hand. We consider 10 different starting points and for each of them we run the EMSAGC for 1000 iterations. 7 starting points yield a solution with loglikelihood equal to -19038, whereas solutions obtained from the remaining 3 starting points have lower likelihood. Therefore, this solution can be assumed to be the maximum likelihood estimate (although there is the possibility that it could be a local maximum).

The solution results into the following clusters: a first cluster consists of each of the days in the first week, as well as of Monday and Tuesday of the second week; the second cluster includes Wednesday, Thursday and Friday of the second week. Thus, the clustering method

[I]Note that the model could be equivalently parametrized as $\log(\mu_{ij}) = \theta_0 + \alpha_i + \alpha_j + \phi_{rs}$ under the additional constraint that $\sum_{i=1}^{v} \alpha_i = 0$.

Table 5.6: **Comparison of block-interactions between clusters.** Employees in departments DSE and DISQ interacted more within their department in the first 7 days, and more between each other in the last 3 days (corresponding parameters are emphasised in bold). Conversely, employees in DMCT and SRH interact more with each other in the first 7 days, and within their own departments in the last 3 days (corresponding parameters are underlined).

Parameter	Estimates	
	Cluster 1	Cluster 2
DMCT	0.71	0.86
DSE	**0.92**	**0.58**
DISQ	**1.43**	**1.10**
SRH	1.83	2.01
DMCT-DSE	-0.15	-0.12
DMCT-DISQ	-0.22	-0.18
DMCT-SRH	-0.34	-0.56
DSE-DISQ	**-0.24**	**0.03**
DSE-SRH	-0.53	-0.49
DISQ-SRH	-0.96	-0.96
$\hat{\sigma}^2_\alpha$	0.030	0.094

seems to detect a change in the interaction patterns between the first 7 days considered, and the final 3 days.

Table 5.6 compares the estimates of the block-interaction parameters in the two clusters. Overall, we find two changes in the pattern of interaction across departments. On the one hand, it seems that members of DISQ and DSE were more active within their department in the first 7 days considered, but then interacted more with each other in the remaining 3 days. On the other hand, employees in DMCT and SRH seem to follow the opposite pattern: in the last 3 days, they reduce interactions between departments and are more active within their own department. Moreover, the variance of the fixed effects ($\hat{\sigma}^2_\alpha$) appears to be higher in cluster 2: this result indicates that the degree distribution became more skewed in the last three days.

Finally, note that the pattern of interactions between departments does not appear to be influenced by their location in the ground or first floor. Instead, it seems that interactions are stronger between the three scientific departments (DMCT, DSE and DISQ) and weaker with the human resources (SRH), which (as a result) features a higher internal connectivity.

5.6 Concluding remarks and future work

In this Chapter, we have considered a collection of graphs defined on the same set of vertices, which we have defined by means of a sequence of graphs $\mathcal{S} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_K\}$ such that $\mathcal{G}_k = (V, E_k),\ k \in \{1, ..., K\}$.

Building on the fact that many network models (e.g., the p_1 and p_2 models and stochastic blockmodels) can be implemented within the framework of generalized linear models, we have proposed to jointly model all the graphs in the sequence $\mathcal{S}$ using a mixture of generalized linear models, where each component $f\,(Y|\theta_m)$ in the mixture is given by a network model of interest for a given subpopulation $\mathcal{S}_m$ of graphs. This model allows to estimate the probability that a graph $\mathcal{G}_k$ belongs to a certain subpopulation, and it can thus be used to cluster the graphs within the sequence. Moreover, it allows to characterize each subpopulation by means of the model estimates $f(Y|\hat{\theta}_m)$.

Since the likelihood of the proposed model depends both on observed data (the graphs) and latent variables (the identifying labels indicating which graph belongs to which subpopulation), we have implemented an EM algorithm [Dempster et al., 1977] to estimate the mixture components and clusters. Our simulations indicate that even though this algorithm seems to perform generally well when the sequence consists of a relatively large number of graphs ($K = 50$), for smaller graph sequences ($K = 10$ or $K = 20$) the accuracy of the resulting clustering appears to be rather low for binary graphs and for edge-valued graphs whose degree distribution is highly overdispersed.

With the aim of improving the performance of the optimization of the likelihood, as well as the accuracy of the induced clusters, we have thus proposed an alternative algorithm, which we call EMSAGC. EMSAGC is an extension of the EM algorithm, which integrates it with a Simulated Annealing [Eglese, 1990] strategy. The recourse to Simulated Annealing is motivated by the need to ensure a wider exploration of the likelihood surface than that performed by the simple EM. Indeed, EMSAGC appears to improve considerably both the optimization, as well as the accuracy of the resulting clusters.

Although the simulations presented in Sections 5.3.2 and 5.4.2 focus on a scenario where $\mathcal{S}$ features the presence of two subpopulations of

graphs, future work includes the evaluation of the performance of the proposed EM and EMSAGC algorithms in more complex scenarios, with 3 or more subpopulations.

In conclusion, we observe that, in principle, mixture models could be employed to cluster graphs also in conjunction with network models that cannot (or should not) be specified as generalized linear models, such as Exponential Random Graphs (ERGMs). However, the large number of iterations involved in the EMSAGC algorithm require the estimation of several network models and this currently prevents the use of ERGMs therein, because of the computational burden that MCMC estimation of ERGMs requires. A compromise strategy could be that of estimating ERGMs with pseudolikelihood, which allows to resort to GLM routines and is rather inexpensive computationally: however, one should bear in mind the fact that pseudolikelihood is known to yield biased parameter estimates for ERGMs when attempting this approach.

Bibliography

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014.

Barabasi, A.-L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B*, pages 1–38.

Durante, D., Dunson, D. B., and Vogelstein, J. T. (2016a). Nonparametric bayes modeling of populations of networks. *arXiv preprint arXiv:1406.7851*.

Durante, D., Paganin, S., Scarpa, B., and Dunson, D. B. (2016b). Bayesian modeling of networks in complex business intelligence problems. *Journal of the Royal Statistical Society. Series C*.

Eglese, R. (1990). Simulated annealing: a tool for operational research. *European Journal of Operational Research*, 46(3):271–281.

Frank, O. and Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association*, 81(395):832–842.

Génois, M., Vestergaard, C. L., Fournet, J., Panisson, A., Bonmarin, I., and Barrat, A. (2015). Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. *Network Science*, 3(03):326–347.

Grün, B. and Leisch, F. (2008). Finite mixtures of generalized linear regression models. In *Recent advances in linear models and related areas*, pages 205–230. Springer.

Guimera, R., Mossa, S., Turtschi, A., and Amaral, L. N. (2005). The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences*, 102(22):7794–7799.

Hanneke, S., Fu, W., Xing, E. P., et al. (2010). Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4:585–605.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137.

Holland, P. W. and Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50.

Klovdahl, A. S. (1985). Social networks and the spread of infectious diseases: the AIDS example. *Social Science & Medicine*, 21(11):1203–1216.

Matias, C. and Miele, V. (2017). Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society. Series B*.

McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models*, volume 37. CRC press.

Moreno, J. L. (1934). *Who shall survive?*, volume 58. Nervous and mental disease monograph series.

Newman, M. (2010). *Networks: an introduction*. Oxford University Press.

Sewell, D. K. and Chen, Y. (2015). Latent space models for dynamic networks. *Journal of the American Statistical Association*, 110(512):1646–1657.

Signorelli, M., Vinciotti, V., and Wit, E. (2016). Neat: an efficient network enrichment analysis test. *BMC Bioinformatics*, 17(352):1–17.

Snijders, T. A. (2001). The statistical evaluation of social network dynamics. *Sociological Methodology*, 31(1):361–395.

van Duijn, M. A., Snijders, T. A., and Zijlstra, B. J. (2004). p2: a random effects model with covariates for directed graphs. *Statistica Neerlandica*, 58(2):234–254.

Wang, Y. J. and Wong, G. Y. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19.

Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and applications*, volume 8. Cambridge University Press.

Appendix A

Vignettes of the R package neat

An introduction to the R package neat

Mirko Signorelli

Introduction

What's neat?

neat is the R package that implements NEAT, the Network Enrichment Analysis Test which is presented in Signorelli, M., Vinciotti, V., Wit, E. C. (2016). *NEAT: an efficient network enrichment analysis test.* BMC Bioinformatics, 17:352.

The article is freely available from the website of BMC Bioinformatics.

What's "network" enrichment analysis?

Network enrichment analysis is an extension of traditional gene enrichment analysis (GEA) tests, which are typically used to provide a characterization of a target gene set by relating it to gene sets (such as Gene Ontologies or KEGG pathways) whose function is already known.

A known limitation of GEA tests is that they ignore associations and dependences between genes. The purpose of network enrichment analysis is thus to integrate GEA tests with information on known relations between genes, represented by means of a gene network.

Loosely speaking, we can say that network enrichment analysis incorporates genetic networks, with their information on gene dependences, into gene enrichment tests. Hence, the name "network" enrichment analysis.

Get started

In order to be able to use the package, you need to install it with

```
install.packages('neat')
```

and, then, to load it with the command

```
library('neat')
```

A first example

Let's first have a quick look at an example of how a network enrichment analysis can be carried out with NEAT.

The analysis will typically consist of three steps: preparation of the data, computation of the test and inspection of the results.

Preparation of the data

Let's start by loading `yeast`, a list which contains the data that we will need for the analysis:

```
data(yeast) # load the data
ls(yeast) # display the content of the list
```

```
## [1] "esr1"       "esr2"       "goslimproc" "kegg"       "yeastnet"
## [6] "ynetgenes"
```

Let's say that we are interested to know whether a set of differentially expressed genes, `yeast$esr2`, can be related to some functional gene sets contained in `yeast$goslimproc`. Let's focus the attention on two of these processes, namely 'response to heat' and 'response to starvation'.

Before we can proceed with the analysis, we have to create two lists of gene sets, one (which we will call `induced_genes`) containing the set of differentially expressed genes and the other (called `functional_sets`) with the functional sets of interest:

```
induced_genes = list('ESR 2' = yeast$esr2) # set of differentially expressed genes
#(ESR 2 is the set of induced ESR genes)
functional_sets = yeast$goslimproc[c(72,75)] # two functional gene sets of interest:
#response to heat and to starvation
```

Besides these two lists, we will need two further objects:

- `yeast$yeastnet`, a two-column matrix that contains YeastNet (a network incorporating known functional couplings between yeast genes, see the help page `?yeast` for more details);
- `yeast$ynetgenes`, a vector containing the names of all the genes that are present in the network.

Computation of the test

The idea behind NEAT is that if two gene sets are related, then in the network we expect them to be connected by a larger (or smaller) number of links than we would expect to observe by chance. Our null hypothesis, thus, is that if A and B are unrelated, then links are randomly placed between the two groups, so that the total number of links between A and B can be assumed to follow an hypergeometric distribution.

If, however, the number of links that we actually observe between A and B turns out to be significantly different from what we would expect to get if links were placed randomly, then we take this fact as potential evidence of a relation between the two groups and we say that there is "enrichment" between them.

The computation of the test can be done with the function `neat` as follows:

```
test = neat(alist = induced_genes, blist = functional_sets, network = yeast$yeastnet,
            nettype = 'undirected', nodes = yeast$ynetgenes, alpha = 0.01)
```

Analysis of the results

The results are now saved in the object `test`, which we can display with the command `print`:

```
print(test)
```

```
##       A                       B nab expected_nab pvalue     conclusion
## 1 ESR 2       response_to_heat  86         96.9 0.2518  No enrichment
## 2 ESR 2 response_to_starvation 459        331.4 0.0000 Overenrichment
```

From the table we can see that the set of differentially expressed genes (`ESR 2`) is not enriched with respect to the set of genes involved in response to heating, whereas it is overenriched with respect to the set of genes that are responsible for response to starvation (that is to say, the observed number of links, 459, is significantly higher than what we would expect to get by chance, i.e. 331). Thus, we can conclude that genes in `ESR 2` are regulated when the yeast cell is exposed to starvation, but not when exposed to heating.

A closer look to the package

The core of the package is the function `neat`:

```
neat(alist, blist, network, nettype, nodes, alpha = NULL,
     anames = NULL, bnames = NULL)
```

The fundamental arguments of the function are:

- `alist` and `blist`, two lists of gene sets;
- `network`, which can be specified in three different formats;
- `nettype`, either `'undirected'` or `'directed'`;
- `nodes`, a vector containing the names of all nodes in the network.

Moreover, three optional arguments are `alpha`, which allows to specify the significance level of the test, and `anames` and `bnames` (they can be used to name the elements of `alist` and `blist`, if not already named).

As a (toy) example, let's consider a partially directed network with 7 nodes defined by the following adjacency matrix

```
A = matrix(0, nrow=7, ncol=7)
labels = letters[1:7]
rownames(A) = labels; colnames(A) = labels
A[1,c(2,3)]=1; A[2,c(5,7)]=1;A[3,c(1,4)]=1;A[4,c(2,5,7)]=1;A[6,c(2,5)]=1;A[7,4]=1
print(A)
```

```
##   a b c d e f g
## a 0 1 1 0 0 0 0
## b 0 0 0 0 1 0 1
## c 1 0 0 1 0 0 0
## d 0 1 0 0 1 0 1
## e 0 0 0 0 0 0 0
## f 0 1 0 0 1 0 0
## g 0 0 0 1 0 0 0
```

How to specify the lists of gene sets

Let's consider three sets of genes {a,e}, {c,g} and {d,f} and suppose we want to test whether there is enrichment from the first two sets to the third one.

First of all, let's create a vector for each of the three sets:

```
set1 = c('a','e')
set2 = c('c','g')
set3 = c('d','f')
```

As we want to know whether there is enrichment from `set1` and `set2` to `set3`, we can create two gene lists, one (`alist`) containing `set1` and `set2` and the other (`blist`) containing `set3`:

```
alist = list('set 1' = set1, 'set 2' = set2)
blist = list('set 3' = set3)
```

Alternative network formats

Above we have defined the network with its adjacency matrix `A`. However, the network can be passed to `neat` in three alternative formats:

- a sparse adjacency matrix, e.g.

```
library(Matrix)
as(A, 'sparseMatrix')
```

```
## 7 x 7 sparse Matrix of class "dgCMatrix"
##   a b c d e f g
## a . 1 1 . . . .
## b . . . . 1 . 1
## c 1 . . 1 . . .
## d . 1 . . 1 . 1
## e . . . . . . .
## f . 1 . . 1 . .
## g . . . 1 . . .
```

- an `igraph` graph;
- a two-column matrix where every row represents and edge (for directed and mixed networks, parent nodes must be in the first column, and child nodes in the second), e.g.:

```
##       [,1] [,2]
##  [1,] "a"  "b"
##  [2,] "a"  "c"
##  [3,] "b"  "e"
##  [4,] "b"  "g"
##  [5,] "c"  "a"
##  [6,] "c"  "d"
##  [7,] "d"  "b"
##  [8,] "d"  "e"
##  [9,] "d"  "g"
## [10,] "f"  "b"
## [11,] "f"  "e"
## [12,] "g"  "d"
```

Network type

Set the argument `nettype` equal to `'undirected'` if an undirected network is at hand, and equal to `'directed'` if you are considering a directed or partially directed network.

Compute the test

Once you have prepared the lists of gene sets and the network, what you need is to run `neat`, without forgetting to specify the correct `nettype` (here `nettype = 'directed'`) and the labels of nodes (here `nodes = labels`):

```
test1 = neat(alist = alist, blist = blist, network = A,
             nettype = 'directed', nodes = labels)
print(test1)
```

```
##       A     B nab expected_nab     pvalue
## 1 set 1 set 3   0       0.3333 0.68181818
```

```
## 2 set 2 set 3   2       0.5000 0.04545455
```

If you want to add to the results a column specifying the conclusion of the test (overenrichment, no enrichment or underenrichment) for a given significance level, you use the option `alpha`:

```
test2 = neat(alist = alist, blist = blist, network = A,
             nettype = 'directed', nodes = labels, alpha = 0.05)
print(test2)
```

```
##       A     B nab expected_nab     pvalue     conclusion
## 1 set 1 set 3   0       0.3333 0.68181818  No enrichment
## 2 set 2 set 3   2       0.5000 0.04545455 Overenrichment
```

Further details and material

The aim of this vignette is to provide a quick introduction to the computation of NEAT using R. Here I focused my attention on the fundamental aspects that one needs to use the package.

Further details, functions and examples can be found in the manual of the package.

The description of the method is available in an article which you can read here. A shorter version of the paper was presented at the 31st IWSM and published in the Conference proceedings.

Appendix B

Manual of the R package neat

Package 'neat'

September 7, 2016

Type Package

Title Efficient Network Enrichment Analysis Test

Version 1.0

Date 2016-09-07

Depends R (>= 3.3.0)

Author Mirko Signorelli, Veronica Vinciotti and Ernst C. Wit

Maintainer Mirko Signorelli <m.signorelli@rug.nl>

URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1203-6, http://mirkosignorelli.wixsite.com/home/software

Description Includes functions and examples to compute NEAT, a network-based test for gene enrichment analysis (Signorelli, Vinciotti and Wit, 2016, <DOI:10.1186/s12859-016-1203-6>).

Suggests igraph, Matrix, knitr, rmarkdown

VignetteBuilder knitr

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-07 08:49:09

R topics documented:

neat-package 2
neat 2
networkmatrix 5
plot.neat 6
print.neat 7
summary.neat 8
yeast 9

Index **12**

neat-package *neat*

Description

Includes functions and examples to compute NEAT (Network Enrichment Analysis Test), a network-based test for genetic enrichment analysis (Signorelli et al., 2016).

Author(s)

Mirko Signorelli

References

Signorelli, M., Vinciotti, V., Wit, E. C. (2016). NEAT: an efficient network enrichment analysis test. BMC Bioinformatics, 17:352. Url: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1203-6.

See Also

`neat`

`neat` *Performs neat for lists of gene sets*

Description

Compute NEAT (Signorelli et al., 2016), a test for network enrichment analysis between/from a first list of sets ('A sets') and/to a second list of sets ('B sets').

Usage

```
neat(alist, blist = NULL, network, nettype, nodes, alpha = NULL,
     anames = NULL, bnames = NULL)
```

Arguments

`alist`	List of A sets. Each element within the list is a vector of genes and represents a gene set
`blist`	List of B sets. Each element within the list is a vector of genes and represents a gene set. If `nettype == "undirected"`, this argument is optional: if provided, every set of `blist` is compared with every set of `alist`; if `NULL`, the function compares sets in `alist` between themselves

network	One of the following objects: an adjacency matrix of class "matrix" (see 'Example 1') or a sparse adjacency matrix of class "dgCMatrix"; an igraph object (see 'Example 2'); a two-column matrix where every row represents and edge (for directed networks, parent nodes must be in the first column, and child nodes in the second)
nettype	Either 'directed' or 'undirected'
nodes	Vector containing the (ordered) names of all nodes in the network
alpha	Significance level of the test (optional). If specified, a column with the conclusion of the test is added to the output
anames	Vector of names for the elements of alist (optional: it has to be provided only if the elements of alist are not named)
bnames	Vector of names for the elements of blist (optional: it has to be provided only if the elements of blist are not named)

Value

A data frame with the following columns:

A	A set
B	B set
nab	observed number of links from A to B
expected_nab	expected number of links from A to B (in absence of enrichment)
pvalue	p-value of the test
conclusion	conclusion of the test (only if alpha is specified): no enrichment, overenrichment or underenrichment

Author(s)

Mirko Signorelli

References

Signorelli, M., Vinciotti, V., Wit, E. C. (2016). NEAT: an efficient network enrichment analysis test. BMC Bioinformatics, 17:352. Url: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1203-6.

See Also

networkmatrix, plot.neat, print.neat, summary.neat

Examples

```
# Example 1: network given as adjacency matrix:
A = matrix(0, nrow=7, ncol=7)
A[1,c(2,3)]=1; A[2,c(5,7)]=1;A[3,c(1,4)]=1;A[4,c(2,5,7)]=1;A[6,c(2,5)]=1;A[7,4]=1
labels = letters[1:7]
set1 = c('a','e')
```

```
set2 = c('c','g')
set3 = c('d','f')
alist = list('set 1' = set1, 'set 2' = set2)
blist = list('set 3' = set3)

test1 = neat(alist = alist, blist = blist, network=A,
             nettype='directed', nodes=labels, alpha=0.05)
print(test1)

# Example 2: network given as igraph object:
library(igraph)
network = erdos.renyi.game(15, 1/3)
set1 = 1:4
set2 = c(2,5,13)
set3 = c(3,9,14)
set4 = c(8,15,20)
alist = list('set 1' = set1, 'set 2' = set2)
blist = list('set 3' = set3, 'set 4' = set4)

test2 = neat(alist, blist, network = network,
             nettype='undirected', nodes=seq(1,15), alpha=NULL)
print(test2)

# Example 3: network given as list of links:
networklist = matrix(nrow=13, ncol=2)
networklist[,1]=c('a','a','b','b','c','d','d','d','f','f','f','h','h')
networklist[,2]=c('d','e','e','g','d','b','e','g','a','b','e','c','g')

labels = letters[1:8]
set1 = c('a','b','e')
set2 = c('c','g')
set3 = c('d','f')
set4 = c('a','b','f')
alist = list('set 1' = set1, 'set 2' = set2)
blist = list('set 3' = set3, 'set4' = set4)

test3 = neat(alist, blist, network = networklist,
             nettype = 'undirected', nodes=labels, alpha=0.05)
print(test3)

alist = list('set 1' = set1, 'set 2' = set2, 'set 3' = set3)
test4 = neat(alist, network = networklist,
             nettype = 'undirected', nodes=labels, alpha=0.05)
print(test4)

# Example 4: ESR data
## Not run:
data(yeast)
esr = list('ESR 1' = yeast$esr1, 'ESR 2' = yeast$esr2)
test = neat(alist = esr, blist = yeast$goslimproc, network = yeast$yeastnet,
            nettype = 'undirected', nodes = yeast$ynetgenes, alpha = 0.01)
# Replace with "blist = yeast$kegg" to use kegg pathways
```

```
m = dim(test)[1]
test1 = test[1:(m/2),]
table(test1$conclusion)
plot(test1)
o1=test1[test1$conclusion=='Overenrichment',]
print(o1, nrows='ALL') #display overenrichments

test2 = test[(m/2+1):m,]
table(test2$conclusion)
plot(test2)
o2=test2[test2$conclusion=='Overenrichment',]
print(o2, nrows='ALL') #display overenrichments

## End(Not run)
```

`networkmatrix` *Creates a network matrix for* `neat`

Description

Internal function, creates a two-column network matrix that can be further processed by `neat`.

Usage

```
networkmatrix(network, nodes, nettype)
```

Arguments

`network`	One of the following objects: an adjacency matrix (class `"matrix"`), a sparse adjacency matrix (class `"dgCMatrix"`) or an igraph graph (class `"igraph"`)
`nodes`	Vector containing the (ordered) names of all nodes in the network
`nettype`	Either `'directed'` or `'undirected'`

Details

This is an internal function, that is called within `neat` to convert different types of network objects (see argument 'network' above) into a standard two-column network matrix, that can then be processed by `neat`.

Value

A two-column matrix, where every row represents and edge. For directed networks, parent nodes must be in the first column, and child nodes in the second.

Author(s)

Mirko Signorelli

References

Signorelli, M., Vinciotti, V., Wit, E. C. (2016). NEAT: an efficient network enrichment analysis test. BMC Bioinformatics, 17:352. Url: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1203-6.

See Also

neat

Examples

```
# First case: adjacency matrix
n<-50
adjacency <- matrix(sample(0:1, n^2, replace=TRUE, prob=c(0.9,0.1)), ncol=n)
diag(adjacency) <- 0
lab = paste(rep('gene'),1:n)
head(networkmatrix(adjacency, lab, 'directed'))

# Second case: sparse adjacency matrix
library(Matrix)
sparse_adjacency<-Matrix(adjacency,sparse=TRUE)
head(networkmatrix(sparse_adjacency, lab, 'directed'))

# Third case: igraph object
library(igraph)
igraph_graph = erdos.renyi.game(15, 1/3)
lab = paste(rep('gene'),1:15)
head(networkmatrix(igraph_graph, lab, 'directed'))
```

---

plot.neat *Plot method of* neat

Description

plot method for class "neat".

Usage

```
## S3 method for class 'neat'
plot(x, nbreaks = 10, ...)
```

Arguments

x	An object of class "neat"
nbreaks	Number of breaks to be used in the histogram (default is 10)
...	Further arguments passed to or from other methods

Value

An histogram showing the distribution of p-values and a p-p plot comparing the distribution of p-values to the uniform distribution.

Author(s)

Mirko Signorelli

References

Signorelli, M., Vinciotti, V., Wit, E. C. (2016). NEAT: an efficient network enrichment analysis test. BMC Bioinformatics, 17:352. Url: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1203-6.

See Also

`neat`, `print.neat`, `summary.neat`

Examples

```
## Not run:
data(yeast)
esr2 = list('ESR 2' = yeast$esr2)

test = neat(alist = esr2, blist = yeast$goslimproc, network = yeast$yeastnet,
            nettype='undirected', nodes = yeast$ynetgenes, alpha = 0.01)

plot(test)

## End(Not run)
```

---

print.neat *Print method of* neat

Description

`print` method for class "`neat`".

Usage

```
## S3 method for class 'neat'
print(x, nrows=10, ...)
```

Arguments

`x`	An object of class "`neat`"
`nrows`	Maximum number of results to print (default is 10). It can be either an integer number or "ALL"
`...`	Further arguments passed to or from other methods

Value

A dataframe showing the first nrows tests contained in a neat object.

Author(s)

Mirko Signorelli

References

Signorelli, M., Vinciotti, V., Wit, E. C. (2016). NEAT: an efficient network enrichment analysis test. BMC Bioinformatics, 17:352. Url: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1203-6.

See Also

neat, plot.neat, summary.neat

Examples

```
A = matrix(0, nrow=7, ncol=7)
A[1,c(2,3)]=1; A[2,c(5,7)]=1;A[3,c(1,4)]=1;A[4,c(2,5,7)]=1;A[6,c(2,5)]=1;A[7,4]=1

labels = letters[1:7]
set1 = c('a','e')
set2 = c('c','g')
set3 = c('d','f')
alist = list('set 1' = set1, 'set 2' = set2)
blist = list('set 3' = set3)

test = neat(alist, blist, network=A, nettype='directed', nodes=labels, alpha=0.05)
print(test)
```

---

summary.neat *Summary method of* neat

Description

summary method for class "neat".

Usage

```
## S3 method for class 'neat'
summary(object, ...)
```

Arguments

object	An object of class "neat"
...	Further arguments passed to or from other methods

Value

The summary.neat function returns the following values:

- the number of tests computed;
- the number of enrichments at 1% and 5% level;
- the p-value of the Kolmogorov-Smirnov test to check if the distribution of p-values is uniform.

Author(s)

Mirko Signorelli

References

Signorelli, M., Vinciotti, V., Wit, E. C. (2016). NEAT: an efficient network enrichment analysis test. BMC Bioinformatics, 17:352. Url: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1203-6.

See Also

neat, plot.neat, summary.neat

Examples

```
## Not run:
data(yeast)
esr = list('ESR 1' = yeast$esr1, 'ESR 2' = yeast$esr2)
test = neat(alist = esr, blist = yeast$goslimproc, network = yeast$yeastnet,
            nettype = 'undirected', nodes = yeast$ynetgenes, alpha = 0.01)

test1 = test[1:99,]
summary(test1)

test2 = test[100:198,]
summary(test2)

## End(Not run)
```

---

yeast *List collecting various yeast data (see 'description')*

Description

yeast is a list that contains:

yeastnet: network matrix representing Yeastnet-v3 (Kim et al., 2013)

ynetgenes: vector with the names of the genes appearing in yeastnet

esr1: vector containing the first of the two gene sets that constitute the "Environmental Stress Response" (ESR) reported by Gasch et al. (2012)

esr2: vector containing the second gene set of the ESR

goslimproc: list containing the gene sets of the GOslim process ontology (Ashburner et al., 2000) for the buddying yeast Saccaromyces Cerevisiae (groups 'biological process' and 'other' are not included)

kegg: list containing the KEGG pathways (Kanehisa and Goto, 2002) for the buddying yeast Saccaromyces Cerevisiae

Format

yeast: list

Source

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. Nat. Genet., 25(1), 25-29.

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., and Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. Mol. Biol. Cell, 11(12), 4241-4257.

Kanehisa, M., and Goto, S. (2002). KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res., 28(1), 27-30.

Kim, H., Shin, J., Kim, E., Kim, H., Hwang, S., Shim, J. E., and Lee, I. (2013). Yeastnet v3: a public database of data-specific and integrated functional gene networks for saccharomyces cerevisiae. Nucleic Acids Res., 42 (D1), D731-6.

Signorelli, M., Vinciotti, V., Wit, E. C. (2016). NEAT: an efficient network enrichment analysis test. BMC Bioinformatics, 17:352. Url: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1203-6.

References

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. Nat. Genet., 25(1), 25-29.

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., and Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. Mol. Biol. Cell, 11(12), 4241-4257.

Kanehisa, M., and Goto, S. (2002). KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res., 28(1), 27-30.

Kim, H., Shin, J., Kim, E., Kim, H., Hwang, S., Shim, J. E., and Lee, I. (2013). Yeastnet v3: a public database of data-specific and integrated functional gene networks for saccharomyces cerevisiae. Nucleic Acids Res., 42 (D1), D731-6.

Signorelli, M., Vinciotti, V., Wit, E. C. (2016). NEAT: an efficient network enrichment analysis test. BMC Bioinformatics, 17:352. Url: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1203-6.

See Also

neat

Examples

```
## Not run:
data(yeast)
esr = list('ESR 1' = yeast$esr1, 'ESR 2' = yeast$esr2)
test = neat(alist = esr, blist = yeast$goslimproc, network = yeast$yeastnet,
            nettype = 'undirected', nodes = yeast$ynetgenes, alpha = 0.01)
# Replace with "blist = yeast$kegg" to use kegg pathways

m = dim(test)[1]
test1 = test[1:(m/2),]
o1=test1[test1$conclusion=='Overenrichment',]
# list of overenrichments for the first ESR set:
print(o1, nrows='ALL')

test2 = test[(m/2+1):m,]
o2=test2[test2$conclusion=='Overenrichment',]
# list of overenrichments for the second ESR set:
print(o2, nrows='ALL')

# the same can be done using KEGG pathways:
keggtest = neat(alist = esr, blist = yeast$kegg, network = yeast$yeastnet,
            nettype = 'undirected', nodes = yeast$ynetgenes, alpha = 0.01)

## End(Not run)
```

Index

∗Topic **datasets**
yeast, 9
∗Topic **htest**
neat, 2
∗Topic **manip**
networkmatrix, 5
∗Topic **methods**
plot.neat, 6
print.neat, 7
summary.neat, 8
∗Topic **package**
neat-package, 2

neat, *2*, 2, *5–9*, *11*
neat-package, 2
neatc *(*neat*)*, 2
networkmatrix, *3*, 5

plot.neat, *3*, 6, *8, 9*
print.neat, *3*, *7*, 7
pvalue *(*neat*)*, 2

summary.neat, *3*, *7, 8*, 8, *9*

yeast, 9

Appendix C

Curriculum Vitae

Mirko Signorelli

CURRICULUM VITAE

Contact Information

• University of Groningen
Johann Bernoulli Institute for Mathematics and Computer Science
Nijenborgh 9 (Bernoulliborg)
9747 AG Groningen (The Netherlands)

• University of Padova
Department of Statistical Sciences
Via Cesare Battisti 241-243
35121 Padova (Italy)

e-mail: m.signorelli@rug.nl, signorelli@stat.unipd.it
Phone: +31 50 3633536

Current Position

Since January 2014 (expected completion: April 2017)
PhD Student in Statistical Sciences
Universities of Padova and Groningen
Thesis title: "Inferring Community-driven Structure in Complex Networks"
Supervisors: Prof. Monica Chiogna and Prof. Ernst C. Wit

Research interests

- Statistical network science
- Penalized inference
- Generalized, linear, and mixed models
- Model-based clustering
- Multiway statistical models
- Statistical applications in biology, economics, sociology and political science

Education

September 2011 – July 2013
Master degree in Statistical and Economic Sciences
University of Milano-Bicocca, Faculty of Statistics
Title of dissertation: "Metodi statistici multivariati e multiway per l'analisi dei bilanci dei comuni italiani"
Supervisor: Dr. Simona C. Minotti
Final mark: 110/110 cum laude

October 2008 – July 2011
Bachelor degree in Statistical and Economic Sciences
University of Milano-Bicocca, Faculty of Statistics
Title of dissertation: "Studio dello stato di salute di un settore economico mediante dati di bilancio: contabilità macroeconomica e analisi delle componenti principali a tre vie"
Supervisor: Dr. Simona C. Minotti
Final mark: 110/110 cum laude

Work experience

February – April 2012 & February – May 2013
Faculty of Statistics, University of Milano-Bicocca.
Teaching assistant for the course "Statistics II".

November 2012 – April 2013
University of Milano-Bicocca.
Computer lab manager.

May 2011 – October 2012
Municipality of Chiuduno.
City councilman and budget committee member.

February – May 2011
Chamber of Commerce of Bergamo.
Internship at the Statistical office of the Chamber of Commerce.

Awards and Scholarship

February 2017
Funding for a Short Term Scientific Mission at the University of Sheffield from the COSTNET Programme.

August and September 2016
Conference grants: Workshop on "Network Science and its Applications" (Cambridge, Aug. 2016) and First Meeting of the European Cooperation for Statistics of Network Data Science (Ribno, Sep. 2016).

2014–2016
PhD scholarship and funding for conferences, Universities of Padova and Groningen.

2015
Erasmus scholarship, University of Padova.

2008–13
Merit scholarships awarded by the University of Milano-Bicocca.

2008
Award from the Italian Ministry of Education: Register of excellent high school students award (*"Registro delle eccellenze"*)

2003-08
Merit scholarships and prizes awarded by Regione Lombardia and the municipality of Chiuduno.

Computer skills

- I am a rather experienced `R` user, and I am the author of the `R` package `neat`.
- I also have a good knowledge of SAS (I am a "Certified Base Programmer for SAS 9").
- I have experience with high performance computer clusters and parallel computing.
- Further statitical software that I sometimes use: Stata, SPSS, Matlab, Gretl, EViews, Gephi.
- I am a LaTeXuser. I have a limited experience with SQL and Python languages.

Language skills

- Italian: mother tongue.
- English: working proficiency (CEFR level: C1).
- Dutch: elementary proficiency (CEFR level: A1).

Publications

- Signorelli, M., Vinciotti, V., Wit, E. C. (2016). NEAT: an efficient network enrichment analysis test. *BMC Bioinformatics*, 17:352, DOI: 10.1186/s12859-016-1203-6.
- Signorelli, M., Wit, E. C. (submitted). A penalized inference approach to stochastic block-modelling of community structure in the Italian Parliament. *ArXiv preprint:* arXiv:1607.08743
- Signorelli, M., Vinciotti, V., Wit, E. C. (2016). NEAT: an efficient Network Enrichment Analysis Test. *Proceedings of the 31st International Workshop on Statistical Modelling*, vol. 1, pp. 289-294.

Conference presentations

Signorelli, M., Wit, Ernst C. (2016). Reconstructing collaborations between political parties from bill cosponsorship networks (invited talk). *First Meeting of the European Cooperation for Statistics of Network Data Science (COSTNET)*, Ribno, Slovenia, September 2016.

Signorelli, M., Wit, Ernst C. (2016). Modelling community structure in the Italian Parliament: a penalized inference approach (invited talk). *Workshop on Network Science and its applications*, Isaac Newton Institute, Cambridge, United Kingdom, August 2016.

Signorelli, M., Vinciotti, V., Wit, Ernst C. (2016). NEAT: an efficient Network Enrichment Analysis Test (contributed talk). *31st International Workshop on Statistical Modelling*, Rennes, France, July 2016.

Signorelli, M., Vinciotti, V., Wit, Ernst C. (2015). PNEA: a Parametric approach to Network Enrichment Analysis (poster). *44th annual Meeting of Dutch Statisticians and Probabilists*, Lunteren, The Netherlands, November 2015.

Teaching experience

May 2015 – February 2017
Co-supervision of two master students involved in research projects on the analysis of social and biological networks
Degrees: Master programmes in "Mathematics" and in "Education and Communication in Mathematics and Natural Sciences"
Faculty of Mathematics and Natural Sciences, University of Groningen

February – April 2012 & February – May 2013
Course name: Statistics II
Degrees: Bachelor programmes in "Statistical and Economic Sciences" and "Statistics and Information Management"
Teaching assistant for the laboratory module (50 hours in total)
Faculty of Statistics, University of Milano-Bicocca
Instructors: Prof. Donata Marasini and Prof. Piergiorgio Lovaglio

References

Prof. Ernst C. Wit
Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen
Nijenborgh 9 (Bernoulliborg), 9747 AG Groningen (NL)
Contact: e.c.wit@rug.nl
Phone: +31 50 363 5170

Prof. Monica Chiogna
Department of Statistical Sciences, University of Padova
Via Cesare Battisti 241-243, 35121 Padova (IT)
E-mail: monica@stat.unipd.it
Phone: +39 49 8274183

Abstract

Despite a long tradition in the study of graphs and relational data, for decades the analysis of complex networks was limited by difficulties in data collection and computational burdens. The advent of new technologies in life sciences, as well as in our daily life, has suddenly shed light on the many interconnections that our world features, from friendships and collaborations between individuals or organizations, to functional couplings between cellular molecules. This has highly facilitated the collection of relational data, fostering an unprecedented interest in network science.

Understanding relations encoded in complex networks, however, still represents a challenging task, and statistical methods that can help to summarize and simplify complex networks are needed. In this thesis we show that often one can gain a deep insight of a network by focusing their attention on communities, i.e. on clusters of nodes, and on the relations that exist between them.

We begin by presenting NEAT, a network-based test that allows to assess relations between gene sets in a gene interaction network. NEAT extends traditional gene enrichment analysis tests by incorporating information on interactions between genes and it overcomes some limitations of existing network enrichment analysis approaches.

Then, we propose two extended stochastic blockmodels that allow to infer the relations that exist between communities from relations between pairs of individuals in a social network. We advocate the use of penalized inference to estimate these models, with the aim of deriving a sparse reduced graph between communities. Application of these models to bill cosponsorship networks in the Italian Chamber of Deputies allows us to reconstruct the pattern of collaborations between Italian political parties from 2001 to 2015.

Finally, we propose a novel clustering strategy for sequences of graphs, based on mixtures of generalized linear models. We show that the proposed clustering method not only is capable to retrieve subpopulations of networks within a cross-sectional or longitudinal sequence of networks, but it also allows to directly characterize them by considering each of the components that form the mixture model.

Samenvatting

Ondanks een lange traditie in de studie van grafen en relationele gegevens, werd decennia lang de analyse van complexe netwerken beperkt door de problemen bij het verzamelen van gegevens en computationele lasten. De komst van nieuwe technologieën in de levenswetenschappen, maar ook in het dagelijks leven, is het plotseling mogelijk de vele verbanden in de wereld, van vriendschap en samenwerking tussen individuen of organisaties tot functionele koppelingen tussen de cellulaire moleculen, te analyseren. Dit heeft het verzamelen van relationele gegevens vergemakkelijkt en maakt het concept van "netwerk" van centraal belang in de wetenschap en in sociale en economische praktijken.

Inzicht relaties gecodeerd in complexe netwerken echter nog altijd een uitdaging en statistische methoden die kunnen helpen bij het samenvatten en vereenvoudigen van complexe netwerken zijn urgent nodig. In dit proefschrift laten we zien dat men vaak een diep inzicht van een netwerk kan krijgen door zich te richten op de gemeenschappen, dat wil zeggen op clusters van knooppunten, en op de relaties die er tussen hen bestaan.

We beginnen met de presentatie van NEAT, een test-netwerk op basis die het mogelijk maakt om de relaties tussen genenverzamelingen te evalueren in een gen interactie netwerk. NEAT breidt traditionele gen verrijking analyse uit door het opnemen van informatie over interacties tussen genen. Het overkomt daarbij een aantal beperkingen van de bestaande netwerk verrijking benaderingen.

Dan introduceren we een uitbreiding van stochastische blokmodellen die het mogelijk maken om de relaties die bestaan tussen de gemeenschappen van de betrekkingen tussen paren van individuen in een sociaal netwerk af te leiden. We pleiten voor het gebruik van geregulariseerde statistische inferentie van deze modellen, met het doel het afleiden van een interpreteerbare gereduceerde grafiek tussen gemeenschappen. We passen deze modellen toe om cosponsorship netwerken te beschrijven in de Italiaanse Kamer van Afgevaardigden en om het patroon van de samenwerkingen tussen de Italiaanse politieke partijen te reconstrueren in de periode van 2001-2015.

Tenslotte stellen we een nieuwe strategie voor het clusteren van grafen op basis van een mixture van generaliseerde lineaire modellen.

We tonen aan dat de voorgestelde methode clustering niet alleen in staat stelt om subpopulaties van netwerken te identificeren, maar ook om ieder netwerk individueel te karakteriseren.

Acknowledgements

Before I begin to acknowledge the persons who, directly or indirectly, contributed to my doctoral studies and to this thesis, let me first of all observe that, *as a network scientist*, I am fully aware that undertaking research at a university - or actually, in my case, at *two* universities - is one of those human activities which involve a plentiful of interactions, discussions and collaborations. Thus, it is nearly impossible, for me, to draw here a full picture of the *complex network* that supported me over the last three years, but I will do my best to minimize the amount of missing edges.

I would like to express my gratitude to Prof. Ernst Wit, who has wisely guided me into the challenging and fascinating field of statistical network science and encouraged me to become an independent researcher. I truly enjoyed our fruitful discussions and the time spent working together on such a multidisciplinary research project.

I am highly indebted to Prof. Monica Chiogna for her constant and unconditional support, from the beginning to the end of my PhD. Thanks for giving me confidence during the most critical passages of my PhD, as well as for your frank opinion whenever needed.

I would also like to thank Prof. Veronica Vinciotti, who collaborated to the development of NEAT (Chapter 2), and Dr. Daniele Durante, whose work inspired the development of our method for graph clustering (Chapter 5).

I would like to thank the members of the Assessment Committee, Prof. Alessandra Brazzale, Prof. Matischa De Gunst, Prof. Gianpaolo Scalia Tomba and Prof. Tom Snijders for evaluating this thesis and their useful comments. A special thanks goes to Prof. Alessandra Brazzale, who gently helped me to choose my research project – eventually making this double PhD possible.

I am grateful to the professors who are part of the PhD Course in Statistics at the University of Padova for their efforts to provide a educational programme of high quality. I am also grateful to the members of the Statistics and Probability Unit of the Johann Bernoulli Institute, whom I shared discussions and research interest with for two uninterrupted years.

I would also like to thank the professors, colleagues and staff of the Department of Statistical Sciences of the University of Padova, and of

the Johann Bernoulli Institute for Mathematics and Computer Science of the University of Groningen. The possibility to pursue my doctoral studies at two scientific institutes in different countries has been for me an outstanding opportunity, and I am deeply grateful for it.

A special thanks goes to Patrizia Piacentini, who has always been available to clarify any administrative issue and has always been a source of energy and smile.

Last but not least, I would like to thank my family, which supported me while distant from home and abroad; Filippo, who has been a motivator and an important part of this long journey; Vera, Gianluca and Saverio, whom I shared the most hilarious moments of my PhD with; Xandra, who welcomed me in Groningen, hosted me in my very first days in the Netherlands and has been a great friend since then; and Fernanda, Andreas, Nick, Jing, Michael and Vania, whose friendships have been a source of relief even in the most difficult moments.