ISSN 1688-2784

Universidad de la República
Facultad de Ingeniería

Graph inference and graph matching problems: theory and algorithms

Tesis presentada a la Facultad de Ingeniería de la Universidad de la República por

Marcelo Fiori

en cumplimiento parcial de los requerimientos
para la obtención del título de
Doctor en Ingeniería Eléctrica.

Directores de Tesis

Pablo Musé	Universidad de la República
Guillermo Sapiro	Duke University

Tribunal

Alex Bronstein	Tel Aviv University
Marcelo Lanzilotta	Universidad de la República
Gonzalo Mateos	University of Rochester
Gadiel Seroussi	Universidad de la República

Director Académico

Pablo Musé	Universidad de la República

Montevideo
Monday 11th May, 2015

Graph inference and graph matching problems: theory and algorithms,
Marcelo Fiori.

ISSN 1688-2784

Esta tesis fue preparada en LaTeX usando la clase iietesis (v1.1).
Contiene un total de 124 páginas.
`http://iie.fing.edu.uy/~mfiori`

ACTA DE DEFENSA

TESIS DE DOCTORADO

Fecha: 11 de mayo de 2015 .-

Lugar: Montevideo, Facultad de Ingeniería – Universidad de la República.-

Plan de Estudio: Doctorado en Ingeniería Eléctrica.-

Aspirante: **Marcelo Fiori Schiavo**.-

Documento de Identidad: 3.600.448-2

Director/es de Tesis: Dr. Guillermo Sapiro y Dr. Pablo Musé.-

Tribunal: **Dr. Alex Bronstein (Tel Aviv University, Israel);**
Dr. Marcelo Lanzilotta (IMERL, Fing, UdelaR);
Dr. Gonzalo Mateos (University of Rochester, USA);
Dr. Gadiel Seroussi (IIE/INCO, Fing, UdelaR).-

Los miembros del Tribunal hacen constar que en el día de la fecha el **Sr. Ing. Marcelo Friori** ha sido **APROBADO** en la defensa de su **Tesis de Doctorado** titulada: **"Graph inference and graph matching problems: theory and algorithms"**

La resolución del Tribunal se fundamenta en los puntos detallados a continuación:

El tribunal considera que la tesis presentada por Marcelo Fiori presenta un muy buen balance entre desarrollos teóricos y resolución de problemas prácticos. Las contribuciones teóricas son no triviales, elegantes y novedosas. La parte práctica trata temas de altísima relevancia y contemporaneidad. A la vez, en ambos casos, los problemas son extremadamente complejos en un área madura donde es difícil hacer nuevos aportes de importancia.

Para resolver los temas abordados, se utiliza una combinación elegante y a la vez poderosa de herramientas de teoría clásica de grafos y herramientas más recientes de optimización en alta dimensionalidad y procesamiento de señales en grafos.

La presentación mostró claridad, y buen balance entre la explicación general para la audiencia no especializada y el dominio de los detalles técnicos. Las respuestas a las numerosas preguntas del jurado fueron seguras y contundentes, mostrando una vez más el dominio del tema y sus aptitudes como docente.

Por consiguiente, el tribunal estima que esta tesis es brillante y excepcional, y está convencido que Marcelo Fiori merece el título de Doctor en Ingeniería Eléctrica de la Universidad de la República con las mayores distinciones.

Para que conste,

ABronstein

Dr. Alex Bronstein

Dr. Marcelo Lanzilotta

Dr. Gonzalo Mateos

Gadiel Seroussi

Dr. Gadiel Seroussi

Acknowledgments

First and foremost, I would like to profoundly thank my thesis advisors Pablo and Guillermo, who guided me through this adventure with passion and patience. Pablo, my “local guide” in Uruguay, had a constant will to meet and discuss ideas; he always had positive and encouraging words for me, and he made me feel extremely comfortable working with him on this thesis. I always felt lucky to have Guillermo as my mentor. His work ethic and enthusiasm for science have enlightened the academic aspect of this journey, and on the other hand, his generosity, humility, and kindness have made me feel at home during the whole time I was in the U.S. I will be forever grateful to both of them.

I would like to express my gratitude to Profs. Alex Bronstein, Gonzalo Mateos, Gadiel Seroussi, and Marcelo Lanzilotta, for serving on my committee. They generously accepted the invitation to be a part of the committee, took the time to read this manuscript, and provided me with valuable comments and discussions, which I very much appreciate.

In part, this thesis is the result of the strong collaboration between Guillermo Sapiro and the Engineering School at Universidad de la República, Uruguay. The main ideas were conceived while I was with Guillermo at the University of Minnesota, and since then, I have been working with Guillermo and Pablo in Minneapolis (at University of Minnesota), Durham (at Duke University), and Montevideo. I am very grateful for having had the opportunity to be an active part of the scientific and social life of these cities along this process.

This unsettled adventure has left me with beautiful friends in several parts of the world. I would like to name a few of them in particular, and I apologize beforehand to those who are not explicitly mentioned due to the cruel limitation of space.

The warmth of the Wales House, of Kelly, Julie, Aly and Nicole, and of all the people I met there, was a nice way to compensate the characteristic winter cold in Minneapolis. My memories of the land of 10,000 lakes will always include the wonderful friends with whom I shared so many moments: Gonchi, Nacho, Paty, Juan, Ana, Alexey, Pablo Cancela, Mariano, Jordan, Iman, and especially Pablo Sprechmann.

In that sense, my life in Durham was not very different: with my dear friends Mariano, Luciana, Chupete, Ceci, Sira, Valerie, Fantine, and Pablo, I enjoyed enriching discussions, tasty coffees, and some sports evenings.

The rest of my support team was in beautiful Montevideo. I've had the pleasure of working alongside great people everywhere, and the Engineering School in Uruguay is no exception. From all my colleagues at IIE and IMERL, I want to highlight and express my sincerest thanks to Paola, Laurita, Jorge G., Corez, Fabián, Rocamora, and fefo.

I am beyond grateful to and all my outstanding friends outside academia who endlessly encouraged me. My family, as expected, holds a special place in these acknowledgments. Thank you for your love and support, particularly to my parents and sister, who always had an unwavering and exaggerated belief in me.

I feel enormously happy and lucky to have shared this adventure with my beloved Tefi. Always supportive, always proud.

Gracias.

A mis viejos.

This page was intentionally left blank.

Abstract

Almost every field has some problems related with graphs or networks. From natural examples in physics and mathematics, to applications in medicine and signal processing, graphs are either a very powerful tool, or a very rich object of interest.

In this thesis we address two classes of graph-related problems. First, we focus on graph-inference problems, consisting in the estimation of a graph or network from a dataset. In this part of the manuscript, we modify the existing formulations of the inference problem to incorporate prior topological information of the graph, and to jointly infer several graphs in a collaborative way. We apply these techniques to infer genetic regulation networks, brain connectivity patterns, and economy-related networks. We also present a new problem, which consists of the estimation of mobility patterns from highly asynchronous and incomplete data. We give a first formulation of the problem with its corresponding optimization, and present results for airplane routes and New York taxis mobility patterns.

The second class consists of the so-called graph matching problems. In this type of problems two graphs are given, and the objective is to find the best alignment between them. This problem is of great interest both from an algorithmic and theoretical point of view, besides the very important applications. Its interest and difficulty lie in the combinatorial nature of the problem: the cost of seeking among all the possible permutations grows exponentially with the number of nodes, and hence becomes intractable even for small graphs.

First, we focus on the algorithmic aspect of the graph matching problem. We present two methods based on relaxations of the discrete optimization problem. The first one is inspired in ideas from the sparse modeling community, and the second one is based on a theorem presented in this manuscript. The importance of these methods is illustrated with several applications.

Finally, we address some theoretical aspects about graph matching and other related problems. The main question tackled in the last chapter is the following: when do the graph matching problem and its convex relaxation have the same solution? A probabilistic approach is first given, showing that, asymptotically, the most common convex relaxation fails, while a non-convex relaxation succeeds with probability one if the graphs to be matched are correlated enough, showing a phase-transition type of behavior. On the other hand, a deterministic approach is presented, stating conditions on the eigenvectors and eigenvalues of the adjacency matrix for guaranteeing the correctness of the convex relaxation solution. Other results and conjectures relating the spectrum and symmetry of a graph are presented as well.

This page was intentionally left blank.

Resumen

En prácticamente todos los campos hay problemas relacionados con grafos o redes. Desde los ejemplos más naturales en física y matemática, hasta aplicaciones en medicina y procesamiento de señales, los grafos son una herramienta muy poderosa, o un objeto de estudio muy rico e interesante.

En esta tesis atacamos dos clases de problemas relacionados con grafos. Primero, nos enfocamos en problemas de inferencia de grafos, que consisten en estimar un grafo o red a partir de cierto conjunto de datos. En esta parte del manuscrito, modificamos las formulaciones existentes de inferencia de grafos para incorporar información topológica previamente conocida sobre el grafo, y para inferir de manera conjunta varios grafos, en un modo colaborativo. Aplicamos estas técnicas para inferir redes de regulación genética, patrones de conectividad cerebral, y redes relacionadas con el mercado accionario. También presentamos un nuevo problema, que consiste en la estimación de patrones de movimiento a partir de un conjunto de datos incompleto, y altamente asíncrono. Mostramos primero una formulación del problema con su correspondiente optimización, y presentamos resultados para rutas de aviones en Estados Unidos, y patrones de movilidad de taxis en New York.

La segunda clase consiste en los llamados *graph matching problems* (problemas de apareamiento de grafos). En este tipo de problemas, dos grafos son dados, y el objetivo es encontrar el mejor alineamiento entre ellos. Este problema es de gran interés tanto desde un punto de vista algorítmico como teórico, además de las importantes aplicaciones que tiene. El interés y la dificultad de este problema tienen raíz en la naturaleza combinatoria del mismo: el costo de buscar entre todas las permutaciones posibles crece exponencialmente con el número de nodos, y por lo tanto se vuelve rápidamente intratable, incluso para grafos chicos.

Primero, nos enfocamos en el aspecto algorítmico del problema de *graph matching*. Presentamos dos métodos basados en relajaciones del problema de optimización discreta. El primero de ellos está inspirado en ideas de la comunidad de *sparse modeling*, y el segundo está basado en un teorema presentado en este manuscritp. La importancia de estos métodos es ilustrada con varias aplicaciones a lo largo del capítulo.

Finalmente, atacamos algunos aspectos teóricos sobre *graph matching* y otros problemas relacionados. La pregunta principal que se encara en el último capítulo es la siguiente: ¿cuándo el problema de *graph matching* y su relajación convexa tienen la misma solución? Primero damos un enfoque probabilístico mostrando que, asintoticamente, la relajación convexa más común falla, mientras que una relajación no convexa es capaz de resolver el problema con probabilidad uno, siempre y cuando los grafos originales estén lo suficientemente correlacionados, mostrando un comportamiento del estilo de transición de fases. Por otro lado, un enfoque determinístico es también presentado, estableciendo condiciones sobre los valores y vectores propios de las matrices de adjacencia de los grafos, que garantizan que el problema de *graph matching* y su relajación convexa tienen la misma solución. Otros resultados y conjeturas relacionando el espectro y la simetría de un grafo son presentados también en este capítulo.

Contents

This page was intentionally left blank.

Chapter 1

Introduction

The terms *graph* and *network* are of daily use nowadays, mainly because of the massive (and maybe abusive) use of social networks, but problems related with graphs have been an enjoyable challenge for the scientific community for centuries. While some of the oldest questions remain unsolved, as technology advances and the application of graph-related techniques grows, new problems and challenges arise. In this thesis we address some of these new problems, as well as some old classic questions about graphs and their spectrum.

Informally, a graph is a set of points connected by lines, therefore, it is not surprising that an enormous amount of real world problems can be modeled by graphs. Let us enumerate some of these applications to illustrate the strength of this tool, starting with real world problems resembling three of the classic puzzles in graph theory.

The Seven Bridges of Königsberg is a problem solved in 1735 by Leonhard Euler, in what is said to be the first graph theory paper. The Pregel River crossed the city of Königsberg, dividing it in four land regions, two of which were large islands (see Figure 1.1). These regions were connected by seven bridges, and the problem solved by Euler was to find a path through the city crossing each bridge exactly once. Euler proved the impossibility of such walk, associating a node to each land region, connecting them according to the existent bridges, and then developing the basis of graph theory. The same concepts arise for planning or analyzing a transportation system for a city. The metro and bus designs can be thought as a graph (and actually they are displayed as graphs in every map), for which certain constraints should be satisfied: for instance, the average trip time cannot be too large, if one segment breaks there should be alternative paths connecting every pair of stations, and all this must be done with limited resources: we cannot connect every station with each other.

There is another famous “impossible” puzzle: connect three houses with the water, energy and gas supply stations, without any pipe crossing another one. This is a planar puzzle (two lines cannot cross each other in the drawing), and of course we can use the third dimension to solve the problem in the real world, but the design and modification of (much larger and complex) graphs of energy companies poses a tremendous challenge, again because of limited resources and

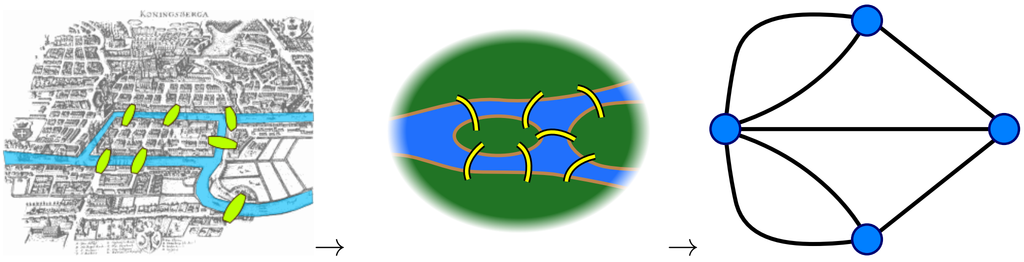
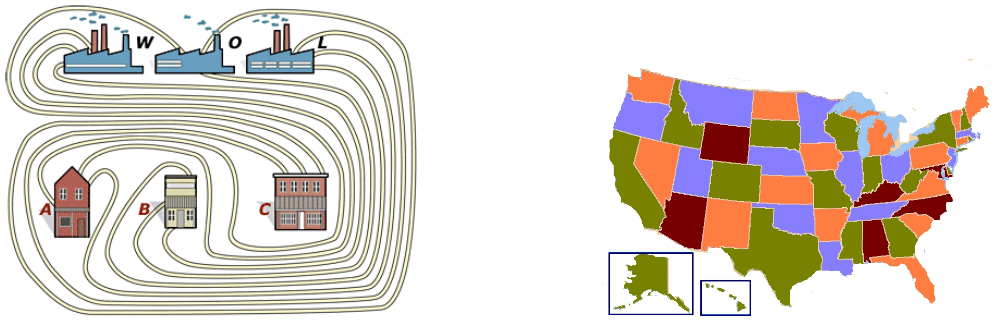


Figure 1.1: The Seven Bridges of Königsberg problem. Images from Wikipedia.

very complex constraints (see Figure 1.2a). Nevertheless, maybe the most faithful example of an application of this puzzle is the printing of electronic circuits, where, of course, two lines cannot cross each other.

The last classic problem we mention here is the graph coloring problem. In 1852, Francis Guthrie stated the four color conjecture, which says that four colors are sufficient to color a map so that no adjacent regions are painted with the same color. This conjecture remained unsolved until 1976, when Appel and Haken proved it.[1] Besides the obvious application of coloring maps (see Figure 1.2b), this area also helps to solve scheduling, pattern matching, compiler register allocation, and frequency allocation problems, to name a few.



(a)

(b)

Figure 1.2: An incorrect solution (why?) of the house supply puzzle (image taken from `www.archimedes-lab.org`), and coloring of the US map using the four color theorem (image from Wikipedia).

We already mentioned social networks, where each node represents a person, and one edge between two people represents a connection in the social network (typically meaning that these people know each other). Here, graph problems appear everywhere and constantly, from recommendation of people to complex searches in the network.

Maybe the largest and most complex network is the Internet itself, but every communication network involves a number of graph problems, which have been pushing and extending the limits of the field for several years now.

[1] This theorem is also famous for being the first major theorem proved with the aid of a computer.

How does a certain population move within a metro station? How are the different regions of the brain connected one to each other? What is the dependency of the socio-economic aspect of a country with respect to others? All these applications can be cast into a graph problem. We will address some of these aspects along this manuscript, from *functional magnetic resonance imaging* (fMRI) data of the brain activity to infer the significant connections in the brain, to the estimation of mobility patterns of subjects using only local and static information.

The beauty of the underlying mathematics justifies by itself the interest in graph problems, and all these applications (and those we do not mention here) complement the unneeded justification. Now, let us formalize the main concepts and analyze where the difficulty in these graph problems lies.

An undirected graph $G = (V, E)$ consists of a vertex set V and an edge set E, where E contains unordered pairs of vertices. All the information of the graph can be conveniently and compactly expressed as the so-called adjacency matrix. If the graph G has n vertices, then its adjacency matrix $\mathbf{A}$ is an $n \times n$ matrix,[2] such that $\mathbf{A}(i, j) = 1$ if and only if there is an edge joining the nodes i and j, and $\mathbf{A}(i, j) = 0$ otherwise. This matrix representation allow us to use all the powerful tools from linear algebra to deduce properties of the graph.

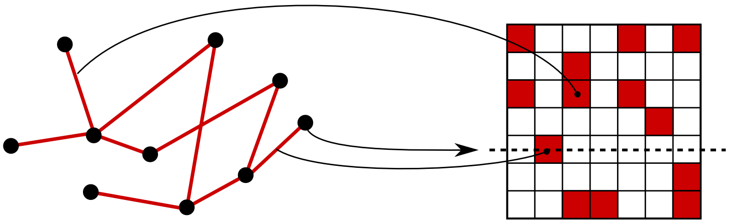


Figure 1.3: Graph adjacency matrix.

If instead of unordered pairs, the edges set E is formed by ordered pairs, we say that the graph is *directed*, since each link has an origin and destination node. In this case, the corresponding adjacency matrix is not necessarily symmetric.

On the other hand, both for directed and undirected graphs, we can add weights to each edge, which corresponds to adjacency matrices with real entries instead of binary ones.

All these concepts and classes of graphs are used along this manuscript, addressing problems that we can divide in two large groups. The first type of problems consists on estimating a graph of particular interest to explain or analyze some given data. In the second class of problems, the graph or graphs are given, and the goal is to decide something about the graph, or about the relationship between the graphs.

There are several ways to infer a graph from a dataset, and in general the most appropriate one depends on the nature of the data, on the problem, and on the objective of the estimation. Let us assume that we have a data matrix $\mathbf{X}$ of size

[2] Throughout this manuscript, we refer to matrices in uppercase bold font (e.g., $\mathbf{A}$), vectors in lowercase bold font (e.g., $\mathbf{v}$). The matrix/vector entries are denoted in lowercase (e.g., a_{ij}, v_i).

$t \times n$, where each column $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ corresponds to a certain entity, and each row corresponds to a certain measurement of the system. The goal is to detect the dependence relations between the entities. Very often, the problem with these estimations is that the number of measurements is much smaller than the dimension of the problem [60]: i.e. $t << n$, which makes the estimation problem ill-posed in general.[3] If $\mathbf{X}$ is a multivariate normal distributed variable, then two coordinates $\mathbf{X}_i$ and $\mathbf{X}_j$ are conditionally independent given all the other coordinates if and only if the (i, j) entry of the inverse covariance matrix $\mathbf{\Sigma}^{-1}$ is zero. This property makes the inference of the inverse covariance matrix (and moreover the structure of the non-zero values) of enormous importance. Nevertheless, the inference of this structure is also of great interest for several distributions of $\mathbf{X}$, even for discrete ones [66]. When the amount of data is small in comparison with the number of dimensions the problem is ill-posed, and therefore it is reasonable to add some prior information in order to stabilize the solution. In this manuscript, we assume that the number of non-zero entries of the inverse covariance matrix is small in comparison with the total number of entries, i.e., the matrix is sparse. This is a very reasonable assumption for a large class of problems, and during the last decade there has been an outstanding progress both in theoretical guarantees and methods to solve optimization problems with sparsity regularizers.

In this thesis, we address the graph inference problem following this methodology in Chapter 2. First, we add topological information to the estimation problem, and illustrate the applicability of the resulting algorithms using stock market data and genetic regulatory networks. Then, we adapt an idea from the sparse modeling community to jointly estimate two or more graphs, which we assume have similar structure. This is true, for instance, for brain connectivity graphs of certain population, which is the application illustrated in the corresponding section.

The last graph inference problem tackled in this manuscript is somehow different, mainly due to the dynamic aspect. Indeed, a simple example to illustrate the problem is the following: let us assume that we know how many airplanes are in each airport at each time interval. The available information has a time component, since we know the number of airplanes for several time intervals, but no explicit information about the dynamics is available. The goal is then to infer the routes of the airplanes. There are several problems falling into this category, including for example the estimation of the mobility pattern of people from counting data at specific places.

Back to the airplanes example, since each route has different average flight durations, and when an airplane is flying we cannot see it in the available data, the timing component of this model is very different from the previous formulation, and makes the problem very challenging. Moreover, since there may be several combination of routes that could explain the data, the problem is generally severely ill-posed. To finish the second chapter of this thesis, we present a model and an algorithm to infer the mobility pattern of entities from counting data, with applications to real world scenarios.

[3] In statistics, this is known as the $n << p$ problem, due to the standard notation in the field.

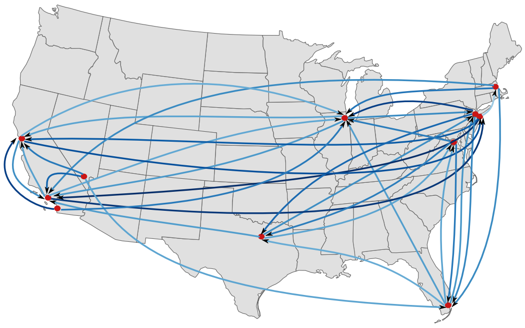


Figure 1.4: Some domestic routes between 11 US airports.

Let us briefly introduce now some of the problems of the second category, where instead of inferring a network, the graphs are given.

From its formal definition, a graph is a set of vertices and edges, but of course the picture of the graph is a much direct way to visualize the graph. However, we can represent a graph in many different ways, with different node positions and order (see Figure 1.5). Therefore, how can we recognize a graph from its picture? Given two different pictures, can we decide whether they represent the same graph or not?

The formal formulations of this problem are the so-called Graph Matching Problem and Graph Isomorphism Problem, which we briefly describe in what follows.

An isomorphism is a bijection between the vertex sets of the graphs, preserving the edge structure, i.e., a re-ordering of the nodes.

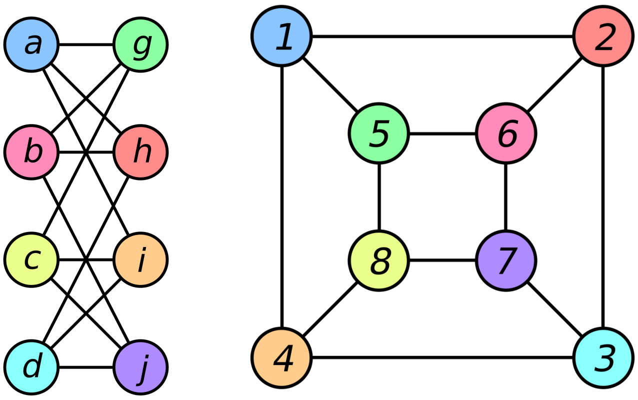


Figure 1.5: Two isomorphic graphs. Image from Wikipedia.

The Graph Isomorphism Problem consists in deciding whether two given graphs are isomorphic or not (i.e., if there exists an isomorphism between them). The computational complexity aspect of this problem is very particular, since its complexity class is still unsolved: it is one of the few problems in NP not yet classified as P nor NP-complete [25].

In terms of the adjacency matrices, since an isomorphism is simply a bijection between the vertex sets, it can be thought of as a permutation matrix. Indeed, two graphs with adjacency matrices $\mathbf{A}$ and $\mathbf{B}$ are isomorphic if and only if there exists a permutation matrix P such that $\mathbf{A} = \mathbf{PBP}^T$. The number of permutation matrices of a given size is finite, and therefore one could try all the possibilities to decide whether two graphs are isomorphic or not. However, the number of permutation matrices grows exponentially with the number of nodes, which makes this naïve approach computationally intractable very fast; in fact, even for a moderate number of nodes, like a dozen, the computational cost is enormous.

Another related problem is the Graph Matching Problem, which consists in finding the "best" bijection between the vertex sets, given certain criteria. If the graphs are isomorphic, the goal is to find the isomorphism. Remember that in the Graph Isomorphism Problem, one has to decide whether two graphs are isomorphic or not, a *yes or no* question, while in this Graph Matching Problem the objective is to find the isomorphism itself, and hence it is at least as hard as the Graph Isomorphism Problem.

Since the problem of finding a permutation matrix $\mathbf{P}$ such that $\mathbf{A} = \mathbf{PBP^T}$, or equivalently, minimizing $\|\mathbf{AP} - \mathbf{PB}\|_F^2$ over the set of permutation matrices, is computationally intractable, relaxation techniques are often used to find an approximate solution. Two very important questions arise immediately: (i) Which is the "best" relaxation of the problem? and (ii) When does it give the correct answer? These two questions set the roadmap of the two final chapters of this manuscript.

In Chapter 3 we focus on the algorithmic aspect. First, we borrow some ideas from sparse modeling to define a new objective function, and after the description of the corresponding optimization, we test the algorithm and present results for several databases. The basic idea behind the classic minimization of $\|\mathbf{AP} - \mathbf{PB}\|_F^2$ is to match globally $\mathbf{AP}$ and $\mathbf{PB}$. The obtained $\mathbf{P}$ may result in a good matching for $\mathbf{A}$ and $\mathbf{PBP^T}$, which is the ultimate goal. The main difference of this sparsity based approach with respect to the classic minimization of $\|\mathbf{AP} - \mathbf{PB}\|_F^2$, is that we aim to match the supports of $\mathbf{AP}$ and $\mathbf{PB}$ (i.e., the non-zero structure), which represents better the edge structure of the graph. The results suggest that this approach tends to outperform the classic graph matching algorithms especially when the graphs are weighted, or multimodal.

The second algorithmic section is strongly related to some theoretical analyses presented in the Chapter 4. We discuss convex and non-convex relaxations of the graph matching problem, with a deep experimental analysis of several techniques, not only for undirected graphs, but also for directed and weighted graphs, as well as graph matching with features and seeds.

This algorithmic chapter leaves us with a battery of methods for graph matching problems, but also with the open question: do we have guarantees for the success of these methods? For this reason, the fourth chapter of this manuscript is dedicated to investigate under which circumstances the solution to the relaxations coincides with the solution of the original graph matching problem.

In the first half of Chapter 4, we study the problem form a probabilistic point of view, assuming a Bernoulli model for the graphs, which is the most general edge independent random graph model. The main result from this analysis has two components: a pessimistic result for the classical graph matching relaxation, and an optimistic result for a non-convex relaxation, showing a phase-transition type of behavior.

Then, in the second half of the chapter, we tackle the problem from a deterministic point of view. It is well known that spectral analysis is fundamental in graph theory, although often this analysis is focused in the spectrum of the Laplacian matrix of the graph. In a recent paper [1], the authors prove the equivalence of the graph matching problem with its classical relaxation for a certain class of graphs, defined by the spectrum of the adjacency matrix. In this manuscript, we extend these results, also deducing important properties of the graphs from the spectral analysis of the adjacency matrix, and providing tools for further understanding the deep relationship between graph spectrum and symmetry.

Outline of the manuscript

The rest of the document is organized as follows:

Chapter 2: Presents three problems related with graph inference. First, the incorporation of topological information to the graph inference problem. Second, the joint and collaborative inference of multiple graphs. And finally, the estimation of mobility patterns from counting data.

Chapter 3: Is devoted to the derivation of graph matching algorithms. First, a new formulation based on sparse modeling techniques is introduced, which tends to outperform classic graph matching methods for weighted and multimodal graphs. Then, an exhaustive experimental analysis of several graph matching algorithms is presented, and in collaboration with some theoretical results of Chapter 4, we suggest new techniques for graph matching problems.

Chapter 4: Tackles the question of when certain graph matching relaxations give the correct answer. In the first place, a probabilistic approach is described, with pessimistic and optimistic results for different convex and non-convex relaxations. Finally, a deterministic analysis is presented, showing conditions on the spectrum of the adjacency matrix of a graph that are sufficient for the equivalence of the graph matching problem and its relaxation.

The manuscript ends with conclusions and bibliography.

Publications

During the time this thesis was written, the following articles were published. Some of them are not included in this manuscript.

- Marcelo Fiori and Guillermo Sapiro, *On spectral properties for graph matching and graph isomorphism problems.* Information and Inference, vol. 4, no. 1, pp. 63–76, 2015; doi: 10.1093/imaiai/iav002.

- Vince Lyzinski, Donniell Fishkind, Marcelo Fiori, Joshua T. Vogelstein, Carey E. Priebe, and Guillermo Sapiro, *Graph Matching: Relax at Your Own Risk*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015.

- Marcelo Fiori, Pablo Musé, Guillermo Sapiro, *A Complete System for Candidate Polyps Detection in Virtual Colonoscopy.* International Journal of Pattern Recognition and Artificial Intelligence Vol. 28, No. 07, 2014.

- Matías Di Martino, Guzman Hernández, Marcelo Fiori, Alicia Fernández, *A new framework for optimal classifier design.* Pattern Recognition 46 (8), pp. 2249-2255, 2013.

- Kimberly Carpenter, Pablo Sprechmann, Marcelo Fiori, Robert Calderbank, Helen Egger, Guillermo Sapiro, *Questionnaire simplification for fast risk analysis of children's mental health.* ICASSP 2014.

- Marcelo Fiori, Pablo Sprechmann, Joshua Vogelstein, Pablo Musé, Guillermo Sapiro, *Robust Multimodal Graph Matching: Sparse Coding Meets Graph Matching.* Advances in Neural Information Processing Systems 26 (NIPS 2013).

- Marcelo Fiori, Pablo Musé, Guillermo Sapiro, *Polyps Flagging in Virtual Colonoscopy.* CIARP 2013.

- Marcelo Fiori, Pablo Musé, Ahmad Hariri, Guillermo Sapiro, *Multimodal Graphical Models via Group Lasso.* Signal Processing with Adaptive Sparse Structured Representations (SPARS), 2013.

- Marcelo Fiori, Pablo Musé, Guillermo Sapiro, *Topology Constraints in Graphical Models.* Advances in Neural Information Processing Systems 25 (NIPS 2012), pp. 800-808.

Chapter 2

Graph Inference

2.1 Introduction

Among all the network related problems presented in the previous chapter, here we focus on graph inference problems. We work with different problem formulations, different data modalities, and different algorithms.

The first two sections have something in common, which is the type of data. Indeed, in these sections we assume that we have a data matrix $\mathbf{X}$ which comes from a multivariate distribution, for instance, a Gaussian multivariate distribution. Each of the n columns $\mathbf{X_i} \in \mathbb{R}^m$ of $\mathbf{X}$ corresponds to a coordinate of the random variable, and each row to an observation. In a general setting, the objective is to infer a graph which illustrates the dependencies between the coordinates of the random variable. It is not rare to have fewer observations than the problem dimension, either because the number of coordinates is extremely large, or the cost of measuring each observation is too high. This situation leads to an extremely ill-posed problem, unless some other prior information is added. A common and reasonable assumption for a large class of real problems is the sparsity of the underlying graph: only a small portion of all the $n(n-1)$ possible edges is active. This prior information can be incorporated to the inference formulations by means of ℓ_1 penalty terms, as successfully done in several fields [72,81,89,97]. Besides this sparsity constraint, in Section 2.2 we incorporate prior topological information to the graph inference problem, and in Section 2.3 we present a formulation to jointly estimate several graphs, which are assumed to have a common structure.

The problem presented in Section 2.4, on the other hand, has never been introduced to the best of our knowledge. Although it is still a graph inference problem, the type of data is completely different. The timing aspect is fundamental, and for the applications addressed in this final section, the asynchronicity component makes the graph inference problem a very challenging task.

2.1.1 Sparse modeling background

Let us first describe some basic concepts of sparse modeling, which are present along several sections in this manuscript.

Assume we have a set of n data points $\mathbf{x_i} \in \mathbb{R}^m$, and a linear model represented in the $m \times p$ matrix $\mathbf{D}$, such that each data point $\mathbf{x_i}$ can be written as $\mathbf{x_i} = \mathbf{D}\mathbf{a_i} + \boldsymbol{\varepsilon}$, $\mathbf{a_i} \in \mathbb{R}^p$, $\boldsymbol{\varepsilon} \in \mathbb{R}^m$. That is, each $\mathbf{x_i}$ can be expressed as a linear combination of the columns of $\mathbf{D}$, plus some (small) noise. In the sparse modeling community, this matrix $\mathbf{D}$ is usually called a *dictionary*, and each column of $\mathbf{D}$ is called an *atom*. The underlying assumption in sparse modeling theory is that only a few of these atoms are enough for representing most of the $\mathbf{x_i}$. This can be expressed formally by means of the ℓ_0 pseudo-norm, which counts the number of non-zero elements of a vector $\mathbf{v} = (v_1, v_2, \ldots, v_p) \in \mathbb{R}^p$: $\|\mathbf{v}\|_0 = \#\{j : v_j \neq 0\}$. Then, if one wants to find the sparsest solution to $\mathbf{Da} = \mathbf{x_i}$, the corresponding formulation would be:

$$\min_{\mathbf{a}\in\mathbb{R}^p} \|\mathbf{a}\|_0 \quad s.t. \quad \mathbf{Da} = \mathbf{x_i},$$

or more generally, allowing some reconstruction error upper bounded by ϵ,

$$\min_{\mathbf{a}\in\mathbb{R}^p} \|\mathbf{a}\|_0 \quad s.t. \quad \|\mathbf{Da} - \mathbf{x_i}\|_2^2 < \epsilon.$$

The equivalent unconstrained formulation to this last problem (meaning that for each ϵ there exist a λ such that the solutions of the two problmes coincide) is

$$\min_{\mathbf{a}\in\mathbb{R}^p} \|\mathbf{Da} - \mathbf{x_i}\|_2^2 + \lambda\|\mathbf{a}\|_0.$$

However, the combinatorial nature of this problem, due to the ℓ_0 pseudo-norm, makes this an NP-hard problem. Therefore, a common way to deal with this problem is to relax the ℓ_0 pseudo-norm to its closest ℓ^p convex norm, which is the ℓ_1 norm. The relaxed optimization problem becomes

$$\min_{\mathbf{a}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{Da} - \mathbf{x_i}\|_2^2 + \lambda\|\mathbf{a}\|_1,$$

which is a convex problem (referred as Lasso [87] or Basis Pursuit [32] in the community), and can be solved efficiently by modern optimization techniques, designed for these non-differentiable optimization problems [13, 98].

The relationship between the original ℓ_0 constrained problem and its relaxation has been deeply studied during the last decade, and under some conditions the solutions of the two problems coincide [20, 33]. The intuition behind this equivalence can be nicely illustrated with the following formulation:

$$\min_{\mathbf{a}\in\mathbb{R}^p} \|\mathbf{Da} - \mathbf{x_i}\|_2^2 \quad s.t. \quad \|\mathbf{a}\|_1 \leq \mu, \tag{2.1}$$

whose geometric representation is shown in Figure 2.1. Note how the ℓ_1 constraint promotes sparsity, unlike the ℓ_2 constraint.

In some cases, it makes sense to impose that some variables have to be active at the same time. This can be achieved by means of the Group Lasso [100]. Given a partition $\mathcal{G}$ of $\{1, 2, \ldots, p\}$, the Group Lasso problem is:

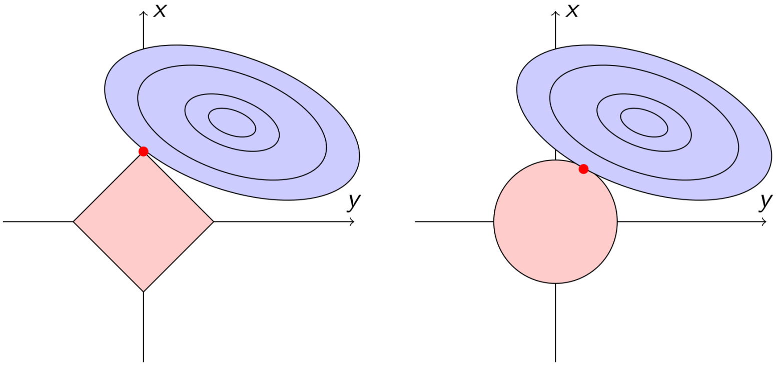


Figure 2.1: The ℓ_1 norm promotes sparsity: geometric representation of problem (2.1). Level lines of $\|\mathbf{Da}-\mathbf{x}\|_2$, intersected with the balls satisfying the constraints. Left: problem with ℓ_1 norm constraint, where the solution is achieved at a vertex, therefore achieving sparsity. Right: non-sparse solution of problem with ℓ_2 norm constraint.

$$\min_{\mathbf{a}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{Da}-\mathbf{x_i}\|_2^2 + \lambda \sum_{G\in\mathcal{G}} \|\mathbf{a_{[G]}}\|_2,$$

where $\mathbf{a_{[G]}}$ stands for the coefficients of $\mathbf{a}$ corresponding to group G.

This is a generalization of the ℓ_1 minimization problem, in the sense that when the partition $\mathcal{G}$ consists of each coordinate as a singleton, then the Group Lasso formulation reduces to the classical Lasso. The effect on the sparsity is also a generalization: only a portion of the groups are active.

Both the Lasso and Group Lasso are used along this and the following chapters, applied first to graph inference problems, and then to graph matching problems.

2.2 Topology Constraints for Graph Inference

Section summary

Graphical models are a very useful tool to describe and understand natural phenomena, from gene expression to climate change and social interactions. The topological structure of these graphs/networks is a fundamental part of the analysis, and in many cases the main goal of the study. However, little work has been done on incorporating prior topological knowledge onto the estimation of the underlying graphical models from sample data. In this section we propose extensions to the basic joint regression model for network estimation, which explicitly incorporate graph-topological constraints into the corresponding optimization approach. The first proposed extension includes an eigenvector centrality constraint, thereby promoting this important prior topological property. The second developed extension promotes the formation of certain motifs, triangle-shaped ones in particular, which are known to exist for example in genetic regulatory networks. The presentation of the underlying formulations, which serve as examples of the introduction of topological constraints in network estimation, is complemented with examples in diverse datasets demonstrating the importance of incorporating such critical prior knowledge.

2.2.1 Problem description

The estimation of the inverse of the covariance matrix (also referred to as *precision matrix* or *concentration matrix*) is a very important problem with applications in a number of fields, from biology to social sciences, and is a fundamental step in the estimation of underlying data networks. The *covariance selection* problem, as introduced by [30], consists in identifying the zero pattern of the precision matrix. Let $X = (X_1 \dots X_p)$ be a p-dimensional multivariate normal distributed vector, $X \sim \mathcal{N}(0, \mathbf{\Sigma})$, and $\mathbf{C} = \mathbf{\Sigma}^{-1}$ its concentration matrix. Then two coordinates X_i and X_j are conditionally independent given the other variables if and only if $\mathbf{C}(i,j) = 0$ [63]. This property motivates the representation of the conditional dependency structure in terms of a graphical model $G = (V, E)$, where the set of nodes V corresponds to the p coordinates and the edges E represent conditional dependency. Note that the zero pattern of the G adjacency matrix coincides with the zero pattern of the concentration matrix. Therefore, the estimation of this graph G from k random samples of X is equivalent to the covariance selection problem. The estimation of G using ℓ_1 (sparsity promoting) optimization techniques has become very popular in recent years.

This estimation problem becomes particularly interesting and hard at the same time when the number of samples k is smaller than p. Several real life applications lie in this "small k-large p" setting. One of the most studied examples, and indeed with great impact, is the inference of genetic regulatory networks (GRN) from DNA microarray data, where typically the number p of genes is much larger than the number k of experiments. Like in the vast majority of applications, these networks have some very well known topological properties, such as sparsity (each node is connected with only a few other nodes), scale-free behavior, and the presence of hubs (nodes connected with many other vertices). All these properties are shared with many other real life networks like Internet, citation networks, and social networks [74].

Genetic regulatory networks also contain a small set of recurring patterns called *motifs*. The systematic presence of these motifs was first discovered in *Escherichia coli* [84], where it was found that the frequency of these patterns is much higher than in random networks, and since then they have been identified in other organisms, from bacteria to yeast, plants and animals.

The topological analysis of networks is fundamental, and often the essence of the study. For example, the proper identification of hubs or motifs in GRN is crucial. Thus, the agreement of the reconstructed topology with the original or expected one is critical. Sparsity has been successfully exploited via ℓ_1 penalization in order to obtain consistent estimators of the precision matrix, but little work has been done with other graph-topological properties, often resulting in the estimation of networks that lack critical known topological structures, and therefore do not look natural. Incorporating such topological knowledge in network estimation is the main goal of this work.

Eigenvector centrality (see Section 2.2.3 for the precise definition) is a well-known measure of the importance and the connectivity of each node, and typical centrality distributions are known (or can be estimated) for several types of networks. Therefore, we first propose to incorporate this structural information into the optimization procedure for network estimation in order to control the topology of the resulting network. This centrality constraint is useful when some prior information about the graphical model is known, for example, in dynamic networks, where the topology information of the past can be used; in networks which we know are similar to other previously studied graphs; or in networks that model a physical phenomenon for which a certain structure is expected.

As mentioned, it has been observed that genetic regulatory networks are composed by a few geometric patterns, repeated several times. One of these motifs is the so-called *feedforward loop*, which is manifested as a triangle in the graph. Although it is thought that these important motifs may help to understand more complex organisms, no effort has been made to include this prior information in the network estimation problem. As a second example of the introduction of topological constraints, we propose a simple modification to the ℓ_1 penalty, weighting the edges according to their local structure, in order to favor the appearance of these motifs in the estimated network.

Both developed extensions here presented are very flexible, and they can be combined with each other or with other extensions reported in literature.

To recapitulate, we propose several contributions to the network estimation problem: we show the importance of adding topological constraints; we propose an extension to ℓ_1 models in order to impose the eigenvector centrality; we show how to transfer topology from one graph to another; we show that even with the centrality estimated from the same data, the proposed extension outperforms the basic model; we present a weighting modification to the ℓ_1 penalty favoring the appearance of motifs; as illustrative examples, we show how the proposed framework improves the edge and motif detection in the *E. coli* network, and how the approach is important as well in financial applications.

The rest of this section is organized as follows. In Section 2.2.2 we describe the basic precision matrix estimation models used in this chapter. In Section 2.2.3 we introduce the eigenvector centrality and describe how to impose it in graph estimation. We propose the weighting method for motifs estimation in Section 2.2.5. Experimental results are presented in Section 2.2.6, and we conclude in Section 2.2.8.

2.2.2 Graphical model estimation

Let $\mathbf{X}$ be a $k \times p$ matrix containing k independent observations of X, and let us denote by $\mathbf{X}_i$ the i-th column of $\mathbf{X}$. Two main families of approaches use sparsity constraints when inferring the structure of the precision matrix. The first one is based on the fact that the (i, j) element of $\mathbf{\Sigma}^{-1}$ is, up to a constant, the regression coefficient β^i_j in $\mathbf{X}_i = \sum_{l \neq i} \beta^i_l \mathbf{X}_l + \varepsilon_i$, where ε_i is uncorrelated with $\{\mathbf{X}_l | l \neq i\}$.

Following this property, the neighborhood selection technique by [73] consists in solving p independent ℓ_1 regularized problems [87],

$$\arg \min_{\boldsymbol{\beta}^i : \beta^i_i = 0} \frac{1}{k} ||\mathbf{X}_i - \mathbf{X}\boldsymbol{\beta}^i||^2 + \lambda ||\boldsymbol{\beta}^i||_1 \,,$$

where $\boldsymbol{\beta}^i$ is the vector of β^i_js. While this is an asymptotically consistent estimator of the $\boldsymbol{\Sigma}^{-1}$ zero pattern, β^i_j and β^j_i are not necessarily equal since they are estimated independently. A joint regression model which guarantees symmetry is proposed in [80]. This regression of the form $\mathbf{X} \approx \mathbf{XB}$, with $\mathbf{B}$ sparse, symmetric, and with null diagonal, allows to control the topology of the graph defined by the non-zero pattern of $\mathbf{B}$, as it will be later exploited in this work. A symmetric version of the model by [73] is also solved in [48], where the authors incorporate some structural penalties as the group Lasso by [100].

Methods of the second family are based on a maximum likelihood (ML) estimator with an ℓ_1 penalty [9,47,101]. Specifically, if $\mathbf{S}$ denotes the empirical covariance matrix, the solution is the matrix $\boldsymbol{\Theta}$ which solves the optimization problem

$$\max_{\boldsymbol{\Theta} \succ 0} \; \log\det\boldsymbol{\Theta} - \operatorname{tr}(\mathbf{S}\boldsymbol{\Theta}) - \lambda \sum_{i,j} |\boldsymbol{\Theta}_{ij}| \;.$$

An example of an extension to both models (the regression and ML approaches), and the first to explicitly consider additional classical network properties, is the work by [65], which modifies the ℓ_1 penalty to derive a non-convex optimization problem that favors scale-free networks.

A completely different technique for network estimation is the use of the PC-Algorithm to infer acyclic graphs [58]. This method starts from a complete graph and recursively deletes edges according to conditional independence decisions. In this work, we use this technique to estimate the graph eigenvector centrality.

2.2.3 Eigenvector centrality model extension

Node degree (the number of connections of a node) is the simplest algebraic property than can be defined over a graph, but it is very local as it only takes into account the neighborhood of the node. A more global measure of the node importance is the so-called *centrality*, in any of its different variants [74]. In this work, we consider the *eigenvector centrality*, defined as the dominant eigenvector (the one corresponding to the largest eigenvalue) of the corresponding network connectivity matrix. The coordinates of this vector (which are all non-negative by the Perron-Frobenius theorem) indicate the corresponding centrality of each node, and provide a measure of the influence of the node in the network (Google's *PageRank* is a variant of this centrality measure). Distributions of the eigenvector centrality values are well known for a number of graphs, including scale-free networks as the Internet and GRN [74].

In certain situations, we may have at our disposal an estimate of the centrality vector of the network to infer. This may happen, for instance, because we already had preliminary data, or we know a network expected to be similar, or simply someone provided us with some partial information about the graph structure. In those cases, we would like to make use of this important side information, both to improve the overall network estimation and to guarantee that the inferred graph is consistent with our prior topological knowledge. In what follows we propose an extension of the joint regression model which is capable of controlling this topological property of the estimated graph.

To begin with, let us remark that as $\mathbf{\Sigma}$ is positive-semidefinite and symmetric, all its eigenvalues are non-negative, and thus so are the eigenvalues of $\mathbf{\Sigma}^{-1}$. By virtue of the Perron-Frobenius Theorem, for any adjacency matrix $\mathbf{A}$, the eigenvalue with largest absolute value is positive. Therefore for precision and graph connectivity matrices it holds that $\max_{||\mathbf{v}||=1} |\langle \mathbf{Av}, \mathbf{v}\rangle| = \max_{||\mathbf{v}||=1}\langle \mathbf{Av}, \mathbf{v}\rangle$, and moreover, the eigenvector centrality is $\mathbf{c} = \arg\max_{||\mathbf{v}||=1}\langle \mathbf{Av}, \mathbf{v}\rangle$.

Suppose that we know an estimate of the centrality $\mathbf{c} \in \mathbb{R}^p$, and want the inferred network to have centrality close to it. We start from the basic joint regression model,

$$\min_{\mathbf{B}} ||\mathbf{X} - \mathbf{XB}||_F^2 + \lambda_1 ||\mathbf{B}||_1 \ , \qquad s.t.\ \mathbf{B} \text{ symmetric, } \mathbf{B}_{ii} = 0 \ \forall i, \tag{2.2}$$

and add the centrality penalty,

$$\min_{\mathbf{B}} ||\mathbf{X}{-}\mathbf{XB}||_F^2{+}\lambda_1 ||\mathbf{B}||_1{-}\lambda_2 \langle \mathbf{Bc}, \mathbf{c}\rangle \ , \qquad s.t.\ \mathbf{B} \text{ symmetric, } \mathbf{B}_{ii} = 0 \ \forall i \tag{2.3}$$

where $||\cdot||_F$ is the Frobenius norm and $||\mathbf{B}||_1 = \sum_{i,j} |\mathbf{B}_{ij}|$. The minus sign is due to the minimization instead of maximization, and since the term $\langle \mathbf{Bc}, \mathbf{c}\rangle$ is linear, the problem is still convex.

Although $\mathbf{B}$ is intended to be a good estimation of the precision matrix (up to constants), formulations (2.2) or (2.3) do not guarantee that $\mathbf{B}$ will be positive-semidefinite, and therefore the leading eigenvalue might not be positive. One way to address this is to add the positive-semidefinite constraint in the formulation, which keeps the problem convex. However, in all of our experiments with model (2.3) the spectral radius resulted positive, so we decided to use this simpler formulation due to the power of the available solvers.

Note that we are imposing the dominant eigenvector of the graph connectivity matrix $\mathbf{A}$ to a non-binary matrix $\mathbf{B}$. We have exhaustive empirical evidence that the leading eigenvector of the matrix $\mathbf{B}$ obtained by solving (2.3), and the leading eigenvector corresponding to the resulting connectivity matrix (the binarization of $\mathbf{B}$) are very similar (see Section 2.2.6). In addition, based on [96], these type of results can be proved theoretically [103].

As shown in Section 2.2.6, when the correct centrality is imposed, our proposed model outperforms the joint regression model, both in correct reconstructed edge rates and topology. This is still true when we only have a noisy version of $\mathbf{c}$. Even if we do not have prior information at all, and we estimate the centrality from the data with a pre-run of the PC-Algorithm, we obtain improved results.

The model extension here presented is general, and the term $\langle \mathbf{Bc}, \mathbf{c}\rangle$ can be included in maximum likelihood based approaches like [9, 47, 101].

2.2.4 Implementation

Following [80], the matrix optimization (2.3) can be cast as a classical vector ℓ_1 penalty problem. The symmetry and null diagonal constraints are handled considering only the upper triangular sub-matrix of $\mathbf{B}$ (excluding the diagonal), and forming a vector $\boldsymbol{\theta}$ with its entries: $\boldsymbol{\theta} = (\mathbf{B}_{12}, \mathbf{B}_{13}, \ldots, \mathbf{B}_{(p-1)p})$. Let us consider a $pk \times 1$ column vector $\mathbf{y}$ formed by concatenating all the columns of $\mathbf{X}$. It is easy to find a $pk \times p(p-1)/2$ matrix $\mathbf{X_t}$ such that $||\mathbf{X} - \mathbf{XB}||_F^2 = ||\mathbf{y} - \mathbf{X_t}\boldsymbol{\theta}||_2^2$ (see [80] for details), and trivially $||\mathbf{B}||_{\ell_1} = 2||\boldsymbol{\theta}||_1$. The new term in the cost function is $\langle \mathbf{Bc}, \mathbf{c} \rangle$, which is linear in $\mathbf{B}$, thus there exists a matrix $\mathbf{C_t} = \mathbf{C_t}(c)$ such that $\langle \mathbf{Bc}, \mathbf{c} \rangle = \langle \mathbf{C_t}, \boldsymbol{\theta} \rangle$. The construction of $\mathbf{C_t}$ is similar to the construction of $\mathbf{X_t}$. The optimization problem (2.3) then becomes

$$\min_{\theta} ||\mathbf{y} - \mathbf{X_t}\boldsymbol{\theta}||_2^2 + \lambda_1 ||\boldsymbol{\theta}||_1 - \lambda_2 \langle \mathbf{C_t}, \boldsymbol{\theta} \rangle,$$

which can be efficiently solved using any modern ℓ_1 optimization method [98].

2.2.5 Favoring motifs in graphical models

One of the biggest challenges in bioinformatics is the estimation and understanding of genetic regulatory networks. It has been observed that the structure of these graphs is far from being random: transcription networks seem to be conformed by a small set of regulation patterns that appear much more often than in random graphs. It is believed that each one of these patterns, called motifs, are responsible of certain specific regulatory functions. Three basic types of motifs are defined [84], the "feedforward loop" being one of the most significant. This motif involves three genes: a regulator X which regulates Y, and a gene Z which is regulated by both X and Y. The representation of these regulations in the network takes the form of a triangle with vertices X, Y, Z.

Although these triangles are very frequent in GRN, the common algorithms discussed in Section 2.2.2 seem to fail at producing them. As these models do not consider any topological structure, and the total number of reconstructed triangles is usually much lower than in transcription networks, it seems reasonable to help in the formation of these motifs by favoring the presence of triangles.

In order to move towards a better motif detection, we propose an iterative procedure based on the joint regression model (2.2). After a first iteration of solving (2.2), a preliminary symmetric matrix $\mathbf{B}$ is obtained. Recall that if $\mathbf{A}$ is a graph adjacency matrix, then $\mathbf{A}^2$ counts the paths of length 2 between nodes. More specifically, the entry (i, j) of $\mathbf{A}^2$ indicates how many paths of length 2 exist from node i to node j. Back to the graphical model estimation, this means that if the entry $(\mathbf{B}^2)_{ij} \neq 0$ (a length 2 path exists between i and j), then by making $\mathbf{B}_{ij} \neq 0$ (if it is not already), at least one triangle is added. This suggests that by including weights in the ℓ_1 penalization, proportionally decreasing with $\mathbf{B}^2$, we are favoring those edges that, when added, form a new triangle.

Given the matrix $\mathbf{B}$ obtained in the preliminary iteration, we consider the cost matrix $\mathbf{M}$ such that $\mathbf{M}_{ij} = e^{-\mu(\mathbf{B}^2)_{ij}}$, μ being a positive parameter. This way, if $(\mathbf{B}^2)_{ij} = 0$ the weight does not affect the penalty, and if $(\mathbf{B}^2)_{ij} \neq 0$, it favors motifs detection. We then solve the optimization problem

$$\min_{\mathbf{B}} ||\mathbf{X} - \mathbf{X}\mathbf{B}||_F^2 + \lambda_1 ||\mathbf{M} \cdot \mathbf{B}||_{\ell_1} \ , \tag{2.4}$$

where $\mathbf{M} \cdot \mathbf{B}$ is the pointwise matrix product.

The algorithm iterates between reconstructing the matrix $\mathbf{B}$ and updating the weight matrix $\mathbf{M}$ (initialized as the identity matrix). Usually after two or three iterations the graph stabilizes.

2.2.6 Experimental results

In this section we present numerical and graphical results for the proposed models, and compare them with the original joint regression one.

As discussed in the introduction, there is evidence that most real life networks present scale-free behavior. Therefore, when considering simulated results for validation, we use the preferred-attachment model by [11] to generate graphs with this property. Namely, we start from a random graph with 4 nodes and add one node at a time, randomly connected to one of the existing nodes. The probability of connecting the new node to the node i is proportional to the current degree of node i.

Given a graph with adjacency matrix $\mathbf{A}$, we simulate the data $\mathbf{X}$ as follows [65]: let $\mathbf{D}$ be a diagonal matrix containing the degree of node i in the entry $\mathbf{D}_{ii}$, and consider the matrix $\mathbf{L} = \eta\mathbf{D} - \mathbf{A}$ with $\eta > 1$ so that $\mathbf{L}$ is positive definite. We then define the concentration matrix $\mathbf{\Theta} = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{L}\mathbf{\Lambda}^{\frac{1}{2}}$, where $\mathbf{\Lambda}$ is the diagonal matrix of $\mathbf{L}^{-1}$ (used to normalize the diagonal of $\mathbf{\Sigma} = \mathbf{\Theta}^{-1}$). Gaussian data $\mathbf{X}$ is then simulated with distribution $\mathcal{N}(0, \mathbf{\Sigma})$. For each algorithm, the parameters are set such that the resulting graph has the same number of edges as the original one. As the total number of edges is then fixed, the false positive (FP) rate can be deduced from the true positive (TP) rate. We therefore report the TP rate only, since it is enough to compare the different performances.

Including actual centrality

In this first experiment we show how our model (2.3) is able to correctly incorporate the prior centrality information, resulting in a more accurate inferred graph, both in terms of detected edges and in topology.
The graph of the example in Figure 2.2 contains 20 nodes. We generated 10 samples and inferred the graph with the joint regression model and with the proposed model (2.3) using the correct centrality.

The following more comprehensive test shows the improvement with respect to the basic joint model (2.2) when the correct centrality is included. For a fixed value of $p = 80$, and for each value of k from 30 to 50, we made 50 runs generating scale-free graphs and simulating data $\mathbf{X}$. From these data we estimated the network with the joint regression model with and without the centrality prior. The TP edge rates in Figure 2.3a are averaged over the 50 runs, and count the correctly

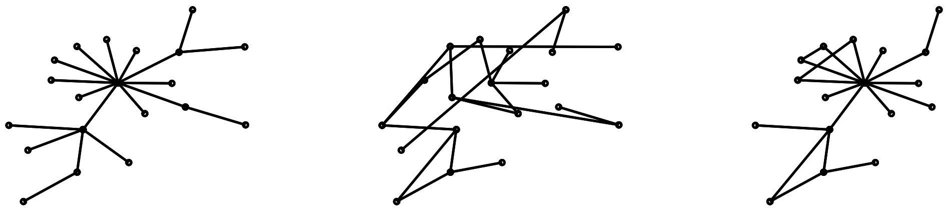
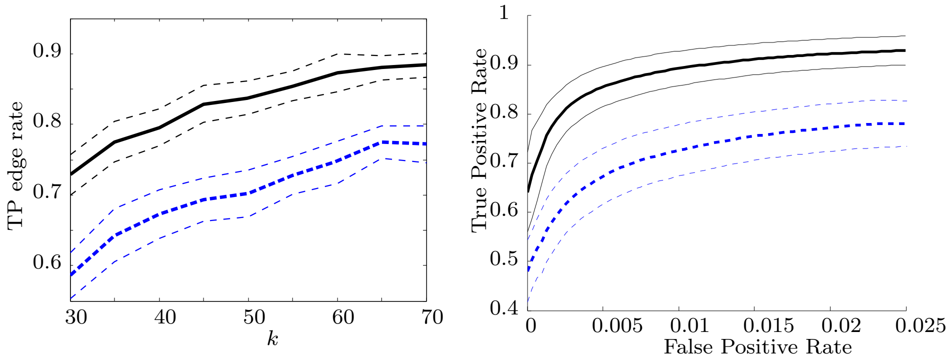


Figure 2.2: Comparison of networks estimated with the simple joint model (2.2) (middle) and with model (2.3) (right) using the eigenvector centrality. Original graph on left.

detected edges over the (fixed) total number of edges in the network. In addition, Figure 2.3b shows a ROC curve. We generated 300 networks and constructed a ROC curve for each one by varying λ_1, and we then averaged all the 300 curves. As expected, the incorporation of the known topological property helps in the correct estimation of the graph.



(a) True positive rates for different sample sizes on networks with 80 nodes.

(b) Edge detection ROC curve for networks with $p = 80$ nodes and $k = 50$.

Figure 2.3: Performance comparison of models (2.3) and (2.2). In blue (dashed), the standard joint model (2.2), and in black the proposed model with centrality (2.3). In thin lines, curves corresponding to 95% confidence intervals.

Following the previous discussion, Figure 2.4 shows the inner product $\langle \mathbf{v_B}, \mathbf{v_C} \rangle$ for several runs of model (2.3), where $\mathbf{v_B}$ is the leading eigenvector of the obtained matrix $\mathbf{B}$, $\mathbf{C}$ is the resulting connectivity matrix (the binarized version of $\mathbf{B}$), and $\mathbf{v_C}$ its leading eigenvector.

Imposing centrality estimated from data

The previous section shows how the performance of the joint regression model (2.2) can be improved by incorporating the centrality, when this topology information is available. However, when this vector is unknown, it can be estimated from the data, using an independent algorithm, and then incorporated to the optimization

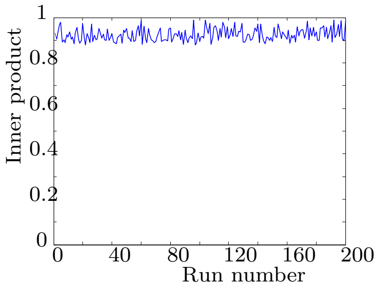


Figure 2.4: Inner product $\langle \mathbf{v_C}, \mathbf{v_B} \rangle$ for 200 runs.

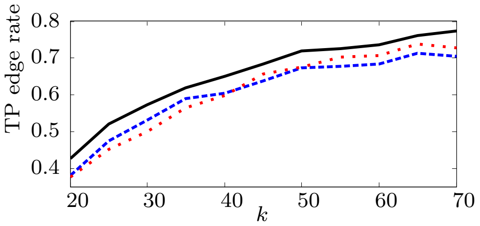


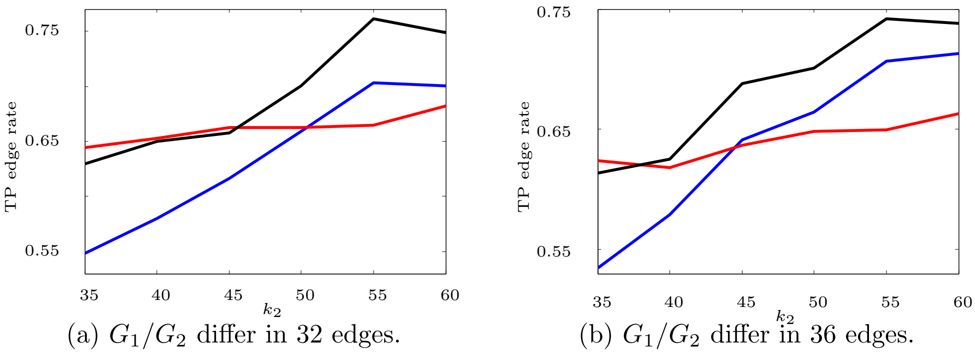
Figure 2.5: True positive edge rates for different sample sizes on a network with 100 nodes. Dashed, the joint model (2.2), dotted, the PC-Algorithm, and solid the model (2.3) with centrality estimated from data.

in model (2.3). We use the PC-Algorithm to estimate the centrality (by computing the dominant eigenvector of the resulting graph), and then we impose it as the vector $\mathbf{c}$ in model (2.3). It turns out that even with a technique not specialized for centrality estimation, this combination outperforms both the joint model (2.2) and the PC-Algorithm.

We compare the three mentioned models on networks with $p = 100$ nodes for several values of k, ranging from 20 to 70. For each value of k, we randomly generated ten networks and simulated data $\mathbf{X}$. We then reconstructed the graph using the three techniques and averaged the edge rate over the ten runs. The parameter λ_2 was obtained via cross validation. Figure 2.5 shows how the model imposing centrality can improve the other ones without any external information.

Transferring centrality

In several situations, one may have some information about the topology of the graph to infer, mainly based on other data/graphs known to be similar. For instance, dynamic networks are a good example where one may have some (maybe abundant) old data from the network at a past time T_1, some (maybe scarce) new data at time T_2, and know that the network topology is similar at the different times. This may be the case of financial, climate, or any time-series data. Outside of temporal varying networks, this topological transfer may be useful when we have two graphs of the same kind (say biological networks), which are expected to share some properties, and lots of data is available for the first network but very few samples for the second network are known. We would like to transfer our inferred centrality-based topological knowledge from the first network into the



(a) G_1/G_2 differ in 32 edges.

(b) G_1/G_2 differ in 36 edges.

Figure 2.6: True positive edge rate when estimating the network G_2 vs amount of data. In blue, the basic joint model using only $\mathbf{X}_2$, in red using the concatenation of $\mathbf{X}_1$ and $\mathbf{X}_2$, and in black the model (2.3) using only $\mathbf{X}_2$ with centrality estimated from $\mathbf{X}_1$ as prior.

second one, and by that improving the network estimation from limited data.

For these examples, we have an unknown graph G_1 corresponding to a $k_1 \times p$ data matrix $\mathbf{X}_1$, which we assume is enough to reasonably estimate G_1, and an unknown graph G_2 with a $k_2 \times p$ data matrix $\mathbf{X}_2$ (with $k_2 \ll k_1$). Using $\mathbf{X}_2$ only might not be enough to obtain a proper estimate of G_2, and considering the whole data together (concatenation of $\mathbf{X}_1$ and $\mathbf{X}_2$) might be an artificial mixture or too strong and lead to basically reconstructing G_1. What we really want to do is to transfer some high-level structure of G_1 into G_2, e.g., just the underlying centrality of G_1 is transferred to G_2.

In what follows, we show the comparison of inferring the network G_2 using only the data $\mathbf{X}_2$ in the joint model (2.2); the concatenation of $\mathbf{X}_1$ and $\mathbf{X}_2$ in the joint model (2.2); and finally the centrality estimated from $\mathbf{X}_1$, imposed in model (2.3), along with data $\mathbf{X}_2$. We fixed the networks size to $p = 100$ and the size of data for G_1 to $k_1 = 200$. Given a graph G_1, we construct G_2 by randomly changing a certain number of edges (32 and 36 edges in Figure 2.6). For k_2 from 35 to 60, we generate data $\mathbf{X}_2$, and we then infer G_2 with the methods described above. We averaged over 10 runs.

As it can be observed in Figure 2.6, the performance of the model including the centrality estimated from $\mathbf{X}_1$ is better than the performance of the classical model, both when using just the data $\mathbf{X}_2$ and the concatenated data $\mathbf{X}_1|\mathbf{X}_2$. Therefore, we can discard the old data $\mathbf{X}_1$ and keep only the structure (centrality) and still be able to infer a more accurate version of G_2.

2.2.7 Experiments on real data

International stock market data

The stock market is a very complicated system, with lots of time-dependent underlying relationships. In this example we show how the centrality constraint can help to understand these relationships with limited data on times of crisis and times of stability.

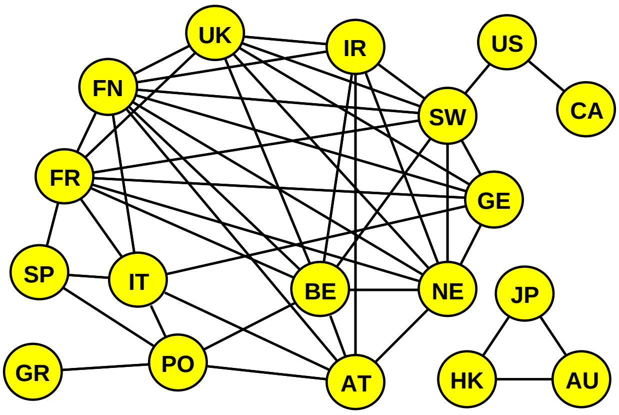


Figure 2.7: Countries network learned with the centrality model.

	97-99	07-09	09-12
LS	2.7	3.5	14.4
Model (2.2)	2.5	0.9	4.0
Model (2.3)	**1.9**	**0.6**	**2.4**

Table 2.1. Mean square error ($\times 10^{-3}$) for the different models.

We use the daily closing values (π_k) of some relevant stock market indices from U.S., Canada, Australia, Japan, Hong Kong, U.K., Germany, France, Italy, Switzerland, Netherlands, Austria, Spain, Belgium, Finland, Portugal, Ireland, and Greece. We consider 2 time periods containing a crisis, 5/2007-5/2009 and 5/2009-5/2012, each of which was divided into a "pre-crisis" period, and two more sets (training and testing) covering the actual crisis period. We also consider the relatively stable period 6/1997-6/1999, where the division into these three subsets was made arbitrarily. Using as data the return between two consecutive trading days, defined as $100\log(\frac{\pi_k}{\pi_{k-1}})$, we first learned the centrality from the "pre-crisis" period, and we then learned three models with the training sets: a classical least-squares regression (LS), the joint regression model (2.2), and the centrality model (2.3) with the estimated eigenvector. For each learned model $\mathbf{B}$ we computed the "prediction" accuracy $||\mathbf{X}_{test} - \mathbf{X}_{test}\mathbf{B}||_F^2$ in order to evaluate whether the inclusion of the topology improves the estimation. The results are presented in Table 2.1, illustrating how the topology helps to infer a better model, both in stable and highly changing periods. Additionally, Figure 2.7 shows a graph learned with the model (2.3) using the 2009-2012 training data. The discovered relationships make sense, and we can easily identify geographic or socio-economic connections.

Motif detection in *Escherichia Coli*

Along this section and the following one, we use as base graph the actual genetic regulation network of the *E. coli*. This graph contains $\approx$ 400 nodes, but for practical issues we selected the sub-graph of all nodes with degree > 1. This sub-graph G_E contains 186 nodes and 40 feedforward loop motifs.

For the number of samples k varying from 30 to 120, we simulated data $\mathbf{X}$ from G_E and reconstructed the graph using the joint model (2.2) and the iterative method (2.4). We then compared the resulting networks to the original one, both in true positive edge rate (recall that this analysis is sufficient since the total number of edges is made constant), and number of motifs correctly detected. The numerical results are shown in Figure 2.8, where it can be seen that model (2.4) correctly detect more motifs, with better TP vs FP motif rate, and without detriment of the true positive edge rate.

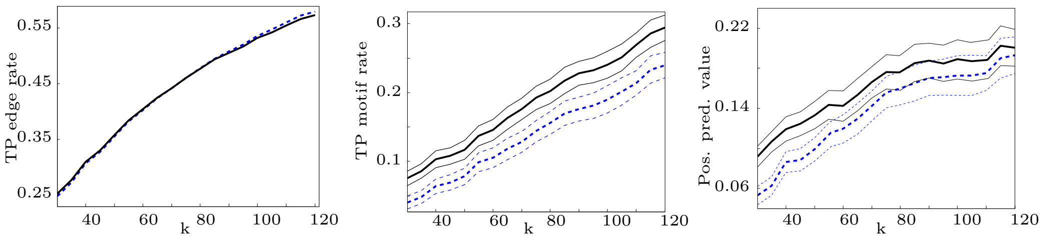


Figure 2.8: Comparison of model (2.2) (dashed) with proposed model (2.4) (solid) for the *E. coli* network. Left: TP edge rate. Middle: TP motif rate (motifs correctly detected over the total number of motifs in G_E). Right: Positive predictive value (motifs correctly detected over the total number of motifs in the inferred graph).

Centrality + motif detection

The simplicity of the proposed models allows to combine them with other existing network estimation extensions. We now show the performance of the two models here presented combined (centrality and motifs constraints), tested on the *Escherichia coli* network.

We first estimate the centrality from the data, as in Section 2.2.6. Let us assume that we know which ones are the two most central nodes (genes).[1] This information can be used to modify the centrality value for these two nodes, by replacing them by the two highest centrality values typical of scale-free networks [74]. For the fixed network G_E, we simulated data of different sizes k and reconstructed the graph with the model (2.2) and with the combination of models (2.3) and (2.4). Again, we compared the TP edge rates, the percentage of motifs detected, and the TP/FP motifs rate. Numerical results are shown in Figure 2.9, where it can be seen that, in addition to the motif detection improvement, now the edge rate is also better. Figure 2.10 shows the obtained graphs for a specific run.

[1] In this case, it is well known that *crp* is the most central node (gene), followed by *fnr*.

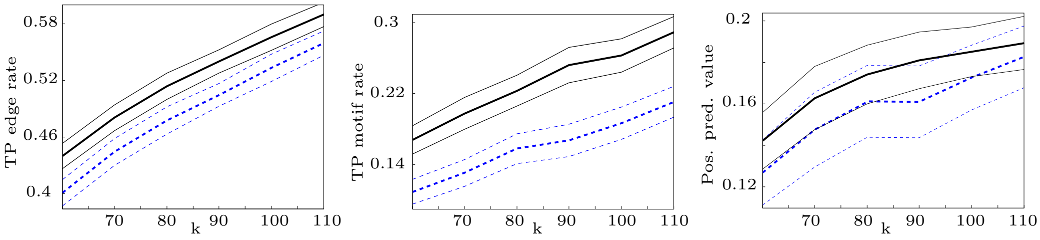


Figure 2.9: Comparison of model (2.2) (dashed) with the combination of models (2.3) and (2.4) (solid) for the *E. coli* network. The combination of the proposed extensions is capable of detecting more motifs while also improving the accuracy of the detected edges. Left: TP edge rate. Middle: TP motif rate. Right: Positive predictive value.

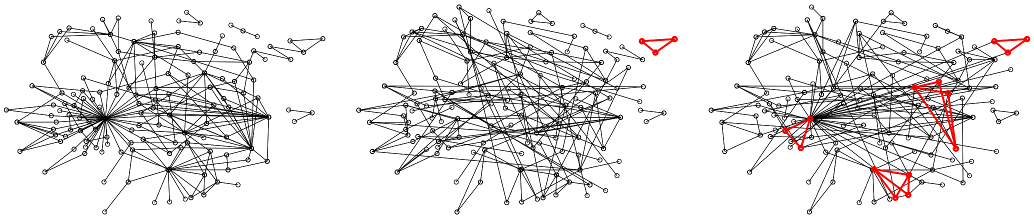


Figure 2.10: Comparison of graphs for the *E. coli* network with $k = 80$. Original network, inferred with model (2.2) and with the combination of (2.3) and (2.4). Note how the combined model is able to better capture the underlying network topology, as quantitative shown in Figure 2.9. Correctly detected motifs are highlighted.

2.2.8 Conclusions of the section and future work

We proposed two extensions to ℓ_1 penalized models for precision matrix (network) estimation. The first one incorporates topological information to the optimization, allowing to control the graph centrality. We showed how this model is able to capture the imposed structure when the centrality is provided as prior information, and we also showed how it can improve the performance of the basic joint regression model even when there is no such external information. The second extension favors the appearance of triangles, allowing to better detect motifs in genetic regulatory networks. We combined both models for a better estimation of the *Escherichia coli* GRN.

There are several other graph-topological properties that may provide important information, making it interesting to study which kind of structure can be added to the optimization problem. An algorithm for estimating with high precision the centrality directly from the data would be a great complement to the methods here presented. It is also important to find a model which exploits all the prior information about GRN, including other motifs not explored in this work. Finally, the exploitation of the methods here developed for ℓ_1-graphs, is the subject of future research.

2.3 Collaborative Graph Inference

Section summary

As mentioned in the introduction, graphical models are a very useful tool to describe and understand natural phenomena, from gene expression and brain networks to climate change and social interactions. In many cases, the data is multimodal. For example, one may want to build one network from several fMRI (functional magnetic resonance imaging) studies from different subjects, or combine different data modalities (as fMRI and questionnaires) for several subjects. To this end, in this section we combine the graph inference techniques with constraints from the sparse modeling community, which have been proven succesfull for collaborative filtering, classification and reconstruction tasks. We then derive an iterative shrinkage thresholding algorithm for solving the proposed optimization problem. Finally, the framework is validated with synthetic data and real fMRI data, showing the advantages of combining different modalities in order to infer the underlying network structure.

2.3.1 Problem formulation

The estimation of the inverse of the covariance matrix, also known as precision matrix, is a very important problem with applications in a large number of fields. The *covariance selection* problem consists in identifying the zero pattern of the precision matrix, which is of particular interest for analyzing dependencies between random variables. When the dimension of the problem is too large, or obtaining observations of the random variable is expensive, the resulting inference problem is ill-conditioned, and therefore prior information must be incorporated in order to stabilize the solution. This path was already taken in the previous section, where we assumed sparsity in the objective graph (leading to ℓ_1 penalty terms) and prior topological knowledge. In what follows, we keep the sparsity assumption, and we add a collaborative prior, in order to jointly estimate several networks sharing some structure, aiming to tackle a collaborative inference of brain connections from fMRI data.

The use of graph inference methods to analyze brain connectivity from fMRI data has been growing in the last few years, with very impressive results both from a modeling point of view, as well as algorithmic, in the sense of fast and scalable methods [56, 93]. The nature of this application makes the use of prior information a requirement: the amount of voxels measured by the equipment is extremely large, but the person cannot be inside the acquisition equipment for too long, and therefore the number of measurements is limited. However, there is a lot of information that could be used to tackle this problem effectively: on the one hand, the spatial dependency between measurements in voxels near to each other, which is usually taken into account by grouping voxels in regions according to their anatomical structure, and on the other hand, the similarity of brain connectivity patterns between different subjects.

As in the previous section, let $X = (X_1 \dots X_p)$ be a p-dimensional multivariate normal distributed variable, $X \sim \mathcal{N}(0, \mathbf{\Sigma})$, and $\mathbf{C} = \mathbf{\Sigma}^{-1}$ its concentration or precision matrix. Then two coordinates X_i and X_j are conditionally independent given the other variables if and only if $\mathbf{C}(i,j) = 0$. This property motivates the representation of the conditional dependency structure in terms of a graphical model.

Let us recall the two basic approaches that have been developed for estimating the structure of the graphical model when a sparse structure is assumed, and have been proved successful specially when working with a few data points. First, the regression approach [80]

$$\min_{B} ||\mathbf{X} - \mathbf{X}\mathbf{B}||_F^2 + \lambda_1 ||\mathbf{B}||_{\ell_1} \ , \qquad s.t.\ \mathbf{B} \text{ symmetric}, \ \mathbf{B}_{ii} = 0 \ \forall\, i, \tag{2.5}$$

and the maximum likelihood, also called Graphical Lasso [10]:

$$\max_{\mathbf{\Theta} \succ 0} \ \log\det\mathbf{\Theta} - \mathrm{tr}(\mathbf{S}\mathbf{\Theta}) - \lambda ||\mathbf{\Theta}||_{\ell_1}, \tag{2.6}$$

where $\mathbf{S}$ is the empirical covariance matrix and $||\mathbf{\Theta}||_{\ell_1} = \sum_{i,j} |\mathbf{\Theta}_{ij}|$.

Let us suppose now that we have n data matrices $\mathbf{X}^h$, $h = 1 \dots n$, which may come from different distributions, but such that the dependency graphs of all of them share the same structure, at least approximately. Then, the goal is to infer n covariance matrices, but such that they (roughly) share the non-zero pattern. If we are only interested in the structure of the networks, a naïve approach could be to concatenate all the data matrices together, and run one of the techniques mentioned above to infer one graph. However, the mix of data from different distributions would lead to a poor estimation of the inverse covariance matrix. Additionally, we lose the possibility of estimating one matrix per dataset, which could be of interest for the analysis.

A more suitable approach is to keep all the individual estimators for each inverse covariance matrix, and incorporate one regularizer promoting that all these matrices share the same support. The corresponding regularization term comes from the Group Lasso [100], described in the beginning of this section.

Let us describe the particular use of the Group Lasso for formulation (2.6), being the extension to the other formulation completely analogous. In this case, we group all the entries (i,j) of the matrices $\mathbf{\Theta}^h$, and form an $n-$dimensional vector whose l_2 norm will be penalized in the objective optimization function. This way, the sum of penalty terms for all groups promotes sparsity, in the sense that only a few groups will be active (and so each matrix $\mathbf{\Theta}^h$ will be sparse), but once a group is active, the corresponding n coefficients (the (i,j) entries for all $\mathbf{\Theta}^h$) will be all non-zero in general. The optimization problem to solve is then

$$\max_{\mathbf{\Theta}^1, \mathbf{\Theta}^2, \dots, \mathbf{\Theta}^n \succ 0} \sum_h \log\det\mathbf{\Theta}^h - \sum_h \mathrm{tr}(\mathbf{S}^h\mathbf{\Theta}^h) - \lambda \sum_{i,j} || \left(\mathbf{\Theta}_{ij}^1, \mathbf{\Theta}_{ij}^2, \dots, \mathbf{\Theta}_{ij}^n\right) ||_2 \ . \tag{2.7}$$

This problem is still convex, and we have adapted the ISTA algorithm [82]. The code is available in `www.fing.edu.uy/~mfiori`.

2.3.2 Synthetic data examples

In this section the model and algorithm are assessed with two different experiments: in the first one we show how the performance of the grouped methodology improves as the number of groups grows, and we also compare it with concatenating the data instead of the grouping approach. In the second one we show that this methodology is able to mix different kinds of data (e.g. Gaussian and discrete).

For the first experiment, we randomly generated six precision matrices with the same support (but different non-zero values), for $p = 60$. For each matrix we simulated Gaussian data $\mathbf{X}^h \in k \times p$ for $k = 30$. Figure 2.11 (left) shows how the performance of the model (2.7) improves with the number of considered groups, and how the concatenation degrades the performance. In solid black line, estimation using only one dataset ($\mathbf{X}_1$). Below it (dashed black), using the concatenation of different subsets. Above it, using the grouped methodology with: 2, 3, 4, 5 and 6 groups (blue and red).

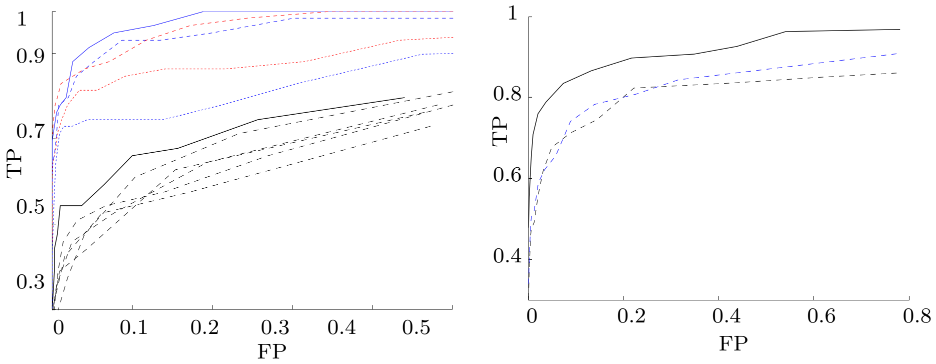


Figure 2.11: True Positives vs False Positives on detected edges of the true graph. Left: comparison for several groups. Right: Discrete and Gaussian data. In dashed blue, using only gaussian data $\mathbf{X}$, in dashed black using only discrete data $\mathbf{Y}$, and in solid black using the grouped methodology.

For the second experiment, we generated a Gaussian Graphical Model and a Discrete Graphical model, sharing the same zero-pattern of the inverse covariance matrix, and simulated data from both of them, $\mathbf{X}$ and $\mathbf{Y}$ respectively. Figure 2.11 (right) shows the performance when inferring the zero-pattern only from $\mathbf{X}$, only from $\mathbf{Y}$, and with the combination of both via the optimization problem (2.7).

2.3.3 Application to fMRI data

Here we show how this collaborative learning can help to build brain networks for different groups of subjects. For an fMRI study of 155 subjects, we split the dataset into 105 for training and 50 for testing (data from `http://www.haririlab.com/brain.php`). With the training we built one network for males ($\mathbf{A_M}$) and another one for females ($\mathbf{A_F}$), using the grouped methodology (2.7). Then, for each subject in the testing set, we built the brain network from the fMRI data, and classified as male of female according to the closest graph adjacency matrix ($\mathbf{A_M}$ or $\mathbf{A_F}$). To compare the performance, we also classified each subject with a nearest neighbor criteria with respect to all the subjects in the training set. The results are shown in Table 2.2, where it can be observed that when building one coherent network for each gender, the classification improves significantly.

	NN	Grouped/NN
Classification Performance	60%	80%

Table 2.2. Comparison of classification results for individual and collaborative brain network estimation.

2.4 Estimation of Dynamic Mobility Graphs

Section summary

The interest in problems related to graph inference has been increasing significantly during the last decade. However, the vast majority of the problems addressed are either static, or systems where changes in one node are immediately reflected in other nodes. In this section we address the problem of mobility graph estimation, when the available dataset has an asynchronous and time-variant nature. More specifically, let us suppose that we are given the number of people at different physical locations at every time. Assume that these people are moving from one location to another, but the travel duration depends on the origin/destination pair, and we cannot observe these transitions between locations. The goal is to infer the mobility pattern of the subjects from these observational data. The problem is very ill-posed, since several combination of paths might explain a certain observation. We present a formulation for this problem consisting on an optimization of a cost function having a fitting term to explain the observations with the dynamics of the system, and a sparsity-promoting penalty term, in order to select the paths actually used. The formulation is tested on two publicly available real datasets on US aviation and NY city taxi traffic, showing the importance of the problem and the applicability of the proposed framework.

2.4.1 Problem description

The significant growth of available data, both in quantity and diversity, has motivated an increased interest in problems related with graph inference or network estimation, from gene regulatory networks and brain connectivity graphs using fMRI data to social networks and micro-blog data.

Several graph inference algorithms have been recently introduced [9, 58, 66, 73, 80, 101], showing that sparse models provide useful formulation for addressing the problem, and introducing a significant number of applications. For instance, in [7] the authors study the problem of inferring the “online news” network topology and dynamics from the spread of blog posts and news articles. Other common applications are the estimation of brain connectivity from fMRI data [93], or gene regulatory networks from micro-array data [80], and the selection of questions from a large and correlated questionnaire [21].

However, all these works address the network inference problem either for static graphs, or for graphs that exhibit very particular dynamics, i.e., where the interactions between nodes are instantaneous, and once the information arrives to a certain node, it gets “infected,” meaning that the node cannot go back to its previous state. This is the case, for instance, of social networks or blogs and micro-blogs data.

In the present work, we study the problem of estimating the mobility pattern of entities in a more general setting. Let us assume that we can count the number of entities at different sites along time. For example, we may know how many people are on each track of a subway station connecting several lines, at every time. The goal is to infer the general mobility pattern within the station, to infer how connections are taken by the passengers.

The main difference with the other graph inference problems previously mentioned is its dynamic aspect: in this problem, the time it takes to go from one site to another depends on the sites and is *unknown*. This simple modification adds a whole layer of complexity to the problem, rendering it very challenging. To the best of our knowledge, this problem is here studied for the first time.

If the time to go from one site to another is the same for all paths and the movements are all synchronized, then this (much easier) problem can be thought as a Markov model, and estimation of the transition matrix is well studied. If, on the other hand, we have two observations of the number of people at each site, but accompanied by tracking information (meaning that we know, for example, that one person was at site i and then, in the following observation, at site j), then the estimation of the mobility matrix is straightforward. This approach is used for instance for income mobility estimation [45].

The variability of the time spent in each path, as well as the asynchronous component, make this hard problem unique. There are several problems falling in the category of the formulation presented in this section. For instance, let us suppose that we know how many airplanes are at each airport at every time, and we want to infer the routes between the airports, and with what frequency an airplane takes one of the paths.

As described throughout this section, this problem is extremely ill-conditioned in general, since there might be several ways to explain certain observations. Thus, the selection of the right type of regularizers plays a critical role, even more so than in other related formulations.

The rest of the section is organized as follows. In Section 2.4.2, we present the problem formulation; in Section 2.4.3, we propose an optimization algorithm for addressing it . Experimental results with real data are presented in Section 2.4.4, and final remarks are provided in Section 2.4.5.

2.4.2 Problem formulation

Let us suppose that we are observing entities moving through different sites over time. Here the entities may be, for example, people, airplanes or cars, and the sites may be physical locations such as train stations, airports, or regions in a city. Given n such sites, we observe (exactly or approximately) the number of entities in each site, at discretized time intervals $t = 1, \dots, T$. Our goal is to infer, from this information, the mobility pattern of the entities.

This problem, although simple to describe, presents several difficulties. First, we cannot observe an entity while it is moving from one site to another. If an entity is traveling from site i to site j with a travel time d, we can only observe it as an aggregate in node i at time t, and then in node j at time $t+d$: in the interval $(t, t+d)$, the entity becomes unobservable. Additionally, these travel times are unknown, and they might depend on the path and also on the particular entity itself. On the other hand, each movement might have several possible explanations (i.e., there are many ways of traveling from site i to site j); these uncertainties make the problem ill-posed in general. These difficulties render this problem extremely hard and challenging.

We will formally model the desired mobility pattern of the entities as a graph of transitions between n nodes, where each node corresponds to a site. We first represent the given/observed information as an $n \times T$ matrix $\mathbf{U}$, where the entry u_{it} contains the number of entities at node i during time interval t.In order to capture the described mobility problem, we augment the graph with $n(n-1)$ extra nodes, that model the transition between every (ordered) pair of original nodes. Observe that each transition node is associated with a directed path (say, from node i to node j), and represents an "in transit" site where the entities virtually stay for the travel duration between node i and node j. Of course, the number of entities in each transition node is not directly observable. We refer to the observable nodes as *original* and the unobservable transition nodes as *transition* nodes.

Notice that any prior knowledge about the routing topology of the system would allow to remove transition nodes, simplifying the problem. In the general case, all $n(n-1)$ paths are eligible a priori.

Transitions from one node to another are modeled stochastically. An entity present in node i at a given time can either stay at the same site i with probability a_i, or choose an outgoing path k, connecting the node i with one of the remaining $n-1$ nodes, with probability q_k. The probability d_k of staying in the transition node k models the travel duration from i to j. See Figure 2.12 for a visual representation og these quantities.

Thus, for each transition node, linking original nodes i and j, we have the following associated unknowns: (1) the probability of going from i to j, (2) the probability of staying in the transition node, and (3) the amount of entities at the transition node at each time interval.

These unknowns are globally represented by a vector $\mathbf{q}$ with $n(n-1)$ entries, containing the probabilities of going from one original node to another (i.e., the probability of going from one original node to the transition node associated with that path); a vector $\mathbf{d}$ with $n(n-1)$ entries, containing the probabilities of staying

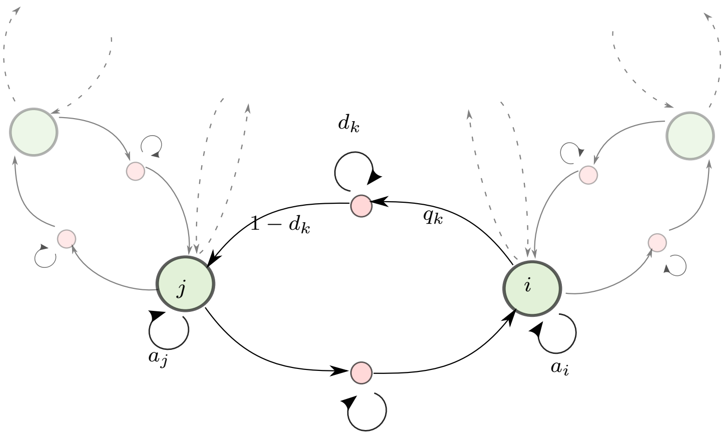


Figure 2.12: Example graph of the formulation. The big nodes (in light green) are the *original nodes*, where we can observe the number of entities at every time. The smaller nodes (in light pink) are the *transition nodes*, which represent the "in transit" state, and are not observable.

in the corresponding transition node; and a $n(n-1) \times T$ matrix $\mathbf{V}$ containing the number of entities at each transition node (or path) at each time.

Each one of the $n(n-1)$ entries of these unknowns ($\mathbf{q}$, $\mathbf{d}$ and the rows of $\mathbf{V}$) are associated with an ordered pair of the original nodes. We order these variables according to the destination node first, and then according to the source node (i.e., co-lexicographic order).

Additionally, we have an n dimensional vector $\mathbf{a}$, with the probability of staying at each original node.

The knowledge of the matrix $\mathbf{V}$ would already give plenty of information about the mobility pattern, since it contains, for instance, the active and non-active paths. Vectors $\mathbf{q}$ and $\mathbf{d}$ complement this information, showing which are the most transited paths, as well as timing information.

Notation. Since the dynamics of the system is the principal element of the proposed formulation, let us summarize the notation for time-shifted versions of the main matrices.

We denote by $\mathbf{U}_2$ and $\mathbf{U}_1$ the $n \times (T-1)$ matrices formed by taking the original matrix $\mathbf{U}$ and removing the first and the last column respectively.

In the same way, we denote by $\mathbf{V}_2$ and $\mathbf{V}_1$ the $n(n-1) \times (T-1)$ matrices formed by taking the matrix $\mathbf{V}$ and removing the first and the last column respectively.

As aforementioned, each entry of vectors $\mathbf{q}$ and $\mathbf{d}$ is associated with a path, and therefore each index corresponds to an ordered pair (i, j). The entries are ordered in co-lexicographic order (i.e., first using destination node j and then origin node i). Since we need information of the mobility from the original nodes to the paths (or transition nodes), and also from the paths to the original nodes, we need to be able to re-order these variables according to the source node (i.e., lexicographic order). Therefore, we denote by $\mathbf{P}$ the permutation matrix such that $\mathbf{Pq}$ is in lexicographic order.

Additionally, in order to obtain the number of entities arriving to a certain node i, we have to add up all the entities arriving from the paths having destination node i. To do so, we construct an $n \times n(n-1)$ matrix $\mathbf{M}$, having $n-1$ ones per row. In the first row, the ones are in the first block of $n-1$ columns, in the second row they are in the following block of $n-1$ columns (meaning starting at column n) and so on:

$$M = \underbrace{\begin{pmatrix} \underbrace{1 \ \cdots \ 1}_{n-1} & & & \mathbf{0} \\ & \underbrace{1 \ \cdots \ 1}_{n-1} & & \\ & & \ddots & \\ \mathbf{0} & & & \underbrace{1 \ \cdots \ 1}_{n-1} \end{pmatrix}}_{n(n-1)}.$$

This way, since $\mathbf{q}$, $\mathbf{d}$, and $\mathbf{V}$ are in co-lexicographic order, by left multiplying by $\mathbf{M}$, we are adding up through all the transition nodes with the same destination node.

In what follows, we present two complementary formulations for the described problem, leading to the results presented in Section 2.4.4.

Joint $(\mathbf{a}, \mathbf{q}, \mathbf{d}, \mathbf{V})$ formulation

The first formulation contemplates the effects of all the unknowns, in the sense described above.

In order to simplify the explanation of the equations governing the system dynamics, which lead to the complete formulation here presented, let us begin by analyzing a single node in particular at certain time.

The number of entities at node i at a given time $t+1$ is equal to the number of entities that were at node i at time t and stayed, plus the number of entities that were traveling towards node i and arrived at time $t+1$. This is

$$u_{i,t+1} = a_i u_{i,t} + \sum_j (1 - d_k) v_{k,t},$$

where a_i is the probability of staying (fraction of entities staying) at node i, and k indexes the transition node associated with the path $j \to i$. Hence, $v_{k,t}$ is the number of entities on their way from i to j at time t, and $1 - d_k$ is the probability of leaving the transition node and arriving to j.

This can be re-written for all nodes and for all time intervals, in the matrix form

$$\mathbf{U}_2 = \mathbf{A}\mathbf{U}_1 + \mathbf{M}(\mathbf{I} - \mathbf{D})\mathbf{V}_1, \tag{2.8}$$

where $\mathbf{A}$ and $\mathbf{D}$ are diagonal matrices with the elements of $\mathbf{a}$ and $\mathbf{d}$ in the diagonal, respectively, and $\mathbf{I}$ is the identity matrix.

In a similar way, we can describe the number of entities at every transition node at each time. The number of entities at the transition node k at a given time $t+1$ is equal to the number of entities that were at that path at time t and stayed (have not arrived to destiny yet), plus the number of entities entering the transition node from the corresponding node i. Now, the same matrix $\mathbf{M}$ used to add up all the paths going to one given node, is now used to replicate each value of $\mathbf{U}$ $n-1$ times, in order to distribute it to all the paths starting at that node. Then, the matrix $\mathbf{P}$ is used to reorder the resulting matrix. An equation similar to (2.8) can then be stated

$$\mathbf{V}_2 = \mathbf{D}\mathbf{V}_1 + \mathbf{Q}\mathbf{P}\mathbf{M}^T\mathbf{U}_1, \tag{2.9}$$

where $\mathbf{Q}$ is a diagonal matrix with the elements of $\mathbf{q}$ in the diagonal.

We can now introduce a fitting function, to be minimized, that penalizes deviations from the model given by (2.8) and (2.9). A suitable choice is, for example,

$$\begin{aligned} f(\mathbf{a}, \mathbf{q}, \mathbf{d}, \mathbf{V}) = & \frac{1}{2}\|\mathbf{A}\mathbf{U}_1 + \mathbf{M}(\mathbf{I}-\mathbf{D})\mathbf{V}_1 - \mathbf{U}_2\|_F^2 \\ & + \frac{1}{2}\|\mathbf{D}\mathbf{V}_1 + \mathbf{Q}\mathbf{P}\mathbf{M}^T\mathbf{U}_1 - \mathbf{V}_2\|_F^2. \end{aligned} \tag{2.10}$$

Let us now add several constraints and priors which help to better solve this ill-posed problem.

First, there are several constraints related to the nature of the unknowns. Namely, the variables $\mathbf{a}$, $\mathbf{q}$, and $\mathbf{d}$ are vectors representing probabilities, so each entry of these three unknowns must be in $[0, 1]$. The matrix $\mathbf{V}$ contains the number of entities in each path, so it must be $v_{k,t} \geq 0$ for all k, t. In addition, for each node the probabilities associated with outgoing edges should add up one. For the transition nodes, this is trivial since there is only one outgoing edge, so in this formulation the probabilities are d_k and $1 - d_k$. For the original nodes, the probability of staying in the node a_i plus the probability of leaving to any path (given by q_k) should add up one. This can be written, for all the nodes, as:

$$\mathbf{a} + \mathbf{M}\mathbf{P}\mathbf{q} = \mathbf{1},$$

where $\mathbf{1}$ is a vector of ones. Here the vector $\mathbf{a}$ can be written in terms of $\mathbf{q}$ as $\mathbf{a} = \mathbf{1} - \mathbf{M}\mathbf{P}\mathbf{q}$, and therefore this constraint can be directly incorporated in the formulation, making f depend only on the other unknowns: $f(\mathbf{q}, \mathbf{d}, \mathbf{V})$.

The last constraint is related to the total number of entities in the system. We assume that the number of entities is constant over time, and therefore each column of $\mathbf{U}$ plus the corresponding column of $\mathbf{V}$ must be constant. This assumption can be easily removed by adding or subtracting the number of entities entering or leaving the system at each time; it is reasonable to assume that this number can be observed in practice.

Note that the function f in (2.10) is biconvex (the variables $\mathbf{d}$ and $\mathbf{V}$ are multiplying each other), and all the constraints are convex sets. A common way to address the minimization of a biconvex function is to alternatively fix one variable and solve for the other ($\mathbf{V}$ and $(\mathbf{q}, \mathbf{d})$ in this case).

Additionally, as mentioned above, some solutions are indistinguishable from each other, making the problem extremely ill-posed. In the mathematical formulation, this can be observed in the matrix $\mathbf{M}$, which basically combines $n-1$ rows into one. Therefore, additional prior information is still needed to regularize this inverse problem.

A very reasonable assumption is that not every original node is connected to all the others (meaning that the graph is not complete), but on the contrary, that most of the paths are unused (non-existent in the physical world). This can be incorporated to the formulation by means of a sparsity promoting norm, such as ℓ_0 or ℓ_1, used as a penalty term for the unknowns we want to sparsify.

Moreover, observe that if a certain entry of $\mathbf{q}$ is non-zero, this means that the corresponding path is active. Thus, the corresponding entry of $\mathbf{d}$ and row of $\mathbf{V}$ should be also active. On the contrary, if a path is inactive, then all the corresponding entries of $\mathbf{q}$, $\mathbf{d}$, and $\mathbf{V}$ should be zero. This suggest a group lasso type of approach [100], which is known to promote either active or inactive groups. In this case, we consider $n(n-1)$ groups (one per transition node), formed by the corresponding entry of $\mathbf{q}$, of $\mathbf{d}$, and the complete corresponding row of $\mathbf{V}$.

This group Lasso approach has both effects at the same time: promoting a sparse number of active paths, and enforcing that if a path is inactive, then all the corresponding variables should be zero.

Then, the resulting formulation is

$$\min_{\substack{\mathbf{q}\geq 0,\, \mathbf{d}\geq 0,\, \mathbf{V}\geq 0 \\ \mathbf{U}^T\mathbf{1}+\mathbf{V}^T\mathbf{1}=N\mathbf{1}}} f(\mathbf{q},\mathbf{d},\mathbf{V}) + \lambda \sum_{k=1}^{n(n-1)} \|\mathbf{w}_k\|_2, \tag{2.11}$$

where N is the total number of entities, λ is a parameter controlling the sparsity of the solution, and $\mathbf{w}_k$ is the $k-th$ row of the matrix $\mathbf{W}$, formed by taking the $k-th$ row of matrix $\mathbf{V}$ concatenated with q_k and d_k.

The second term is the ℓ_1 norm of the vector containing the norms of the rows of $\mathbf{W}$, which is the convex relaxation of the ℓ_0 pseudo-norm of that vector (this ℓ_0 pseudo-norm counts the number of active rows). We would actually like to solve the problem with the ℓ_0 pseudo-norm, but this problem is known to be NP, so this convex relaxation is generally used instead. This $\ell_0 - \ell_1$ relaxation has been studied, and under some conditions, the solutions of the problems coincide [34]. However, in many situations, the ℓ_1 relaxation is not enough to enforce sparsity (solve the original ℓ_0 problem), and an iterative reweighted ℓ_1 or ℓ_2 minimization scheme is used [29]. That is, one computes the solution of the ℓ_1 or ℓ_2 relaxed problem, then uses the obtained solution vector to compute weights for each entry, and iteratively computes new solutions using those weights in the penalty term. This last approach is the one used in this section.

Since this problem is non-convex, a good initialization of the optimization is crucial. In the next section, we provide an independent formulation to estimate $\mathbf{V}$, and use it as initialization for the optimization (2.11).

Estimation of $\mathbf{V}$

We present here an alternative analysis of the system dynamics, based solely in matrices $\mathbf{U}$ and $\mathbf{V}$, in order to obtain an estimate of $\mathbf{V}$ to use as starting point for the formulation in (2.11). However, this problem is interesting by itself, since the matrix $\mathbf{V}$ already contains very useful information, as discussed above.

Let us first focus in the path going from node i to node j, and let us assume it is indexed by k. Suppose an entity is in node i at time t, and decides to go to node j trough the associated transition node k. Then, the difference between $u_{i,t}$ and $u_{i,t+1}$ is the entity who left. On the other hand, the difference between $v_{k,t}$ and $v_{k,t+1}$ is that very same entity who entered the path k (although this is not observable). The same will happen at time $t+d$ when the entity arrives to node j.

Now, let $\mathbf{A}$ and $\mathbf{B}$ be two (unknown) $n(n-1) \times (T-1)$ matrices such that $a_{k,t}$ contains the number of entities leaving the transition node k at time t (and therefore arriving at the associated original node j at time $t+1$), and $b_{k,t}$ contains the number of entities leaving the original node i at time t (and therefore entering the transition node k at time $t+1$).

From the previous description, we have that $b_{k,t} - a_{k,t} = v_{k,t+1} - v_{k,t}$, and hence

$$\mathbf{B} - \mathbf{A} = \mathbf{V}_2 - \mathbf{V}_1.$$

Thus, $\mathbf{V}$ can be reconstructed from $\mathbf{A}$ and $\mathbf{B}$, as the row cumulative sum of $\mathbf{B}-\mathbf{A}$. This can be compactly written as

$$\mathbf{V} = (\mathbf{B} - \mathbf{A})\mathbf{C}, \tag{2.12}$$

where $\mathbf{C}$ is the upper triangular matrix with ones in every entry above the main diagonal, including the diagonal itself.

A similar analysis leads to a relationship including $\mathbf{U}$, $\mathbf{A}$ and $\mathbf{B}$. Namely, adding up all the entities entering a certain node and subtracting all the entities who left that node, we obtain the difference $u_{i,t+1} - u_{i,t}$. This is:

$$\mathbf{U}_2 - \mathbf{U}_1 = \mathbf{M}\mathbf{A} - \mathbf{M}\mathbf{P}^T\mathbf{B}. \tag{2.13}$$

We look for two matrices $\mathbf{A}$ and $\mathbf{B}$, having non-negative elements and satisfying (2.13). The constraint from (2.13) can be enforced by using a penalty function, for instance, $\|\mathbf{U}_2 - \mathbf{U}_1 - \mathbf{M}\mathbf{A} + \mathbf{M}\mathbf{P}^T\mathbf{B}\|_F^2$.

Since all entries in matrix $\mathbf{V}$ must be non-negative, from (2.12), we can impose the constraint directly in terms of $\mathbf{A}$ and $\mathbf{B}$. Namely, $(\mathbf{B} - \mathbf{A})\mathbf{C} \geq 0$.

As with the previous general formulation, this problem is ill-conditioned. Again, this can be noted by observing the role of the matrix $\mathbf{M}$. Nevertheless, there is prior information that can be included in order to make the problem better conditioned. First, the rows of both $\mathbf{A}$ and $\mathbf{B}$ correspond to paths or transition nodes, and therefore these rows should be jointly active or inactive, also promoting that a low number of rows is simultaneously active. This is achieved by the group lasso penalty as discussed in the previous section. However, in this case, given that the unknowns are modeling the differential component of $\mathbf{U}$ and $\mathbf{V}$, it is also reasonable to ask for each row in particular to be sparse, since we are assuming that there are some time intervals where there is no entity entering a particular node.

This leads to the hierarchical lasso formulation [86], where a group lasso penalty is used to select only a few active groups, but also an ℓ_1 (or ℓ_0) penalty to promote sparsity inside each group as well.

Then, $\mathbf{A}$ and $\mathbf{B}$ are estimated by

$$\begin{aligned}\{\mathbf{A}^*, \mathbf{B}^*\} = \underset{\substack{\mathbf{A}\geq 0, \mathbf{B}\geq 0,\\ (\mathbf{B}-\mathbf{A})\mathbf{C}\geq 0}}{\operatorname{argmin}} \; & \|\mathbf{U}_2 - \mathbf{U}_1 - \mathbf{M}\mathbf{A} + \mathbf{M}\mathbf{P}^T\mathbf{B}\|_F^2 \\ & + \lambda_2 \sum_{k=1}^{n(n-1)} \|[\mathbf{A}\,\mathbf{B}]_k\|_2 + \lambda_1 \|[\mathbf{A}\,\mathbf{B}]\|_1,\end{aligned} \tag{2.14}$$

where $[\mathbf{A}\,\mathbf{B}]$ is the horizontal concatenation of $\mathbf{A}$ and $\mathbf{B}$, and $[\mathbf{A}\,\mathbf{B}]_k$ is its $k-th$ row. Once again, to further promote sparsity we used an iterative reweighted approach.

After solving (2.14), the matrix $\mathbf{V}$ is computed as $\mathbf{V} = (\mathbf{B}^* - \mathbf{A}^*)\mathbf{C}$ and used as initialization for the general formulation presented in Section 2.4.2.

2.4.3 Optimization

The optimization of both formulations is standard but not trivial, since there are several components contributing to the difficulty of the minimization such as non-smoothness, constraints, and non-convexity. Let us emphasize that the novelty of this work does not reside in the optimization, but on the presentation of the problem and its formal formulation.

Let us start with the first formulation in (2.11). Since f is biconvex, we proceed by alternating minimization over $\mathbf{V}$ and $(\mathbf{q}, \mathbf{d})$, with the other variable fixed.

For each subproblem, the constraint $\mathbf{U}^T\mathbf{1} + \mathbf{V}^T\mathbf{1} = N\mathbf{1}$ is added to the optimization by means of the augmented Lagrangian method, which consists in adding a smooth term (or two terms that can be combined into one, as done here) with a new auxiliary variable $\mathbf{h}$. For instance, when $\mathbf{q}$ and $\mathbf{d}$ are fixed, the problem to solve is

$$\min_{\mathbf{V}\geq 0} f(\mathbf{q}, \mathbf{d}, \mathbf{V}) \; + \; \lambda \sum_{k=1}^{n(n-1)} \|\mathbf{w}_k\|_2 \; + \; \frac{\mu}{2}\|\mathbf{U}^T\mathbf{1} \; + \; \mathbf{V}^T\mathbf{1} \; - \; N\mathbf{1} \; + \; \mathbf{h}\|_F^2, \tag{2.15}$$

where μ is a parameter which does not affect the convergence, and it was set to $\mu = 2$ in the experiments.

The procedure is iterative. At each iteration l, the objective function in (2.15) is minimized, and then the auxiliary variable is updated as $\mathbf{h}^{l+1} = \mathbf{h}^l + \mathbf{U}^T\mathbf{1} + \mathbf{V}^T\mathbf{1} - N\mathbf{1}$.

At each iteration, the minimization of the non-smooth function in (2.15) is solved by standard techniques, consisting of gradient descent combined with the vector soft-thresholding operator (for more details, see [86, 98]).

The optimization for $(\mathbf{q}, \mathbf{d})$ with $\mathbf{V}$ fixed is completely analogous.

The optimization of the second formulation (2.14) is very similar. The non-negativity constraint can be added in a similar way as the previous one [76], and the minimization of the subproblem, which now has two non-smooth terms (the hierarchical lasso), admits also a very simple methodology with the same computational complexity as the group lasso [86].

2.4.4 Experimental results

We now present experimental results with two publicly available datasets containing real transportation data. The first one collects all the internal flights in the United States in a given month, and the second one contains all the taxi trips in New York in 2013.

Airports routes

Suppose that we are given the number of airplanes at a given set of airports at every time, is it possible to recover the airplane routes and the trip durations? As mentioned in Section 2.4.1, this is an interesting and challenging problem.

We apply the proposed formulation to the analysis of a dataset containing all the US internal flights from 1987 to 2014.[2] We analyze the most recent available month (which is November 2014), then select 11 important airports in the US, and finally considered all the airplanes which have departed from or landed in any of those airports during that month. The reduced number of airports is not due to theoretical limitations of the proposed formulation, and only follows computational power constraints.

The selected airports are: Dallas/Fort Worth International Airport (DFW), Chicago O'Hare International Airport (ORD), Miami International Airport (MIA), Los Angeles International Airport (LAX), John F. Kennedy International Airport (JFK), LaGuardia Airport (LGA), Boston Logan International Airport (BOS), San Francisco International Airport (SFO), Ronald Reagan Washington National Airport (DCA), McCarran International Airport (LAS), and San Diego International Airport (SAN).

Since every airplane has a unique identifier (the *Tail Number*), it can be tracked to determine at which airport it was at every time, or if it was flying (i.e., present in no airport). With this information, and dividing the complete month into 15 minutes intervals,[3] we construct the matrix $\mathbf{U}$ and the ground truth matrix $\mathbf{V}$.

Using as input the matrix $\mathbf{U}$, we estimated $\mathbf{V}$ using the formulation in (2.14), and then used it as starting point for the formulation (2.11). The results are shown in Figure 2.13.

The inferred graph is very similar to the ground truth graph at first sight. Although some edges disappeared, and some other routes appeared, the general

[2] Dataset publicly available at `https://catalog.data.gov/dataset/airline-on-time-performance-and-causes-of-flight-delays`

[3] Notice that the intervals are quite long, considering that, according to Wikipedia, there were 81 and 71 landings/takeoffs per hour in JFK and LGA in 2010, respectively.

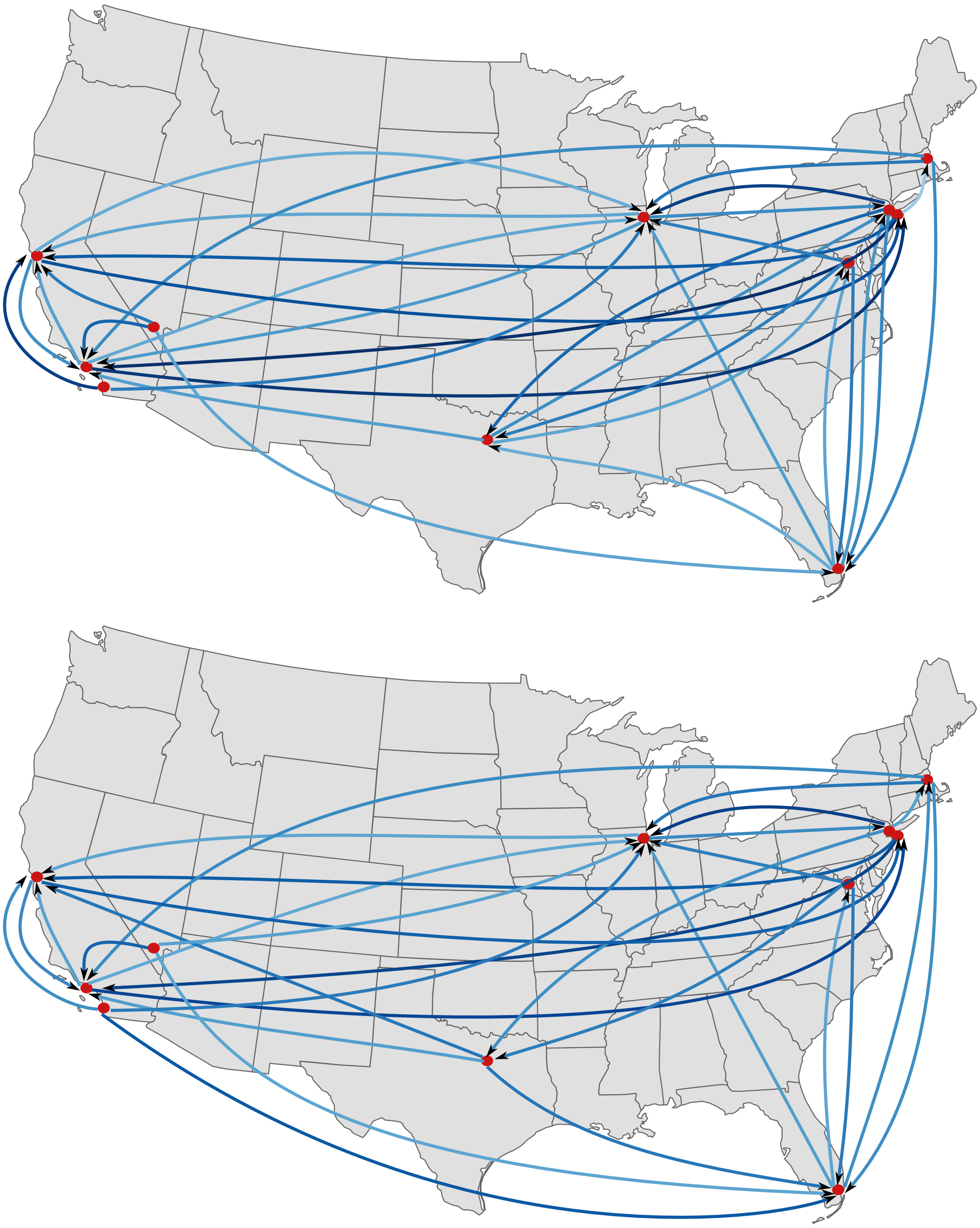


Figure 2.13: Results of the graph estimation for the airplane routes dataset. Above: ground truth. Below: inferred network. The color indicates the probability q_k of the corresponding path.

topology of the network is recovered. Also, it can be observed that the main circuits also have high probability in the estimated network: for instance, flights from LAX to JFK and the way back.

Since the two airports in New York are very close to each other in the figure, it cannot be observed, but there are no edges between LGA and JFK in the estimated graph (and of course neither in the ground truth graph). The same happens with the airports of San Diego and Los Angeles.

New York taxis

A very interesting dataset has been recently released, containing all the trips of New York yellow taxis during 2013.[4] The dataset consists of millions of records, each record corresponding to a certain trip. Each record contains (among other information) the vehicle identifier, the pickup and dropoff date and times, trip time in seconds, and the GPS coordinates of the pickup and dropoff locations.

We chose to limit the data to the trips within Manhattan, and we divided it into the 10 regions showed in Figure 2.14.

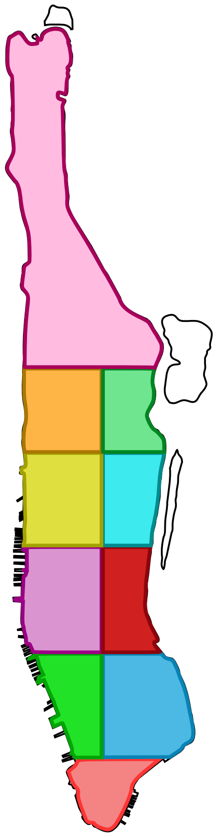


Figure 2.14: Selected regions of Manhattan.

The experiments described below were carried out with the data corresponding to one day, from 8:00am to 9:00pm, dividing the time interval into intervals of one minute.

Let us suppose that we can observe the taxis location only when they are picking up or dropping off a passenger, which is actually the available data, as

[4] Data publicly available at `http://chriswhong.com/open-data/foil_nyc_taxi/`

described above. However, we will only make use of this information, forgetting which point is a pick up or a drop off, the taxi identifier, and the trip time. All this information will be used as ground truth, in order to compare the results obtained by estimating the mobility pattern solely from the described data: time and GPS coordinates for the pickup and drop off locations.

For each time interval, we compute the number of taxis picking up or dropping off a passenger at every zone, in order to construct the matrix $\mathbf{U}$, and then we run the presented algorithms with a slight modification, since this dataset is special in the following sense: the movement is permanent at every zone, there are taxis picking up or dropping off passengers at every time. And also any path, from any zone to another, is physically possible (unlike the airport data for instance, where there might be no route from one airport to another).

These features significantly increase the difficulty of the problem. For example, a solution where every taxi stays at one the zone, making trips between zones less common, is a local minimum of (2.11). Obviously, this is an undesirable outcome. To correct this artifacts, we added a term to penalize the ℓ_2 norm of vector $\mathbf{a}$ (due to the constraint, the penalty term is written in terms of $\mathbf{1} - \mathbf{MPq}$), which contains the loop probabilities. This penalization does not add any difficulty, and the optimization is virtually the same.

The results are shown in Figure 2.15. The ground truth graph is computed with the complete dataset (taking into account which taxi went from which to which region) by simply counting the number of trips between regions, and then computing directly the probability. The inferred mobility pattern is very similar to the ground truth. The estimated graph has some extra arrows, but with the exception of one of them, the width of the arrows is the thinnest in the figure, which means that the associated probability is not significant.

In some cases where a path is present in both graphs, the width of the arrow may differ. However, the general “large scale” pattern is the same: most of the trips are from one region to an adjacent region (observe that the formulation does not include any geographical information, nor any relation between the regions).

In order to compare directly the results, we also include a radar plot of the ground truth and inferred probabilities, shown in Figure 2.16.

2.4.5 Conclusions of the section and future work

In this section we introduced a framework to address the problem of mobility graph estimation, when only counting information on some nodes is available, the movements are asynchronous, and the time it takes to an entity to go from one site to another depends on the origin and destination. Due to these characteristics, this is a very challenging problem.

We introduced the problem, and presented a formulation based on the dynamics of the system, and a probabilistic approach on the behavior of the entities. The proposed formulation leads to an optimization problem. In this problem, the cost function to be minimized incorporates a data fidelity term, and a penalty term to regularize the solution and avoid ill-posedness, promoting at the same time

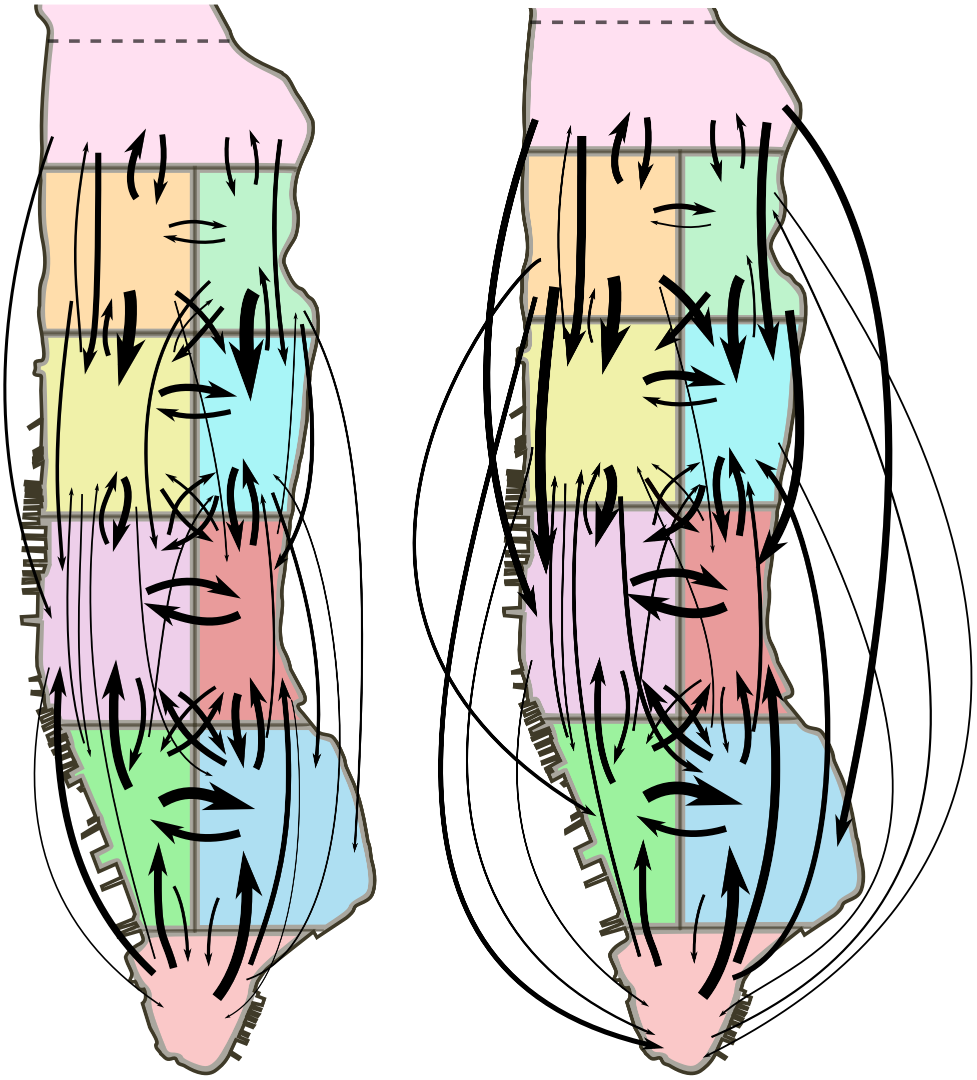


Figure 2.15: Results of the graph inference for the New York taxis dataset. Left: ground truth. Right: inferred graph. The width of the arrows is proportional to the probability q_k of the corresponding path.

sparsity and coherence between the unknowns.

Since the fitting term of this formulation is non-convex, a smart initialization is needed in order to converge to a good local minimum. We propose a second formulation, based only on the mobility and counting information, without taking into account the probabilistic part. This formulation lies on a differential analysis of the events, and also makes use of sparsity promoting terms in order to obtain a reasonable solution of this ill-posed problem.

We solved the proposed optimization problem for two real datatets, which are publicly available: the New York taxis dataset, and the domestic US flights. The results show that the general topology of the mobility pattern can be recovered, and therefore the system can be analyzed from this inferred network. This suggests

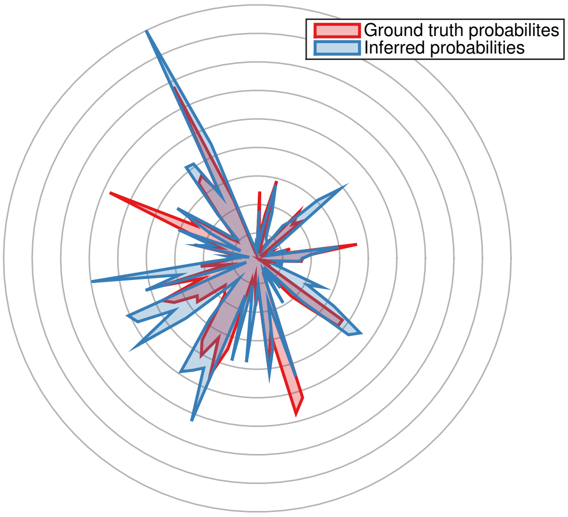


Figure 2.16: Radar plot of the probabilities (vector $\mathbf{q}$) for the New York traffic dataset. In red, the ground truth probabilities, and in blue, the estimated probabilities.

that this is a promising line of work for a very interesting and challenging problem, which can be further improved.

On the other hand, several related applications arise from the presentation of this problem. For instance, given a system, it would be interesting to analyze its mobility network at different times of the day, therefore being able to distinguish between several patterns of behavior.

Another interesting extension is to detect outliers: when the system is stable and the mobility pattern is already learned, a suspicious behavior might be reflected in a sudden increase of the cost function.

This page was intentionally left blank.

Chapter 3

Graph Matching Algorithms

3.1 Introduction

Recently, the problem of matching two graphs has received attention from several scientific communities: from signal processing, computer vision and computer science, to pure mathematics, researchers are using graph matching for important applications, creating new algorithms to solve the problem, and finding more theoretical connections and guarantees for correct recovery of some graph matching methods.

The applications range from pattern recognition [14, 19], computer vision [23, 99, 104], and machine learning [27, 57], to neuro-biology [95], among others.

The particular interest in this problem, both algorithmic and theoretical, comes from its inherent complexity. The cost of searching among all the possible permutations grows exponentially with the number of nodes, and hence becomes intractable even for small graphs. Polynomial time algorithms are known only for a few classes of graphs (e.g., trees [88, 90]; planar graphs [55]; and graphs with some particular spectral properties [1, 43]), and therefore the general graph matching problem is still a significant challenge.

This chapter is deals with graph matching algorithms. In Section 3.2 we present a technique based on ideas borrowed from the sparse modeling community, more specifically the Group Lasso. This way, we formulate the graph matching problem trying to match the supports of the adjacency matrices, instead of a general average error cost.

In Section 3.3 we first state a theorem about the success probability of some common relaxations, and then we present an exhaustive experimental analysis of several methods, including a new combination of algorithm-initialization that is motivated by the theorem.

3.2 Robust Multimodal Graph Matching

Section summary

Graph matching is a challenging problem with very important applications in a wide range of fields, from image and video analysis to biological and biomedical problems. We propose a robust graph matching algorithm inspired in sparsity-related techniques. We cast the problem, resembling group or collaborative sparsity formulations, as a non-smooth convex optimization problem that can be efficiently solved using augmented Lagrangian techniques. The method can deal with weighted or unweighted graphs, as well as multimodal data, where different graphs represent different types of data. The proposed approach is also naturally integrated with collaborative graph inference techniques, solving general network inference problems where the observed variables, possibly coming from different modalities, are not in correspondence. The algorithm is tested and compared with state-of-the-art graph matching techniques in both synthetic and real graphs. We also present results on multimodal graphs and applications to collaborative inference of brain connectivity from alignment-free functional magnetic resonance imaging (fMRI) data. The code is publicly available.

3.2.1 Introduction

Problems related to graph isomorphisms have been an important and enjoyable challenge for the scientific community for a long time. The graph isomorphism problem itself consists in determining whether two given graphs are isomorphic or not, that is, if there exists an edge preserving bijection between the vertex sets of the graphs. This problem is also very interesting from the computational complexity point of view, since its complexity level is still unsolved: it is one of the few problems in NP not yet classified as P nor NP-complete [25]. The graph isomorphism problem is contained in the (harder) graph matching problem, which consists in finding the exact isomorphism between two graphs. Graph matching is therefore a very challenging problem which has several applications, e.g., in the pattern recognition and computer vision areas. In this section we address the problem of (potentially multimodal) graph matching when the graphs are not exactly isomorphic. This is by far the most common scenario in real applications, since the graphs to be compared are the result of a measuring or description process, which is naturally affected by noise.

Given two graphs G_A and G_B with p vertices, which we will characterize in terms of their $p \times p$ adjacency matrices $\mathbf{A}$ and $\mathbf{B}$, the graph matching problem consists in finding a correspondence between the nodes of G_A and G_B minimizing some matching error. In terms of the adjacency matrices, this corresponds to finding a matrix $\mathbf{P}$ in the set of permutation matrices $\mathcal{P}$, such that it minimizes some distance between $\mathbf{A}$ and $\mathbf{PBP}^T$. A common choice is the Frobenius norm $||\mathbf{A} - \mathbf{PBP}^T||_F^2$, where $||\mathbf{M}||_F^2 = \sum_{ij} \mathbf{M}_{ij}^2$. The graph matching problem can be then stated as

$$\min_{\mathbf{P} \in \mathcal{P}} ||\mathbf{A} - \mathbf{PBP}^T||_F^2 = \min_{\mathbf{P} \in \mathcal{P}} ||\mathbf{AP} - \mathbf{PB}||_F^2. \tag{3.1}$$

The combinatorial nature of the permutation search makes this problem NP in general, although polynomial algorithms have been developed for a few special types of graphs, like trees or planar graphs for example [25].

There are several and diverse techniques addressing the graph matching problem, including spectral methods [92] and problem relaxations [4, 95, 102]. A good review of the most common approaches can be found in [25]. In this section we focus on relaxation techniques for solving an approximate version of the problem. Maybe the simplest one is to relax the feasible set (the permutation matrices) to its convex hull, the set of doubly stochastic matrices $\mathcal{D}$, which consist of the matrices with non-negative entries such that each row and column sum up one: $\mathcal{D} = \{\mathbf{M} \in \mathbb{R}^{p \times p} : \mathbf{M}_{ij} \geq 0, \mathbf{M}\mathbf{1} = \mathbf{1}, \mathbf{M}^T\mathbf{1} = \mathbf{1}\}$, $\mathbf{1}$ being the p-dimensional vector of ones. The relaxed version of the problem is

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P} \in \mathcal{D}} ||\mathbf{AP} - \mathbf{PB}||_F^2,$$

which is a convex problem, though the result is a doubly stochastic matrix instead of a permutation. The final node correspondence is obtained as the closest permutation matrix to $\hat{\mathbf{P}}$: $\mathbf{P}^* = \arg\min_{\mathbf{P} \in \mathcal{P}} ||\mathbf{P} - \hat{\mathbf{P}}||_F^2$, which is a linear assignment problem that can be solved in $O(p^3)$ by the Hungarian algorithm [62]. However, this last step lacks any guarantee about the graph matching problem itself. This approach will be referred to as QCP for *quadratic convex problem*.

One of the newest approximate methods is the PATH algorithm in [102], which combines this convex relaxation with a concave relaxation. Another new technique is the FAQ method in [95], which solves a relaxed version of the Quadratic Assignment Problem. We compare the method here proposed to all these techniques in the experimental section.

The main contributions of this section are two-fold. Firstly, we propose a new and versatile formulation for the graph matching problem which is more robust to noise and can naturally manage multimodal data. The technique, which we call GLAG for Group lasso graph matching, is inspired by the recent works on sparse modeling, and in particular group and collaborative sparse coding. We present several experimental evaluations to back up these claims. Secondly, this proposed formulation fits very naturally into the alignment-free collaborative network inference problem, where we collaborative exploit non-aligned (possibly multimodal) data to infer the underlying common network, making this application never addressed before to the best of our knowledge. We assess this with experiments using real fMRI data.

The rest of this section is organized as follows. In Section 3.2.2 we present the proposed graph matching formulation, and we show how to solve the optimization problem in Section 3.2.3. The joint collaborative network and permutation learning application is described in Section 3.2.4. Experimental results are presented in Section 3.2.5, and we conclude in Section 3.2.10.

3.2.2 Graph matching formulation

We consider the problem of matching two graphs that are not necessarily perfectly isomorphic. We will assume the following model: Assume that we have a noise free graph characterized by an adjacency matrix $\mathbf{T}$. Then we want to match two graphs with adjacency matrices $\mathbf{A} = \mathbf{T} + \mathbf{O_A}$ and $\mathbf{B} = \mathbf{P_o^T T P_o} + \mathbf{O_B}$, where $\mathbf{O_A}$ and $\mathbf{O_B}$ have a sparse number of non-zero elements of arbitrary magnitude. This realistic model is often used in experimental settings, e.g., [102].

In this context, the QCP formulation tends to find a doubly stochastic matrix $\mathbf{P}$ which minimizes the "average error" between $\mathbf{AP}$ and $\mathbf{PB}$. However, these spurious mismatching edges can be thought of as outliers, so we would want a metric promoting that $\mathbf{AP}$ and $\mathbf{PB}$ share the same active set (non zero entries representing edges), with the exception of some sparse entries. This can be formulated in terms of the group Lasso penalization [100]. In short, the group Lasso takes a set of groups of coefficients and promotes that only some of these groups are active, while the others remain zero. Moreover, the usual behavior is that when a group is active, all the coefficients in the group are non-zero. In this particular graph matching application, we form p^2 groups, one per matrix entry (i, j), each one consisting of the 2-dimensional vector $\big((\mathbf{AP})_{ij}, (\mathbf{PB})_{ij}\big)$. The proposed cost function is then the sum of the l_2 norms of the groups:

$$f(P) = \sum_{i,j} \big|\big|\big((\mathbf{AP})_{ij}, (\mathbf{PB})_{ij}\big)\big|\big|_2 \ . \tag{3.2}$$

Ideally we would like to solve the graph matching problem by finding the minimum of f over the set of permutation matrices $\mathcal{P}$. Of course this formulation is still computationally intractable, so we solve the relaxed version, changing $\mathcal{P}$ by its convex hull $\mathcal{D}$, resulting in the convex problem

$$\tilde{\mathbf{P}} = \arg \min_{\mathbf{P} \in \mathcal{D}} f(\mathbf{P}). \tag{3.3}$$

As with the Frobenius formulation, the final step simply finds the closest permutation matrix to $\tilde{\mathbf{P}}$.

Let us analyze the case when $\mathbf{A}$ and $\mathbf{B}$ are the adjacency matrices of two isomorphic undirected unweighted graphs with e edges and no self-loops. Since the graphs are isomorphic, there exist a permutation matrix $\mathbf{P_o}$ such that $\mathbf{A} = \mathbf{P_o B P_o^T}$.

Lemma 1. *Under the conditions stated above, the minimum value of the optimization problem* (3.3) *is* $2\sqrt{2}e$ *and it is reached by* $\mathbf{P_o}$*, although the solution is not unique in general. Moreover, any solution* $\mathbf{P}$ *of problem* (3.3) *satisfies* $\mathbf{AP} = \mathbf{PB}$.

Proof: Let $(a)_k$ denote all the p^2 entries of $\mathbf{AP}$, and $(b)_k$ all the entries of $\mathbf{PB}$. Then $f(\mathbf{P})$ can be re-written as $f(\mathbf{P}) = \sum_k \sqrt{a_k^2 + b_k^2}$.

Observing that $\sqrt{a^2 + b^2} \geq \frac{\sqrt{2}}{2}(a + b)$, we have

$$f(P) = \sum_k \sqrt{a_k^2 + b_k^2} \geq \sum_k \frac{\sqrt{2}}{2}(a_k + b_k) \ . \tag{3.4}$$

Now, since $\mathbf{P}$ is doubly stochastic, the sum of all the entries of $\mathbf{AP}$ is equal to the sum of all the entries of $\mathbf{A}$, which is two times the number of edges. Therefore $\sum_k a_k = \sum_k b_k = 2e$ and $f(\mathbf{P}) \geq 2\sqrt{2}e$.

The equality in (3.4) holds if and only if $a_k = b_k$ for all k, which means that $\mathbf{AP} = \mathbf{PB}$. In particular, this is true for the permutation $\mathbf{P_o}$, which completes the proof of all the statements. □

This Lemma shows that the fact that the weights in $\mathbf{A}$ and $\mathbf{B}$ are not compared in magnitude does not affect the matching performance when the two graphs are isomorphic and have equal weights. On the other hand, this property places a fundamental role when moving away from this setting. Indeed, since the group lasso tends to set complete groups to zero, and the actual value of the non-zero coefficients is less important, this allows to group very dissimilar coefficients together, if that would result in fewer active groups. This is even more evident when using the l_∞ norm instead of the l_2 norm of the groups, and the optimization remains very similar to the one presented below. Moreover, the formulation remains valid when both graphs come from different modalities, a fundamental property when for example addressing alignment-free collaborative graph inference as presented in Section 3.2.4 (the elegance with which this graph matching formulation fits into such problem will be further stressed there). In contrast, the Frobenious-based approaches mentioned in the introduction are very susceptible to differences in edge magnitudes and not appropriate for multimodal matching[1].

3.2.3 Optimization

The proposed minimization problem (3.3) is convex but non-differentiable. Here we use an efficient variant of the Alternating Direction Method of Multipliers (ADMM) [15]. The idea is to write the optimization problem as an equivalent artificially constrained problem, using two new variables $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{p \times p}$:

$$\min_{\mathbf{P} \in \mathcal{D}} \sum_{i,j} ||(\boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}_{ij})||_2 \qquad s.t. \quad \boldsymbol{\alpha} = \mathbf{AP}, \quad \boldsymbol{\beta} = \mathbf{PB}. \tag{3.5}$$

The ADMOM method generates a sequence which converges to the minimum of the augmented Lagrangian of the problem:

$$L(\mathbf{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{U}, \mathbf{V}) = \sum_{i,j} ||(\boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}_{ij})||_2 + \frac{c}{2}||\boldsymbol{\alpha} - \mathbf{AP} + \mathbf{U}||^2 + \frac{c}{2}||\boldsymbol{\beta} - \mathbf{PB} + \mathbf{V}||^2 ,$$

where $\mathbf{U}$ and $\mathbf{V}$ are related to the Lagrange multipliers and c is a fixed constant.

The decoupling produced by the new artificial variables allows to update their values one at a time, minimizing the augmented Lagrangian L. We first update the pair $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ while keeping fixed $(\mathbf{P}, \mathbf{U}, \mathbf{V})$; then we minimize for $\mathbf{P}$; and finally update $\mathbf{U}$ and $\mathbf{V}$, as described next in Algorithm 1.

[1] If both graphs are binary and we limit to permutation matrices (for which there are no algorithms known to find the solution in polynomial time), then the minimizers of (2) and (1) are the same (Vince Lyzinski, personal communication).

```
Input  : Adjacency matrices $\mathbf{A}, \mathbf{B}$, $c > 0$.
Output: Permutation matrix $\mathbf{P^*}$
Initialize $\mathbf{U} = \mathbf{0}$, $\mathbf{V} = \mathbf{0}$, $\mathbf{P} = \frac{1}{p}\mathbf{1}^T\mathbf{1}$
while stopping criterion is not satisfied do
    $(\boldsymbol{\alpha}^{t+1}, \boldsymbol{\beta}^{t+1}) =$
    $\arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{i,j} ||\big(\boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}_{ij}\big)||_2 + \frac{c}{2}||\boldsymbol{\alpha} - \mathbf{AP}^t + \mathbf{U}^t||_F^2 + \frac{c}{2}||\boldsymbol{\beta} - \mathbf{P}^t\mathbf{B} + \mathbf{V}^t||_F^2$
    $\mathbf{P}^{t+1} = \arg\min_{\mathbf{P}\in\mathcal{D}} \ \frac{1}{2}||\boldsymbol{\alpha}^{t+1} - \mathbf{AP} + \mathbf{U}^t||_F^2 + \frac{1}{2}||\boldsymbol{\beta}^{t+1} - \mathbf{PB} + \mathbf{V}^t||_F^2$
    $\mathbf{U}^{t+1} = \mathbf{U}^t + \boldsymbol{\alpha}^{t+1} - \mathbf{AP}^{t+1}$
    $\mathbf{V}^{t+1} = \mathbf{V}^t + \boldsymbol{\beta}^{t+1} - \mathbf{P}^{t+1}\mathbf{B}$
end
$\mathbf{P^*} = \text{argmin}_{\mathbf{Q}\in\mathcal{P}} ||\mathbf{Q} - \mathbf{P}||_F^2$
```

Algorithm 1: Robust graph matching algorithm. See text for implementation details of each step.

The first subproblem is decomposable into p^2 scalar problems (one for each matrix entry),

$$\min_{\boldsymbol{\alpha}_{ij},\boldsymbol{\beta}_{ij}} ||\big(\boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}_{ij}\big)||_2 + \frac{c}{2}(\boldsymbol{\alpha}_{ij} - (\mathbf{AP}^t)_{ij} + \mathbf{U}^t_{ij})^2 + \frac{c}{2}(\boldsymbol{\beta}_{ij} - (\mathbf{P}^t\mathbf{B})_{ij} + \mathbf{V}^t_{ij})^2.$$

From the optimality conditions on the subgradient of this subproblem, it can be seen that this can be solved in closed form by means of the well know vector soft-thresholding operator [100]: $S_v(\mathbf{b}, \lambda) = \left[1 - \frac{\lambda}{||\mathbf{b}||_2}\right]_+ \mathbf{b}$.

The second subproblem is a minimization of a convex differentiable function over a convex set, so general solvers can be chosen for this task. For instance, a projected gradient descent method can be used. However, this would require to compute several projections onto $\mathcal{D}$ per iteration, which is one of the computationally most expensive steps. Nevertheless, we can choose to solve a linearized version of the problem while keeping the convergence guarantees of the algorithm [64]. In this case, the linear approximation of the first term is:

$$\frac{1}{2}||\boldsymbol{\alpha}^{t+1} - \mathbf{AP} + \mathbf{U}^t||_F^2 \approx \frac{1}{2}||\boldsymbol{\alpha}^{t+1} - \mathbf{AP}^k + \mathbf{U}^t||_F^2 + \langle \mathbf{g}^k, \mathbf{P} - \mathbf{P}^k \rangle + \frac{1}{2\tau}||\mathbf{P} - \mathbf{P}^k||_F^2,$$

where $\mathbf{g}^k = -\mathbf{A^T}(\boldsymbol{\alpha}^{t+1} + \mathbf{U}^t - \mathbf{AP}^k)$ is the gradient of the linearized term, $\langle \cdot, \cdot \rangle$ is the usual inner product of matrices, and τ is any constant such that $\tau < \frac{1}{\rho(\mathbf{A^T A})}$, with $\rho(\cdot)$ being the spectral norm.

The second term can be linearized analogously, so the minimization of the second step becomes

$$\min_{\mathbf{P}\in\mathcal{D}} \frac{1}{2}||\mathbf{P} - \underbrace{\big(\mathbf{P}^k + \tau\mathbf{A^T}(\boldsymbol{\alpha}^{t+1} + \mathbf{U}^t - \mathbf{AP}^k)\big)}_{\text{fixed matrix } \mathbf{C}}||_F^2 + \frac{1}{2}||\mathbf{P} - \underbrace{\big(\mathbf{P}^k + \tau(\boldsymbol{\beta}^{t+1} + \mathbf{V}^t - \mathbf{P}^k\mathbf{B})\mathbf{B^T}\big)}_{\text{fixed matrix } \mathbf{D}}||_F^2$$

which is simply the projection of the matrix $\frac{1}{2}(\mathbf{C} + \mathbf{D})$ over $\mathcal{D}$.

Summarizing, each iteration consists of p^2 vector thresholdings when solving for $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, one projection over $\mathcal{D}$ when solving for $\mathbf{P}$, and two matrix multiplications for the update of $\mathbf{U}$ and $\mathbf{V}$. The code is publicly available at `www.fing.edu.uy/~mfiori`.

3.2.4 Application to joint graph inference of not pre-aligned data

Estimating the inverse covariance matrix is a very active field of research. In particular the inference of the support of this matrix, since the non-zero entries have information about the conditional dependence between variables. In numerous applications, this matrix is known to be sparse, and in this regard the graphical Lasso has proven to be a good estimator for the inverse covariance matrix [40, 101] (also for non-Gaussian data [66]). Assume that we have a p-dimensional multivariate normal distributed variable $X \sim \mathcal{N}(0, \Sigma)$; let $\mathbf{X} \in \mathbb{R}^{k \times p}$ be a data matrix containing k independent observations of X, and $\mathbf{S}$ its empirical covariance matrix. The graphical Lasso estimator for $\mathbf{\Sigma}^{-1}$ is the matrix $\mathbf{\Theta}$ which solves the optimization problem

$$\min_{\mathbf{\Theta} \succ 0} \operatorname{tr}(\mathbf{S}\mathbf{\Theta}) - \log\det \mathbf{\Theta} + \lambda \sum_{i,j} |\mathbf{\Theta}_{ij}| \ , \tag{3.6}$$

which corresponds to the maximum likelihood estimator for $\mathbf{\Sigma}^{-1}$ with an l_1 regularization.

Collaborative network inference has gained a lot of attention in the last years [22], specially with fMRI data, e.g., [93]. This problem consist of estimating two (or more) matrices $\mathbf{\Sigma}_A^{-1}$ and $\mathbf{\Sigma}_B^{-1}$ from data matrices $\mathbf{X}_A$ and $\mathbf{X}_B$ as above, with the additional prior information that the inverse covariance matrices share the same support. The joint estimation of $\mathbf{\Theta}^A$ and $\mathbf{\Theta}^B$ is performed by solving

$$\min_{\mathbf{\Theta}^A \succ 0, \mathbf{\Theta}^B \succ 0} \operatorname{tr}(\mathbf{S}^A\mathbf{\Theta}^A) - \log\det \mathbf{\Theta}^A + \operatorname{tr}(\mathbf{S}^B\mathbf{\Theta}^B) - \log\det \mathbf{\Theta}^B + \lambda \sum_{i,j} ||\big(\mathbf{\Theta}^A_{ij}, \mathbf{\Theta}^B_{ij}\big)||_2 \ , \tag{3.7}$$

where the first four terms correspond to the maximum likelihood estimators for $\mathbf{\Theta}^A, \mathbf{\Theta}^B$, and the last term is the group Lasso penalty which promotes that $\mathbf{\Theta}^A$ and $\mathbf{\Theta}^B$ have the same active set.

This formulation relies on the limiting underlying assumption that the variables in both datasets (the columns of $\mathbf{X}_A$ and $\mathbf{X}_B$) are in correspondence, i.e., the graphs determined by the adjacency matrices $\mathbf{\Theta}^A$ and $\mathbf{\Theta}^B$ are aligned. However, this is in general not the case in practice. Motivated by the formulation presented in Section 3.2.2, we propose to overcome this limitation by incorporating a permutation matrix into the optimization problem, and jointly learn it on the estimation process. The proposed optimization problem is then given by

$$\min_{\substack{\mathbf{\Theta}^A, \mathbf{\Theta}^B \succ 0 \\ \mathbf{P} \in \mathcal{P}}} \operatorname{tr}(\mathbf{S}^A\mathbf{\Theta}^A) - \log\det \mathbf{\Theta}^A + \operatorname{tr}(\mathbf{S}^B\mathbf{\Theta}^B) - \log\det \mathbf{\Theta}^B + \lambda \sum_{i,j} ||\big((\mathbf{\Theta}^A\mathbf{P})_{ij}, (\mathbf{P}\mathbf{\Theta}^B)_{ij}\big)||_2 . \tag{3.8}$$

Even after the relaxation of the constraint $\mathbf{P} \in \mathcal{P}$ to $\mathbf{P} \in \mathcal{D}$, the joint minimization of (3.8) over $(\mathbf{\Theta}^A, \mathbf{\Theta}^B)$ and $\mathbf{P}$ is a non-convex problem. However it is convex when minimized only over $(\mathbf{\Theta}^A, \mathbf{\Theta}^B)$ or $\mathbf{P}$ leaving the other fixed. Problem (3.8) can be then minimized using a block-coordinate descent type of approach, iteratively minimizing over $(\mathbf{\Theta}^A, \mathbf{\Theta}^B)$ and $\mathbf{P}$.

The first subproblem (solving (3.8) with $\mathbf{P}$ fixed) is a very simple variant of (3.7), which can be solved very efficiently by means of iterative thresholding algorithms [38]. In the second subproblem, since $(\mathbf{\Theta}^A, \mathbf{\Theta}^B)$ are fixed, the only term to minimize is the last one, which corresponds to the graph matching formulation presented in Section 3.2.2.

3.2.5 Experimental results

We now present the performance of our algorithm and compare it with the most recent techniques in several scenarios including synthetic and real graphs, multimodal data, and fMRI experiments. In the cases where there is a "ground truth," the performance is measured in terms of the *matching error*, defined as $||\mathbf{A}_o - \mathbf{P}\mathbf{B}_o\mathbf{P}^{\mathbf{T}}||_F^2$, where $\mathbf{P}$ is the obtained permutation matrix and $(\mathbf{A}_o, \mathbf{B}_o)$ are the original adjacency matrices.

3.2.6 Graph matching: Synthetic graphs

We focus here in the traditional graph matching problem of undirected weighted graphs, both with and without noise. More precisely, let $\mathbf{A}_o$ be the adjacency matrix of a random weighted graph and $\mathbf{B}_o$ a permuted version of it, generated with a random permutation matrix $\mathbf{P}_o$, i.e., $\mathbf{B}_o = \mathbf{P}_o^T\mathbf{A}_o\mathbf{P}_o$. We then add a certain number N of random edges to $\mathbf{A}_o$ with the same weight distribution as the original weights, and another N random edges to $\mathbf{B}_o$, and from these noisy versions we try to recover the original matching (or any matching between $\mathbf{A}_o$ and $\mathbf{B}_o$, since it may not be unique).

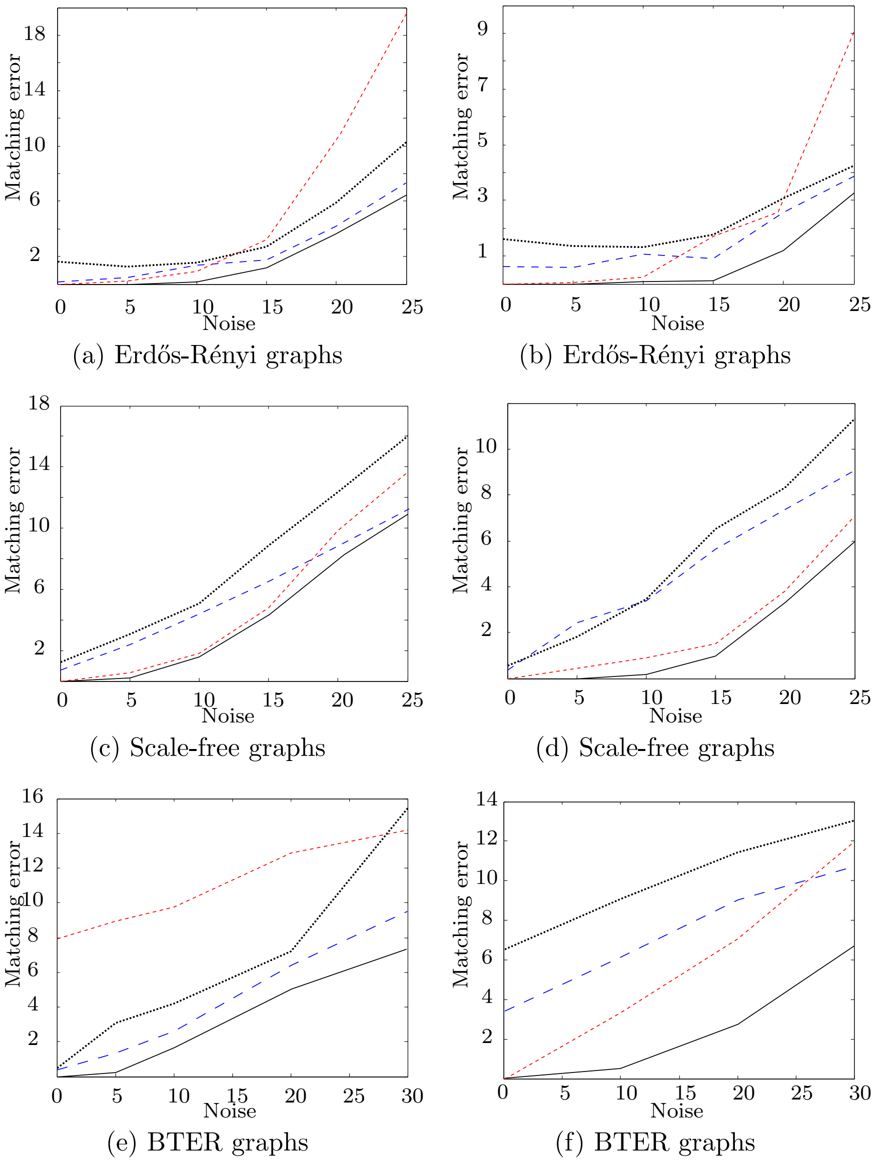
We show the results using three different techniques for the generation of the graphs: the Erdős-Rényi model [35], the model by [11] for scale-free graphs, and graphs with a given degree distribution generated with the BTER algorithm [83]. These models are representative of a wide range of real-world graphs [74]. In the case of the BTER algorithm, the degree distribution was generated according to a geometric law, that is: $\text{Prob}(\text{degree} = t) = (1 - e^{-\mu})e^{\mu t}$.

We compared the performance of our algorithm with the technique by [102] (referred to as PATH), the FAQ method described in [95], and the QCP approach.

Figure 3.1 shows the matching error as a function of the noise level for graphs with $p = 100$ nodes (top row), and for $p = 150$ nodes (bottom row). The number of edges varies between 200 and 400 for graphs with 100 nodes, and between 300 and 600 for graphs with 150 nodes, depending on the model. The performance is averaged over 100 runs. This figure shows that our method is more stable, and consistently outperforms the other methods (considered state-of-the-art), specially for noise levels in the low range (for large noise levels, is not clear what a "true" matching is, and in addition the sparsity hypothesis is no longer valid).

3.2.7 Graph matching: Real graphs

We now present similar experiments to those in the previous section but with real graphs. We use the *C. elegans* connectome. *Caenorhabditis elegans* is an extensively studied roundworm, whose somatic nervous system consists of 279



(a) Erdős-Rényi graphs (b) Erdős-Rényi graphs

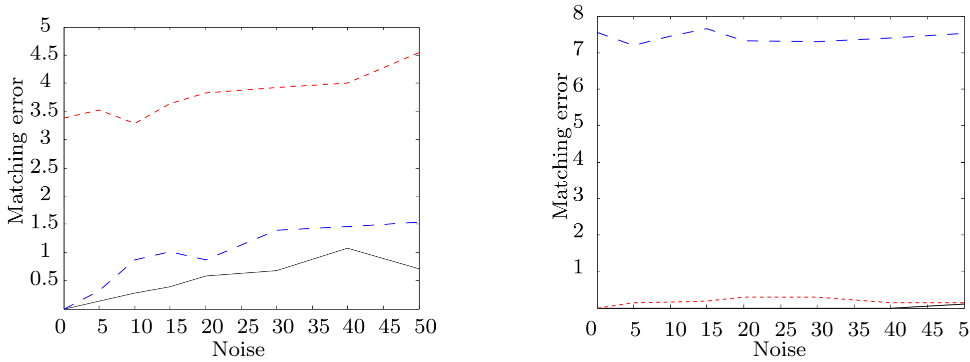
(c) Scale-free graphs (d) Scale-free graphs

(e) BTER graphs (f) BTER graphs

Figure 3.1: Matching error for synthetic graphs with $p = 100$ nodes (left column) and $p = 150$ nodes (right column). In solid black our proposed GLAG algorithm, in long-dashed blue the PATH algorithm, in short-dashed red the FAQ method, and in dotted black the QCP.

neurons that make synapses with other neurons. The two types of connections (chemical and electrical) between these 279 neurons have been mapped [94], and their corresponding adjacency matrices, $\mathbf{A}_c$ and $\mathbf{A}_e$, are publicly available.

We match both the chemical and the electrical connection graphs against noisy artificially permuted versions of them. The permuted graphs are constructed following the same procedure used in Section 3.2.6 for synthetic graphs. The weights of the added noise follow the same distribution as the original weights. The results are shown in Figure 3.2. These results suggest that from the prior art, the PATH algorithm is more suitable for the electrical connection network, while the FAQ algorithm works better for the chemical one. Our method outperforms both of them for both types of connections.



(a) Electrical connection graph

(b) Chemical connection graph

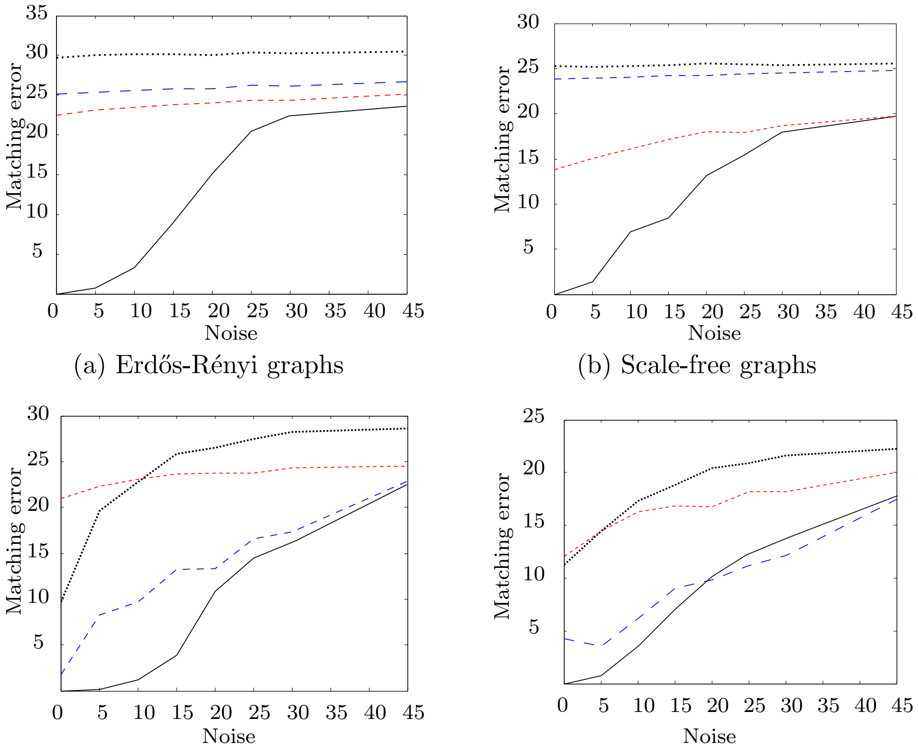
Figure 3.2: Matching error for the C. elegans connectome, averaged over 50 runs. In solid black our proposed GLAG algorithm, in long-dashed blue the PATH algorithm, and in short-dashed red the FAQ method. Note that in the chemical connection graph, the matching error of our algorithm is zero until noise levels of ≈ 50.

3.2.8 Multimodal graph matching

One of the advantages of the proposed approach is its capability to deal with multimodal data. As discussed in Section 3.2.2, the group Lasso type of penalty promotes the supports of $\mathbf{AP}$ and $\mathbf{PB}$ to be identical, almost independently of the actual values of the entries. This allows to match weighted graphs where the weights may follow completely different probability distributions. This is commonly the case when dealing with multimodal data: when a network is measured using significantly different modalities, one expects the underlying connections to be the same but no relation can be assumed between the actual weights of these connections. This is even the case for example for fMRI data when measured with different instruments. In what follows, we evaluate the performance of the proposed method in two examples of multimodal graph matching.

We first generate an auxiliary binary random graph $\mathbf{A}_b$ and a permuted version $\mathbf{B}_b = \mathbf{P}_o^T \mathbf{A}_b \mathbf{P}_o$. Then, we assign weights to the graphs according to distributions p_A and p_B (that will be specified for each experiment), thus obtaining the weighted graphs $\mathbf{A}$ and $\mathbf{B}$. We then add noise consisting of spurious weighted edges following the same distribution as the original graphs (i.e., p_A for $\mathbf{A}$ and p_B for $\mathbf{B}$). Finally, we run all four graph matching methods to recover the permutation. The matching error is measured in the unweighted graphs as $||\mathbf{A}_b - \mathbf{P}\mathbf{B}_b\mathbf{P}^T||_F$. Note that while this metric might not be appropriate for the optimization stage when considering multimodal data, it is appropriate for the actual error evaluation, measuring mismatches. Comparing with the original permutation matrix may not be very informative since there is no guarantee that the matrix is unique, even for the original noise-free data.

Figures 3.3a and 3.3b show the comparison when the weights in both graphs are Gaussian distributed, but with different means and variances. Figures 3.3c and 3.3d show the performances when the weights of $\mathbf{A}$ are Gaussian distributed, and the ones of $\mathbf{B}$ follow a uniform distribution. See captions for details. These results confirm the intuition described above, showing that our method is more suitable for multimodal graphs, specially in the low range of noise.



(a) Erdős-Rényi graphs

(b) Scale-free graphs

(c) Erdős-Rényi graphs

(d) Scale-free graphs

Figure 3.3: Matching error for multimodal graphs with $p = 100$ nodes. In (a) and (b), weights in $\mathbf{A}$ are $\mathcal{N}(1, 0.4)$ and weights in $\mathbf{B}$ are $\mathcal{N}(4, 1)$. In (c) and (d), weights in $\mathbf{A}$ are $\mathcal{N}(1, 0.4)$ and weights in $\mathbf{B}$ are uniform in $[1, 2]$. In solid black our proposed GLAG algorithm, in long-dashed blue the PATH algorithm, in short-dashed red the FAQ method, and in dotted black the QCP.

3.2.9 Collaborative inference

In this last experiment, we illustrate the application of the permuted collaborative graph inference presented in Section 3.2.4 with real resting-state fMRI data, publicly available [77]. We consider here test-retest studies, that is, the same subject undergoing resting-state fMRI in two different sessions separated by a break. Each session consists of almost 10 minutes of data, acquired with a sampling period of $0.645s$, producing about 900 samples per study. The CC200 atlas [28] was used to extract the time-series for the ≈ 200 regions of interest (ROIs), resulting in two data matrices $\mathbf{X}^A, \mathbf{X}^B \in \mathbb{R}^{900\times 200}$, corresponding to test and retest respectively.

To illustrate the potential of the proposed framework, we show that using only part of the data in $\mathbf{X}^A$ and part of the data in a permuted version of $\mathbf{X}^B$, we are able to infer a connectivity matrix almost as accurately as using the whole data. Working with permuted data is very important in this application in order to handle possible miss-alignments to the atlas.

Since there is no ground truth for the connectivity, and as mentioned before the collaborative setting (3.7) has already been proven successful, we take as ground truth the result of the collaborative inference using the empirical covariance matrices of $\mathbf{X}^A$ and $\mathbf{X}^B$, denoted by $\mathbf{S}^A$ and $\mathbf{S}^B$. The result of this collaborative inference procedure are the two inverse covariance matrices $\mathbf{\Theta}^A_{GT}$ and $\mathbf{\Theta}^B_{GT}$. In short, the gold standard built for this experiment is found by solving (obtained with the entire data)

$$\min_{\mathbf{\Theta}^A \succ 0, \mathbf{\Theta}^B \succ 0} \operatorname{tr}(\mathbf{S}^A\mathbf{\Theta}^A) - \log\det\mathbf{\Theta}^A + \operatorname{tr}(\mathbf{S}^B\mathbf{\Theta}^B) - \log\det\mathbf{\Theta}^B + \lambda\sum_{i,j}||(\mathbf{\Theta}^A_{ij}, \mathbf{\Theta}^B_{ij})||_2 \ .$$

Now, let $\mathbf{X}^A_H$ be the first 550 samples of $\mathbf{X}^A$, and $\mathbf{X}^B_H$ the first 550 samples of $\mathbf{X}^B$, which correspond to a little less than 6 minutes of study. We compute the empirical covariance matrices $\mathbf{S}^A_H$ and $\mathbf{S}^B_H$ of these data matrices, and we artificially permute the second one: $\tilde{\mathbf{S}}^B_H = \mathbf{P}_o^T\mathbf{S}^B_H\mathbf{P}_o$. With these two matrices $\mathbf{S}^A_H$ and $\tilde{\mathbf{S}}^B_H$ we run the algorithm described in Section 3.2.4, which alternately computes the inverse covariance matrices $\mathbf{\Theta}^A_H$ and $\mathbf{\Theta}^B_H$ and the matching $\mathbf{P}$ between them.

We compare this approach against the computation of the inverse covariance matrix using only one of the studies. Let $\mathbf{\Theta}^A_s$ and $\mathbf{\Theta}^B_s$ be the results of the graphical Lasso (3.6) using $\mathbf{S}^A$ and $\mathbf{S}^B$:

$$\mathbf{\Theta}^K_s = \operatorname*{argmin}_{\mathbf{\Theta}\succ 0} \ \operatorname{tr}(\mathbf{S}^K\mathbf{\Theta}) - \log\det\mathbf{\Theta} + \lambda\sum_{i,j}|\mathbf{\Theta}_{ij}| \ , \quad \text{for } K = \{A, B\}.$$

This experiment is repeated for 5 subjects in the database. The errors $||\mathbf{\Theta}^A_{GT} - \mathbf{\Theta}^A_s||_F$ and $||\mathbf{\Theta}^A_{GT} - \mathbf{\Theta}^A_H||_F$ are shown in Figure 3.4. The errors for $\mathbf{\Theta}^B$ are very similar. Using less than 6 minutes of each study, with the variables not prealigned, the permuted collaborative inference procedure proposed in Section 3.2.4 outperforms the classical graphical Lasso using the full 10 minutes of study.

3.2.10 Conclusions

We have presented a new formulation for the graph matching problem, and proposed an optimization algorithm for minimizing the corresponding cost function. The reported results show its suitability for the graph matching problem of weighted graphs, outperforming previous state-of-the-art methods, both in synthetic and real graphs. Since in the problem formulation the weights of the graphs are not compared explicitly, the method can deal with multimodal data, outperforming the other compared methods. In addition, the proposed formulation naturally

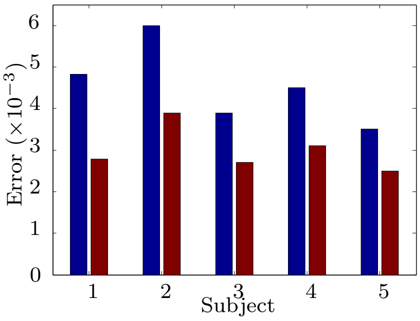


Figure 3.4: Inverse covariance matrix estimation for fMRI data. In blue, error using one complete 10 minutes study: $||\mathbf{\Theta}_{GT}^{A} - \mathbf{\Theta}_{s}^{A}||_F$. In red, error $||\mathbf{\Theta}_{GT}^{A} - \mathbf{\Theta}_{H}^{A}||_F$ with collaborative inference using about 6 minutes of each study, but solving for the unknown node permutations at the same time.

fits into the pre-alignment-free collaborative network inference framework, where the permutation is estimated together with the underlying common network, with promising preliminary results in applications with real data.

3.3 Comparison of Existing Methods With a New One

Section summary

The first part of Chapter 4 is mainly dedicated to prove a theorem connecting the graph matching problem with some relaxations, and giving optimistic and pessimistic results for these relaxations. These theoretical results, which are also stated in this section, suggest that initializing the indefinite algorithm with the convex optimum might yield improved practical performance. Indeed, experimental results illuminate and corroborate these theoretical findings, demonstrating that excellent results are achieved in both benchmark and real data problems by amalgamating the two approaches.

3.3.1 Theoretical and experimental framework

Our theoretical results will be set in the context of correlated random (simple) Bernoulli graphs, which can be used to model many real-data scenarios. Random Bernoulli graphs are the most general edge independent random graphs, and contain many important random graph families including Erdős-Rényi and the widely used stochastic block model of [54].

The random Bernoulli graphs are defined as follows. Given $n \in \mathbb{Z}^+$, a real number $\rho \in [0, 1]$, and a symmetric, hollow matrix $\mathbf{\Lambda} \in [0, 1]^{n \times n}$, define $\mathcal{E} := \{\{i, j\} : i \in [n], j \in [n], i \neq j\}$, where $[n] := \{1, 2, \ldots, n\}$. Two random graphs with respective $n \times n$ adjacency matrices $\mathbf{A}$ and $\mathbf{B}$ are *ρ-correlated Bernoulli*($\mathbf{\Lambda}$) distributed if, for all $\{i, j\} \in \mathcal{E}$, the random variables (matrix entries) $\mathbf{A}_{i,j}, \mathbf{B}_{i,j}$ are Bernoulli($\mathbf{\Lambda}_{i,j}$) distributed, and all of these random variables are collectively independent except that, for each $\{i, j\} \in \mathcal{E}$, the Pearson product-moment correlation coefficient for $\mathbf{A}_{i,j}, \mathbf{B}_{i,j}$ is ρ. It is straightforward to show that the parameters n, ρ, and $\mathbf{\Lambda}$ completely specify the random graph pair distribution, and the distribution may be achieved by first, for all $\{i,j\} \in \mathcal{E}$, having $\mathbf{B}_{ij} \sim$ Bernoulli$(\mathbf{\Lambda}_{i,j})$ independently drawn and then, conditioning on $\mathbf{B}$, have $\mathbf{A}_{i,j} \sim$ Bernoulli $((1-\rho)\mathbf{\Lambda}_{i,j} + \rho\mathbf{B}_{i,j})$ independently drawn. While $\rho = 1$ would imply the graphs are isomorphic, this model allows for a natural vertex alignment (namely the identity function) for $\rho < 1$, i.e. when the graphs are not necessarily isomorphic.

Let us consider a sequence of correlated random Bernoulli graphs for $n = 1, 2, 3, \dots$, where $\mathbf{\Lambda}$ is a function of n. When we say that a sequence of events, $\{E_m\}_{m=1}^{\infty}$, holds *almost always* we mean that almost surely it happens that the events in the sequence occur for all but finitely many m.

The following theorem, which is proved in the next chapter, explores the tradeoff between tractability and correctness when relaxing the graph matching problem.

Theorem 3.1. *Suppose* $\mathbf{A}$ *and* $\mathbf{B}$ *are adjacency matrices for ρ-correlated Bernoulli*$(\mathbf{\Lambda})$ *graphs, and there is an $\alpha \in (0, 1/2)$ such that $\mathbf{\Lambda}_{i,j} \in [\alpha, 1-\alpha]$ for all $i \neq j$. Let $\mathbf{P}^* \in \mathcal{P}$, and denote $\mathbf{A'} := \mathbf{P}^*\mathbf{A}\mathbf{P}^{*\mathbf{T}}$.*
a) If $(1-\alpha)(1-\rho) < 1/2$, then it almost always holds that

$$\arg\min_{\mathbf{D}\in\mathcal{D}} -\langle \mathbf{A'D}, \mathbf{DB} \rangle = \arg\min_{\mathbf{P}\in\mathcal{P}} \|\mathbf{A'} - \mathbf{PBP}^T\|_F = \{\mathbf{P}^*\}.$$

b) If the between graph correlation $\rho < 1$, then it almost always holds that $\mathbf{P}^* \notin \arg\min_{\mathbf{D}\in\mathcal{D}} \|\mathbf{A'D} - \mathbf{DB}\|_F$.

On one hand, we have an optimistic result (Theorem 3.1, part *a)*) about an indefinite relaxation of the graph matching problem. However, since the objective function is nonconvex, there is no efficient algorithm known to exactly solve this relaxation. On the other hand, Theorem 3.1, part *b)*, is a pessimistic result about a commonly used efficiently solvable convex relaxation, which almost always provides an incorrect/non-permutation solution.

After solving (approximately or exactly) the relaxed problem, the solution is commonly projected to the nearest permutation matrix. We have not theoretically addressed this projection step yet. It might be that, even though the solution in $\mathcal{D}$ is not the correct permutation, it is very close to it, and the projection step fixes this. We will show numerically that this is not the case.

We next present simulations that corroborate and illuminate the presented theoretical results, address the projection step, and provide intuition and practical considerations for solving the graph matching problem. Our simulated graphs have $n = 150$ vertices and follow the Bernoulli model described above, where the entries of the matrix Λ are i.i.d. uniformly distributed in $[\alpha, 1-\alpha]$ with $\alpha = 0.1$. In each simulation, we run 100 Monte Carlo replicates for each value of ρ. Note that given this α value, the threshold ρ in order to fulfill the hypothesis of the first part of Theorem 3.1 (namely that $(1-\alpha)(1-\rho) < 1/2$) is $\rho = 0.44$. As in Theorem 3.1, for a fixed $P^* \in \mathcal{P}$, we let $\mathbf{A'} := \mathbf{P}^*\mathbf{A}\mathbf{P}^{*\mathbf{T}}$, so that the correct vertex alignment between $\mathbf{A'}$ and $\mathbf{B}$ is provided by the permutation matrix $\mathbf{P}^*$.

Table 3.1. Notation

Notation	**Algorithm used**	**Ref.**
$\mathbf{D}^* \in \text{argmin}_{\mathbf{D}\in\mathcal{D}}\|\mathbf{A}'\mathbf{D} - \mathbf{DB}\|_F^2$	F-W algorithm run to convergence	[46], [102]
$\mathbf{P_c}$ = projecting $\mathbf{D}^*$ to Π	Hungarian algorithm	[62]
FAQ:$\mathbf{P}^*$	FAQ init. at $\mathbf{P}^*$	[95]
FAQ:$\mathbf{D}^*$	FAQ init. at $\mathbf{D}^*$	[95]
FAQ:$\mathbf{J}$	FAQ init. at $\mathbf{J}$	[95]

We then highlight the applicability of our theory and simulations in a series of real data examples. In the first set of experiments, we match three pairs of graphs with known latent alignment functions. We then explore the applicability of our theory in matching graphs without a pre-specified latent alignment. Specifically, we match 16 benchmark problems (those used in [95, 102]) from the QAPLIB library of [18]. See Section 3.3.3 for more detail. As expected by the theory, in all of our examples a smartly initialized local minimum of the indefinite relaxation achieves best performance.

We summarize the notation we employ in Table 3.1. To find $\mathbf{D}^*$, we employ the F-W algorithm ([46, 102]), run to convergence, to exactly solve the convex relaxation. We also use the Hungarian algorithm ([62]) to compute $\mathbf{P_c}$, the projection of $\mathbf{D}^*$ to $\mathcal{P}$. To find a local minimum of $\min_{\mathbf{D}\in\mathcal{D}} -\langle \mathbf{A}'\mathbf{D}, \mathbf{DB}\rangle$, we use the FAQ algorithm of [95]. We use FAQ:$\mathbf{P}^*$, FAQ:$\mathbf{D}^*$, and FAQ:$\mathbf{J}$ to denote the FAQ algorithm initialized at $\mathbf{P}^*$, $\mathbf{D}^*$, and $\mathbf{J} := \mathbf{1}\cdot\mathbf{1}^T/n$ (the barycenter of $\mathcal{D}$). We compare our results to the GLAG and PATH algorithms, implemented with off-the-shelf code provided by the algorithms' authors. We restrict our focus to these algorithms (indeed, there are a *multitude* of graph matching algorithms present in the literature) as these are the prominent relaxation algorithms; i.e., they all first relax the graph matching problem, solve the relaxation, and then project the solution onto Π.

3.3.2 On the convex relaxed graph matching problem

Theorem 3.1, part *b*, states that we cannot, in general, expect $\mathbf{D}^* = \mathbf{P}^*$. However, $\mathbf{D}^*$ is often projected onto Π, which could potentially recover $\mathbf{P}^*$. Unfortunately, this projection step suffers from the same problems as rounding steps in many integer programming solvers, namely that the distance from the best interior solution to the best feasible solution is not well understood.

In Figure 3.5, we plot $\|\mathbf{A}'\mathbf{D}^* - \mathbf{D}^*\mathbf{B}\|_F^2$ versus the correlation between the random graphs, with 100 replicates per value of ρ. Each experiment produces a pair of dots, either a red/blue pair or a green/grey pair. The energy levels corresponding to the red/green dots correspond to $\|\mathbf{A}'\mathbf{D}^* - \mathbf{D}^*\mathbf{B}\|_F^2$, while the energies corresponding to the blue/grey dots correspond $\|\mathbf{A}'\mathbf{P_c} - \mathbf{P_c}\mathbf{B}\|_F^2$. The colors indicate whether P_c was (green/grey pair) or was not (red/blue pair) $\mathbf{P}^*$.

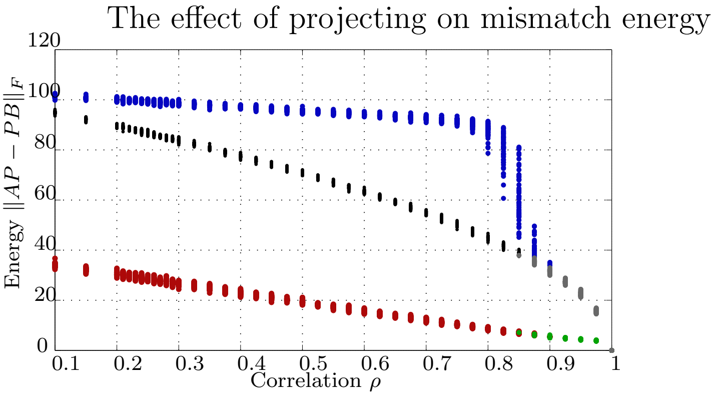


Figure 3.5: For $\rho \in [0.1, 1]$, we plot $\|\mathbf{A'D^*} - \mathbf{D^*B}\|_F^2$ (red /green) and $\|\mathbf{A'P_c} - \mathbf{P_cB}\|_F^2$ (blue/gray). Red/blue dots correspond to simulations where $\mathbf{P_c} \neq \mathbf{P^*}$, and grey/green dots to $\mathbf{P_c} = \mathbf{P^*}$. Black dots correspond to $\|\mathbf{A'P^*} - \mathbf{P^*B}\|_F^2$. For each ρ, we ran 100 MC replicates.

The black dots correspond to the values of $\|\mathbf{A'P^*} - \mathbf{P^*B}\|_F^2$.

Note that, for correlations $\rho < 1$, $\mathbf{D^*} \neq \mathbf{P^*}$, as expected from Theorem 3.1, part *b*. Also note that, even for correlations greater than $\rho = 0.44$, we note $\mathbf{P_c} \neq \mathbf{P^*}$ after projecting to the closest permutation matrix, even though with high probability P^* is the solution to the unrelaxed problem.

We note the large gap between the pre/post projection energy levels when the algorithm fails/succeeds in recovering P^*, the fast decay in this energy (around $\rho \approx 0.8$ in Figure 3.5), and the fact that the value for $\|A'P^* - P^*B\|_F^2$ can be easily predicted from the correlation value. These together suggest that $\|\mathbf{A'P_c} - \mathbf{P_cB}\|_F^2 - \|\mathbf{A'D^*} - \mathbf{D^*B}\|_F^2$ can be used *a posteriori* to assess whether or not graph matching recovered $\mathbf{P^*}$. This is especially true if ρ is known or can be estimated.

How far is $\mathbf{D^*}$ from $\mathbf{P^*}$? When the graphs are isomorphic (i.e., $\rho = 1$ in our setting), then for a large class of graphs, with certain spectral constraints, then $\mathbf{P^*}$ is the unique solution of the convex relaxed graph matching problem [1]. Indeed, in Figure 3.5, when $\rho = 1$ we see that $\mathbf{P^*} = \mathbf{D^*}$ as expected. On the other hand, we know from Theorem 3.1, part *b* that if $\rho < 1$, it is often the case that $\mathbf{D^*} \neq \mathbf{P^*}$. We may think that, via a continuity argument, if the correlation ρ is very close to one, then $\mathbf{D^*}$ will be very close to $\mathbf{P^*}$, and $\mathbf{P_c}$ will probably recover $\mathbf{P^*}$.

We empirically explore this phenomena in Figure 3.6. For $\rho \in [0.1, 1]$, with 100 MC replicates for each ρ, we plot the (Frobenius) distances from $\mathbf{D^*}$ to $\mathbf{P_c}$ (in blue), from $\mathbf{D^*}$ to $\mathbf{P^*}$ (in red), and from $\mathbf{D^*}$ to a uniformly random permutation in Π (in black). Note that all three distances are very similar for $\rho < 0.8$, implying that $\mathbf{D^*}$ is very close to the barycenter and far from the boundary of $\mathcal{D}$. With this in mind, it is not surprising that the projection fails to recover $\mathbf{P^*}$ for $\rho < 0.8$ in Figure 3.5, as at the barycenter, the projection onto Π is uniformly random.

For very high correlation values ($\rho > 0.9$), the distances to $\mathbf{P_c}$ and to $\mathbf{P^*}$ sharply decrease, and the distance to a random permutation sharply increases. This suggests that at these high correlation levels $\mathbf{D^*}$ moves away from the barycenter and towards $\mathbf{P^*}$. Indeed, in Figure 3.5 we see for $\rho > 0.9$ that $\mathbf{P^*}$ is the closest permutation to $\mathbf{D^*}$, and is typically recovered by the projection step.

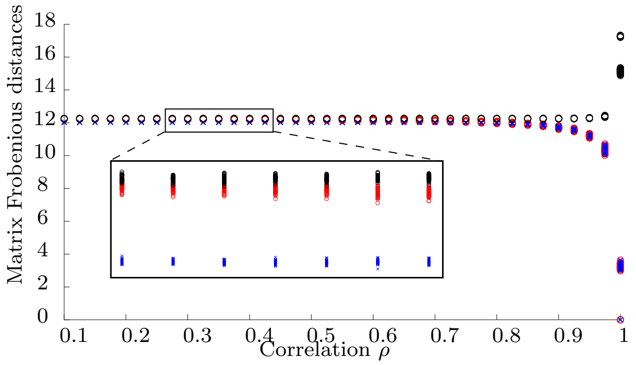


Figure 3.6: Distance from $\mathbf{D}^*$ to $\mathbf{P_c}$ (in blue), to $\mathbf{P}^*$ (in red), and to a random permutation (in black). For each value of ρ, we ran 100 MC replicates.

3.3.3 On indefinite relaxed graph matching problem

The continuous problem one would like to solve, $\min_{\mathbf{D}\in\mathcal{D}} -\langle \mathbf{A'D}, \mathbf{DB}\rangle$ (since its optimum is P^* with high probability), is indefinite. One option is to look for a local minimum of the objective function, as done in the FAQ algorithm of [95]. The FAQ algorithm uses F-W methodology ([46]) to find a local minimum of $-\langle \mathbf{A'D}, \mathbf{DB}\rangle$. Not surprisingly (as there are many local minima), the performance of the algorithm is heavily dependent on the initialization. Below we study the effect of initializing the algorithm at the non-informative barycenter, at $\mathbf{D}^*$ (a principled starting point), and at $\mathbf{P}^*$. We then compare performance of the different FAQ initializations to the PATH algorithm [102] and to the GLAG algorithm [39].

The GLAG algorithm presents an alternate formulation of the graph matching problem. The algorithm convexly relaxes the alternate formulation, solves the relaxation and projects it onto Π. As demonstrated in [39], the algorithm's main advantage is in matching weighted graphs and multimodal graphs. The PATH algorithm begins by finding $\mathbf{D}^*$, and then solves a sequence of concave and convex problems in order to improve the solution. The PATH algorithm can be viewed as an alternative way of projecting $\mathbf{D}^*$ onto Π. Together with FAQ, these algorithms achieve the current best performance in matching a large variety of graphs (see [39], [95], [102]). However, we note that GLAG and PATH often have significantly longer running times than FAQ (even if computing $\mathbf{D}^*$ for FAQ:$\mathbf{D}^*$); see [70, 95].

Figure 3.7 shows the success rate of the graph matching methodologies in recovering $\mathbf{P}^*$. The vertical dashed red line at $\rho = 0.44$ corresponds to the threshold in Theorem 3.1 part a (above which $\mathbf{P}^*$ is optimal whp) for the parameters used in these experiments, and the solid lines correspond to the performance of the different methods: from left to right in gray, FAQ:$\mathbf{P}^*$, FAQ:$\mathbf{D}^*$, FAQ:$\mathbf{J}$; in black, the success rate of P_c; the performance of GLAG and PATH are plotted in blue and red respectively.

Observe that, when initializing with $\mathbf{P}^*$, the fact that FAQ succeeds in recovering $\mathbf{P}^*$ means that $\mathbf{P}^*$ is a local minimum, and the algorithm did not move from the initial point. From the theoretical results, this was expected for $\rho > 0.44$, and the experimental results show that this is also often true for smaller values of ρ.

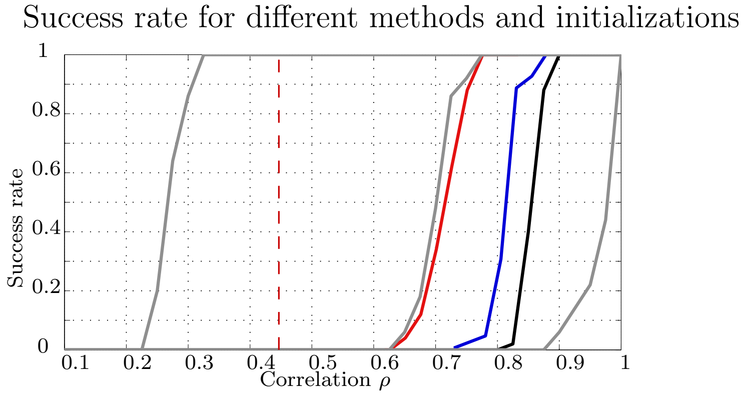


Figure 3.7: Success rate in recovering $\mathbf{P}^*$. In gray, FAQ starting at, from left to right, $\mathbf{P}^*$, $\mathbf{D}^*$, and $\mathbf{J}$; in black, $\mathbf{P_c}$; in red, PATH; in blue, GLAG. For each ρ, we ran 100 MC replicates.

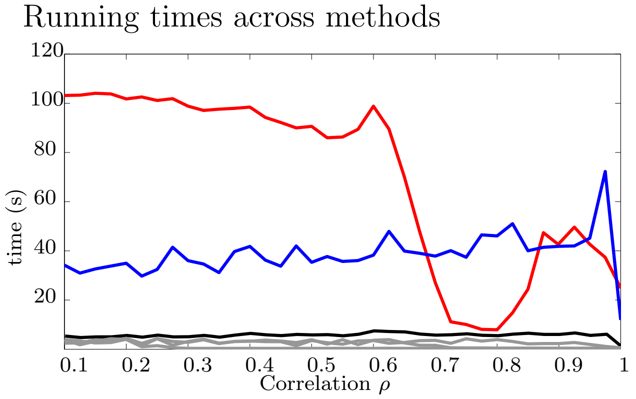


Figure 3.8: Average run time for FAQ:$\mathbf{D}^*$ (note that this does not include the time to find $\mathbf{D}^*$) and FAQ:$\mathbf{J}$ in gray; finding $\mathbf{P_c}$ (first finding $\mathbf{D}^*$) in black; PATH in red; and GLAG in blue. For each ρ, we average over 100 MC replicates. Note that the runtime of PATH drop precipitously at $\rho = 0.6$, which corresponds to the performance increase in Figure 3.7.

However, this only means that $\mathbf{P}^*$ is a local minimum, and the function could have a different global minimum. On the other hand, for very loosely correlated graphs ($\rho < 0.3$), $\mathbf{P}^*$ is not even a local minimum.

The difference in the performance illustrated by the gray lines indicates that the resultant graph matching solution can be improved by using $\mathbf{D}^*$ as an initialization to find a local minimum of the indefinite relaxed problem. We see in the figure that FAQ:$\mathbf{D}^*$ achieves best performance, while being computationally less intensive than PATH and GLAG, see Figure 3.8 for the runtime result. This amalgam of the convex and indefinite methodologies (initialize indefinite with the convex solution) is an important tool for obtaining solutions to graph matching problems, providing a computationally tractable algorithm with state-of-the-art performance.

However, for all the algorithms there is still room for improvement. In these experiments, for $\rho \in [0.44, 0.7]$ theory guarantees that with high probability the global minimum of the indefinite problem is $\mathbf{P}^*$, and we cannot find it with the available methods.

When FAQ:$\mathbf{D}^*$ fails to recover $\mathbf{P}^*$, how close is the objective function at

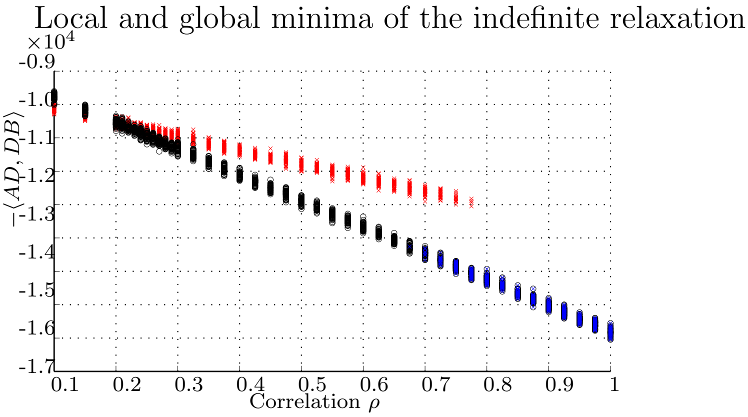


Figure 3.9: Value of $-\langle \mathbf{A}'\mathbf{D}, \mathbf{D}\mathbf{B}\rangle$ for $\mathbf{D} = \mathbf{P}^*$ (black) and for the output of FAQ:$\mathbf{D}^*$ (red/blue indicating failure/success in recovering the true permutation). For each ρ, we ran 100 MC replicates.

the obtained local minima to the objective function at $\mathbf{P}^*$? Figure 3.9 shows $-\langle \mathbf{A}'\mathbf{D}, \mathbf{D}\mathbf{B}\rangle$ for the true permutation, $\mathbf{P}^*$, and for the pre-projection doubly stochastic local minimum found by FAQ:$\mathbf{D}^*$. For $0.35 < \rho < 0.75$, the state-of-the-art algorithm not only fails to recover the correct bijection, but also the value of the objective function is relatively far from the optimal one. There is a transition (around $\rho \approx 0.75$) where the algorithm moves from getting a wrong local minimum to obtaining $\mathbf{P}^*$ (without projection!). For low values of ρ, the objective function values are very close, suggesting that both $\mathbf{P}^*$ and the pre-projection FAQ solution are far from the true global minima. At $\rho \approx 0.3$, we see a separation between the two objective function values (agreeing with the findings in Figure 3.7). As $\rho > 0.44$, we expect that $\mathbf{P}^*$ is the global minima and the pre-projection FAQ solution is far from $\mathbf{P}^*$ until the phase transition at $\rho \approx 0.75$.

Real data experiments

We further demonstrate the applicability of our theory in a series of real data examples. First we match three pairs of graphs where a latent alignment is known. We further compare different graph matching approaches on a set of 16 benchmark problems (those used in [95, 102]) from the QAPLIB QAP library of [18], where no latent alignment is known a priori. Across all of our examples, an intelligently initialized local solution of the indefinite relaxation achieves best performance.

Our first example is from human connectomics. For 45 healthy patients, we have DT-MRI scans from one of two different medical centers: 21 patients scanned (twice) at the Kennedy Krieger Institute (KKI), and 24 patients scanned (once) at the Nathan Kline Institute (NKI) (all data available at `http://openconnecto.me/data/public/MR/MIGRAINE_v1_0/`). Each scan is identically processed via the MIGRAINE pipeline of [53] yielding a 70 vertex weighted symmetric graph. In the graphs, vertices correspond to regions in the Desikan brain atlas, which provides the latent alignment of the vertices. Edge weights count the number of neural fiber bundles connecting the regions. We first average the graphs within each medical

Table 3.2. $\|A'P - PB\|_F$ for the P given by each algorithm together with the number of vertices correctly matched ($n_{corr.}$) in real data experiments

Algorithm		**KKI-NKI**	**Wiki.**	**C. elegans**
Truth	$\|\mathbf{A'P} - \mathbf{PB}\|_F$	82892.87	189.35	155.00
	$n_{corr.}$	70	1381	253
Convex relax.	$\|\mathbf{A'P} - \mathbf{PB}\|_F$	104941.16	225.27	153.38
	$n_{corr.}$	41	97	2
GLAG	$\|\mathbf{A'P} - \mathbf{PB}\|_F$	104721.97	219.98	145.53
	$n_{corr.}$	36	181	4
PATH	$\|\mathbf{A'P} - \mathbf{PB}\|_F$	165626.63	252.55	158.60
	$n_{corr.}$	1	1	1
FAQ:**J**	$\|\mathbf{A'P} - \mathbf{PB}\|_F$	93895.21	205.28	127.55
	$n_{corr.}$	38	30	1
FAQ:D*	$\|\mathbf{A'P} - \mathbf{PB}\|_F$	**83642.64**	**192.11**	**127.50**
	$n_{corr.}$	**63**	**477**	**5**

center and then match the averaged graphs across centers.

For our second example, the graphs consist of the two-hop neighborhoods of the "Algebraic Geometry" page in the French and English Wikipedia graphs. The 1382 vertices correspond to Wikipedia pages with (undirected) edges representing hyperlinks between the pages. Page subject provides the latent alignment function, and to make the graphs of commensurate size we match the intersection graphs.

Lastly, we match the chemical and electrical connectomes of the C. elegans worm. The connectomes consist of 253 vertices, each representing a specific neuron (the same neuron in each graph). Weighted edges representing the strength of the (electrical or chemical) connection between neurons. Additionally, the electrical graph is directed while the chemical graph is not.

The results of these experiments are summarized in Table 3.2. In each example, the computationally inexpensive FAQ:**D*** procedure achieves the best performance compared to the more computationally expensive GLAG and PATH procedures. This reinforces the theoretical and simulation results presented earlier, and again points to the practical utility of our amalgamated approach. While there is a canonical alignment in each example, the results point to the potential use of our proposed procedure (FAQ:**D***) for measuring the strength of this alignment, i.e., measuring the strength of the correlation between the graphs. If the graphs are strongly aligned, as in the KKI-NKI example, the performance of FAQ:**D*** will be close to the truth and a large portion of the latent alignment with be recovered. As the alignment is weaker, FAQ:**D*** will perform even better than the true alignment, and the true alignment will be poorly recovered, as we see in the C. elegans example.

What implications do our results have in graph matching problems without a natural latent alignment? To test this, we matched 16 particularly difficult examples from the QAPLIB library of [18]. We choose these particular examples, Table 3.3. $\|A'P - PB\|_F^2$ for the different tested algorithms on 16 benchmark examples of the QAPLIB library.

QAP	OPT	Convex rel.	GLAG	PATH	Non-Convex. Initialization:	
					Barycenter	Convex sol
chr12c	11156	21142	61430	18048	**13088**	13610
chr15a	9896	41208	78296	19086	29018	**16776**
chr15c	9504	47164	82452	16206	**11936**	18182
chr20b	2298	9912	13728	5560	**2764**	3712
chr22b	6194	10898	21970	8500	8774	**7332**
esc16b	292	314	320	300	314	**292**
rou12	235528	283422	353998	256320	254336	**254302**
rou15	354210	413384	521882	391270	371458	**368606**
rou20	725522	843842	1019622	778284	759838	**754122**
tai10a	135028	175986	218604	152534	157954	**149560**
tai15a	388214	459480	544304	419224	**397376**	397926
tai17a	491812	606834	708754	530978	520754	**516492**
tai20a	703482	810816	1015832	753712	**736140**	756834
tai30a	1818146	2089724	2329604	1903872	1908814	**1858494**
tai35a	2422002	2859448	3083180	2555110	2531558	**2524586**
tai40a	3139370	3727402	4001224	3281830	**3237014**	3299304

because they were previously used in [95, 102] to assess and demonstrate the effectiveness of their respective matching procedures. Results are summarized in Table 3.3. We see that in every example, the indefinite relaxation (suitably initialized) obtains the best possible result. Although there is no latent alignment here, if we view the best possible alignment as the "true" alignment here, then this is indeed suggested by our theory and simulations. As the FAQ procedure is computationally fast (even initializing FAQ at *both* $\mathbf{J}$ and $\mathbf{D}^*$ is often comparatively faster than GLAG and PATH; see [95] and [70]), these results further point to the applicability of our theory. Once again, theory suggests, and experiments confirm, that approximately solving the indefinite relaxation yields the best matching results.

3.3.4 Other random graph models

While the random Bernoulli graph model is the most general edge-independent random graph model, in this section we present analogous experiments for a wider variety of edge-dependent random graph models. For these models, we are unaware of a simple way to exploit pairwise edge correlation in the generation of these graphs, as was present in Section 4.2.1. Here, to simulate aligned non-isomorphic random graphs, we proceed as follows. We generate a graph G_1 from the appropriate underlying distribution, and then model G_2 as an errorful version of G_1; i.e., for each edge in G_1, we randomly flip the edge (i.e., bit-flip from $0 \mapsto 1$ or $1 \mapsto 0$) independently with probability $p \in [0, 1]$. We then graph match G_1 and G_2, and we plot the performance of the algorithms in recovering the latent alignment function across a range of values of p.

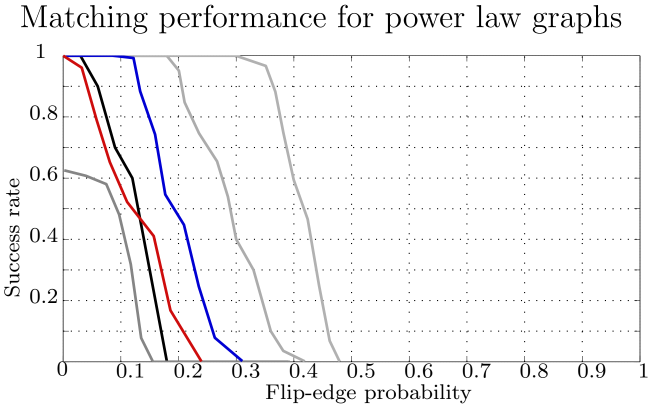


Figure 3.10: Success rate in recovering $\mathbf{P}^*$ for 150 vertex power law graphs with $\beta = 2$ for: In gray, from right to left, FAQ:$\mathbf{P}^*$, FAQ:$\mathbf{D}^*$, and FAQ:$\mathbf{J}$; in black, $\mathbf{P_c}$; in red, PATH; in blue, GLAG. For each value of the bit-flip parameter p, we ran 100 MC replicates.

We first evaluate the performance of our algorithms on *power law* random graphs [11]; these graphs have a degree distribution that follows a power law, i.e., the proportion of vertices of degree d is proportional to $d^{-\beta}$ for some constant $\beta > 0$. These graphs have been used to model many real data networks, from the Internet [3, 36], to social and biological networks [51], to name a few. In general, these graphs have only a few vertices with high degree, and the great majority of the vertices have relatively low degree.

Figure 3.10 shows the performance comparison for the methods analyzed above: FAQ:$\mathbf{P}^*$, FAQ:$\mathbf{D}^*$, FAQ:$\mathbf{J}$, $\mathbf{P_c}$, PATH, and GLAG. For a range of $p \in [0, 1]$, we generated a 150 vertex power law graph with $\beta = 2$, and subsequently graph matched this graph and its errorful version. For each p, we have 100 MC replicates. As with the random Bernoulli graphs, we see from Figure 3.10 that the true permutation is a local minimum of the non-convex formulation for a wide range of flipping probabilities ($p \leq 0.3$), implying that in this range of p, G_1 and G_2 share significant common structure. Across all values of $p < 0.5$, FAQ:$\mathbf{P}^*$ outperforms all other algorithms considered (with FAQ:$\mathbf{D}^*$ being second best across this range). This echoes the results of Sections (3.3.2)–(3.3.3), and suggests an analogue of Theorem 3.1 may hold in the power law setting. We are presently investigating this.

We next evaluate the performance of our algorithms on graphs with bounded maximum degree (also called *bounded valence graphs*). These graphs have been extensively studied in the literature, and for bounded valence graphs, the graph isomorphism problem is in P [68]. For the experiments in this paper we generate a random graph from the model in [8] with maximum degree equal to 4, and vary the graph order from 50 to 350 vertices. Figure 3.11 shows the comparison of the different techniques and initializations for these graphs, across a range of bit-flipping parameters $p \in [0, 1]$.

It can be observed that even for isomorphic graphs ($p = 0$), all but FAQ:$\mathbf{P}^*$ fail to perfectly recover the true alignment. We did not see this phenomena in the other random graph models, and this can be explained as follows. It is a well

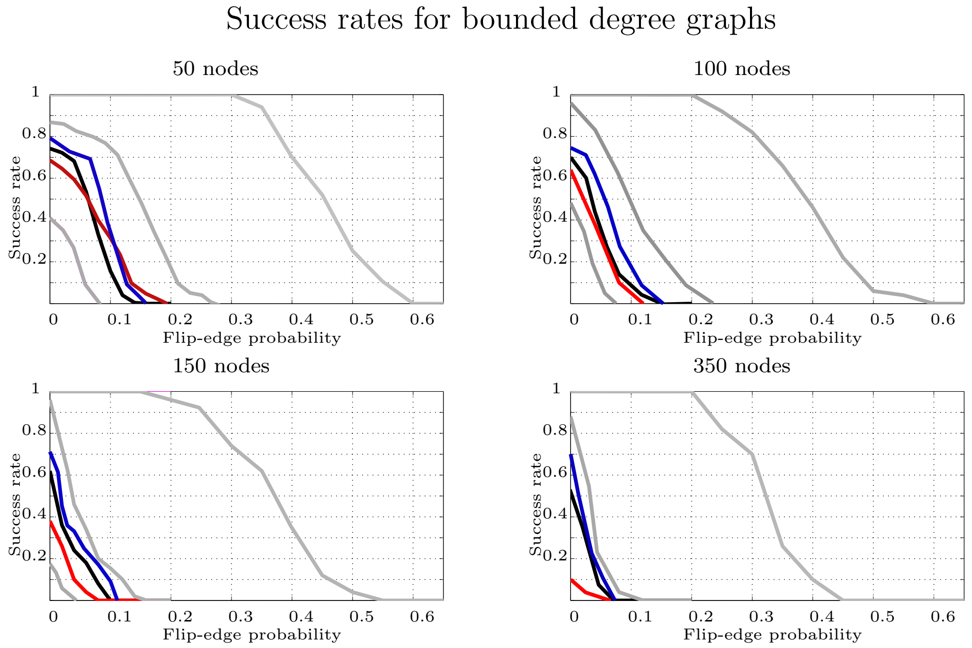


Figure 3.11: Success rate in recovering $\mathbf{P}^*$ for bounded degree graphs (max degree 4). In gray, from right to left, FAQ:$\mathbf{P}^*$, FAQ:$\mathbf{D}^*$, and FAQ:$\mathbf{J}$; in black, $\mathbf{P_c}$; in red, PATH; in blue, GLAG. For each probability we ran 100 MC replicates.

known fact that convex relaxations fail for regular graphs [43], and also that the bounded degree model tends to generate almost regular graphs [61]. Therefore, even without flipped edges, the graph matching problem with the original graphs is very ill-conditioned for relaxation techniques. Nevertheless, the true alignment is a local minimum of the non-convex formulation for a wide range of values of p (shown by FAQ:$\mathbf{P}^*$ performing perfectly over a range of p in Figure 3.11). We again note that FAQ:$\mathbf{D}^*$ outperforms $\mathbf{P_c}$, PATH and GLAG across all graph sizes and bit-flip parameters p. This suggests that a variant of Theorem 3.1 may also hold for bounded valence graphs as well, and we are presently exploring this.

We did not include experiments with any random graph models that are highly regular and symmetric (for example, mesh graphs). Symmetry and regularity have two effects on the graph matching problem. Firstly, it is well known that $\mathbf{P_c} \neq \mathbf{P}^*$ for non-isomorphic regular graphs (indeed, $\mathbf{J}$ is a solution of the convex relaxed graph matching problem). Secondly, the symmetry of these graphs means that there are potentially several isomorphisms between a graph and its vertex permuted analogue. Hence, any flipped edge could make permutations other than $\mathbf{P}^*$ into the minima of the graph matching problem.

3.3.5 Directed graphs

All the theory developed above is proven in the undirected graph setting (i.e., $\mathbf{A}$ and $\mathbf{B}$ are assumed symmetric). However, directed graphs are common in numerous applications. Figure 3.12 repeats the analysis of Figure 3.7 with directed graphs, all other simulation parameters being unchanged. The PATH algorithm is not shown in this new figure because it is designed for undirected graphs, and its performance for directed graphs is very poor. Recall that in Figure 3.7, i.e., in the undirected setting, FAQ:**J** performed significantly worse than $\mathbf{P_c}$. In Figure 3.12, i.e., the directed setting, we note that the performance of FAQ:**J** outperforms $\mathbf{P_c}$ over a range of $\rho \in [0.4, 0.7]$. As in the undirected case, we again see significant performance improvement (over FAQ:**J**, $\mathbf{P_c}$, and GLAG) when starting FAQ from $\mathbf{D}^*$ (the convex solution). Indeed, we suspect that a directed analogue of Theorem 3.1 holds, which would explain the performance increase achieved by the nonconvex relaxation over $\mathbf{P_c}$. Here, we note that the setting for the remainder of the examples considered is the undirected graphs setting.

3.3.6 Seeded graphs

In some applications it is common to have some *a priori* information about partial vertex correspondences, and seeded graph matching includes these known partial matchings as constraints in the optimization (see [1, 44, 71]). However, seeds do more than just reducing the number of unknowns in the alignment of the vertices. Even a few seeds can dramatically increase graph matching performance, and (in the ρ-correlated Erdős-Rényi setting) a logarithmic (in n) number of seeds contain enough signal in their seed–to–nonseed adjacency structure to a.s. perfectly align two graphs [71]. Also, as shown in the deterministic graph setting in [1], very often $\mathbf{D}^*$ is closer to $\mathbf{P}^*$.

In Figure 3.13, the graphs are generated from the ρ-correlated random Bernoulli model with random $\mathbf{\Lambda}$ (entrywise uniform over $[0.1, 0.9]$). We run the Frank-Wolfe method (modified to incorporate the seeds) to solve the convex relaxed graph matching problem, and the method in [44, 71] to approximately solve the nonconvex relaxation, starting from **J**, $\mathbf{D}^*$, and $\mathbf{P}^*$. Note that with seeds, perfect matching is achieved even below the theoretical bound on ρ provided in Theorem 1 (for ensuring $\mathbf{P}^*$ is the global minimizer). This provides a potential way to improve the theoretical bound on ρ in Theorem 3.1, and the extension of Theorem 1 for graphs with seeds is the subject of future research.

With the exception of the nonconvex relaxation starting from $\mathbf{P}^*$, each of the different FAQ initializations and the convex formulation all see significantly improved performance as the number of seeds increases. We also observe that the nonconvex relaxation seems to benefit much more from seeds than the convex relaxation. Indeed, when comparing the performance with no seeds, the $\mathbf{P_c}$ performs better than FAQ:**J**. However, with just five seeds, this behavior is inverted. Also of note, in cases when seeding returns the correct permutation, we've empirically observed that merely initializing the FAQ algorithm with the seeded start, and not enforcing the seeding constraint, also yields the correct permutation as its solution (not shown).

Figure 3.14 shows the running time (to obtain a solution) when starting from $\mathbf{D}^*$ for the nonconvex relaxation, using different numbers of seeds. For a fixed seed level, the running time is remarkably stable across ρ when FAQ does not recover

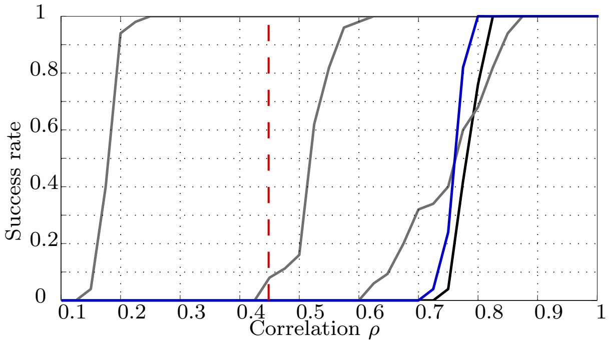


Figure 3.12: Success rate for directed graphs. We plot $\mathbf{P_c}$ (black), the GLAG method (blue), and the nonconvex relaxation starting from different points in green, from right to left: FAQ:$\mathbf{J}$, FAQ:$\mathbf{D}^*$, FAQ:$\mathbf{P}^*$.

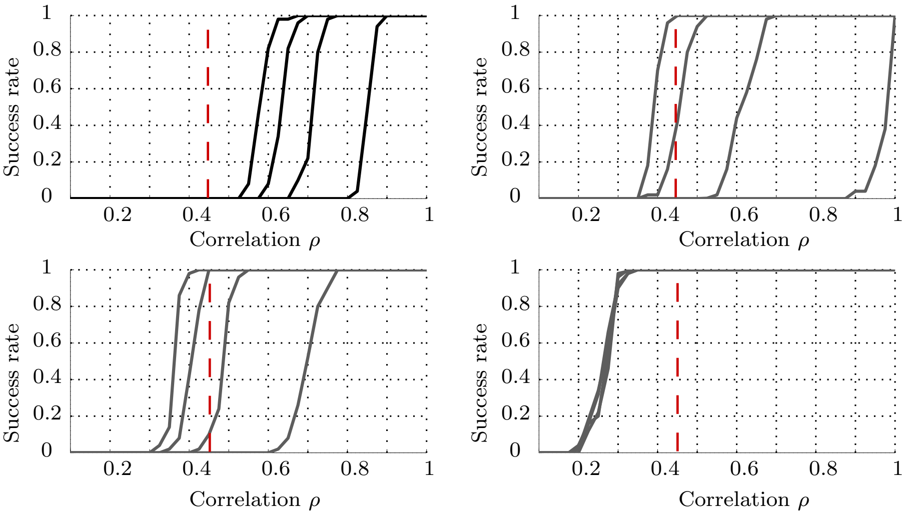


Figure 3.13: Success rate of different methods using seeds. We plot $\mathbf{P_c}$ (top left), FAQ:$\mathbf{J}$ (top right), FAQ:$\mathbf{D}^*$ (bottom left), and FAQ:$\mathbf{P}^*$ (bottom right). For each method, the number of seeds increases from right to left: 0 (black), 5 (green), 10 (blue) and 15 (red) seeds. Note that more seeds increases the success rate across the board.

the true permutation. On the other hand, when FAQ does recover the correct permutation, the algorithm runs significantly faster than when it fails to recover the truth. This suggests that, across all seed levels, the running time might, by itself, be a good indicator of whether the algorithm succeeded in recovering the underlying correspondence or not. Also note that as seeds increase, the overall speed of convergence of the algorithm decreases and, unsurprisingly, the correct permutation is obtained for lower correlation levels.

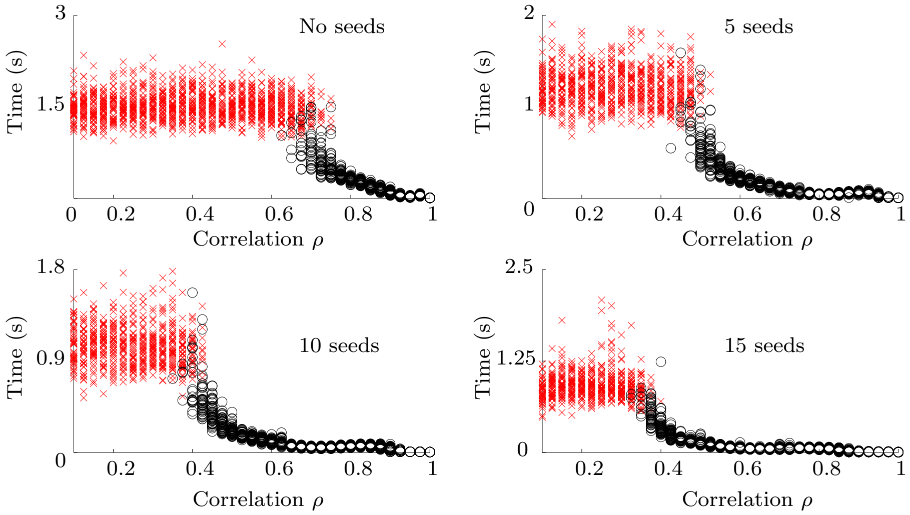


Figure 3.14: Running time for the nonconvex relaxation when starting from $\mathbf{D}^*$, for different number of seeds. A red "x" indicates the algorithm failed to recover $\mathbf{P}^*$, and a black "o" indicates it succeeded. In each, the algorithm was run to termination at discovery of a local min.

3.3.7 Features

Features are additional information that can be utilized to improve performance in graph matching methods, and often these features are manifested as additional vertex characteristics besides the connections with other vertices. For instance, in social networks we may have have a complete profile of a person in addition to his/her social connections.

We demonstrate the utility of using features with the nonconvex relaxation, the standard convex relaxation and the GLAG method, duely modified to include the features into the optimization. Namely, the new objective function to minimize is $\lambda F(\mathbf{P}) + (1-\lambda)\text{trace}(\mathbf{C}^T\mathbf{P})$, where $F(\mathbf{P})$ is the original cost function ($-\langle \mathbf{AP}, \mathbf{PB}\rangle$ in the nonconvex setting, $\|\mathbf{AP} - \mathbf{PB}\|_F^2$ for the convex relaxation and $\sum_{i,j} \|([\mathbf{AP}]_{i,j}, [\mathbf{PB}]_{i,j})\|_2$ for the GLAG method), the matrix $\mathbf{C}$ codes the features fitness cost, and the parameter λ balances the trade-off between pure graph matching and fit in the features domain. For each of the matching methodologies, the optimization is very similar to the original featureless version.

For the experiments, we generate ρ-correlated Bernoulli graphs as before, and in addition we generate a Gaussian random vector (zero mean, unit variance) of 5 features for each node of one graph, forming a $5 \times n$ matrix of features; we permute that matrix according to $\mathbf{P}^*$ to align new features vectors with the nodes of the second graph. Lastly, additive zero-mean Gaussian noise with a range of variance values is added to each feature matrix independently. If for each vertex $v \in [n]$ the resulting noisy feature for G_i, $i = 1, 2$, is $x_v^{(i)}$, then the entries of $\mathbf{C}$ are defined to be $\mathbf{C}_{v,w} = \|x_v^{(1)} - x_w^{(2)}\|_2$, for $v, w \in [n]$. Lastly, we set $\lambda = 0.5$.

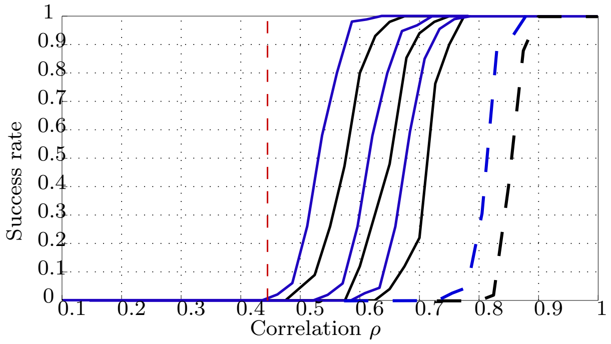


Figure 3.15: Success rate of different methods using features: $\mathbf{P_c}$ (in black), GLAG (in blue), FAQ:$\mathbf{D^*}$ (in red), and FAQ:$\mathbf{P^*}$ (in green). For each method, the noise level (variance of the Gaussian random noise) increases from left to right: 0.3, 0.5, and 0.7. In dashed lines, we show the success of the same methods without features.

Figure 3.15 shows the behavior of the methods when using features for different levels of noise in the feature matrix. Even for highly noisy features (recalling that both feature matrices are contaminated with noise), this external information still helps in the graph matching problem. For all noise levels, all three methods improve their performance with the addition of features, and of course, the improvement is greater when the noise level decreases. Note that, as before, FAQ outperforms both $\mathbf{P_c}$ and GLAG across all noise levels. It is also worth noting that for low noise, FAQ:$\mathbf{D^*}$ performs comparably to FAQ:$\mathbf{P^*}$, which we did not observe in the seeded (or unseeded) setting.

Even for modestly errorful features, including these features improves downstream matching performance versus the setting without features. This points to the utility of high fidelity features in the matching task. Indeed, given that the state-of-the-art graph matching algorithms may not achieve the optimal matching for even modestly correlated graphs, the use of external information like seeds and features can be critical.

This page was intentionally left blank.

Chapter 4

Graph Matching Theory

4.1 Introduction

The graph matching problem (aligning a pair of graphs to minimize their edge disagreements) has several interesting aspects: algorithms, applications, and theory. In the previous chapter we presented methods and applications of the graph matching problem, and we left some theoretical questions unanswered. In this chapter we tackle some of these open questions, both from probabilistic and deterministic perspectives.

For several applications, it is usual to model the graphs with certain probabilistic distributions, in order to better analyze or draw conclusions about the system. In Section 4.2 we address the graph matching problem from this probabilistic setting. The graph distribution used along this section is the most general edge independent model. Under this model, given two correlated random graphs, we prove some probability success results for two graph matching relaxations. Namely, the classical convex relaxation, and the non-convex relaxation described in [95].

In Section 4.3 we address some open problems in graph matching from a deterministic approach, and we also present some related results of graph theory in general. More specifically, as stated in Chapter 1, the graph matching problem is closely related to the graph isomorphism problem, and hence to the automorphism group of graphs.

The results proven in Section 4.3 are related to the spectral decomposition of the adjacency matrix. Indeed, we prove that certain conditions on the spectrum of the adjacency matrix are sufficient to guarantee the equivalence of the graph matching problem and its convex relaxation. Moreover, we provide an interpretation of these spectral conditions, we also prove other related results, and finally we state a conjecture, leaving open questions about the connection between the symmetry of a graph and the spectrum of its adjacency matrix.

4.2 Probabilistic Results for Graph Matching

Section summary

Graph matching has received wide-spread attention from both theoretical and applied communities over the past several decades, including combinatorics, computer vision, and connectomics. Its attention can be partially attributed to its computational difficulty. Although many heuristics have previously been proposed in the literature to approximately solve graph matching, very few have any theoretical support for their performance. A common technique is to relax the discrete problem to a continuous problem, therefore enabling practitioners to bring gradient-descent-type algorithms to bear. We prove that an indefinite relaxation (when solved exactly) almost always discovers the optimal permutation, while a common convex relaxation almost always fails to discover the optimal permutation. These theoretical results suggest that initializing the indefinite algorithm with the convex optimum might yield improved practical performance.

4.2.1 Introduction

Several problems related to the isomorphism and matching of graphs have been an important and enjoyable challenge for the scientific community for a long time, with applications in pattern recognition (see, for example, [14,19]), computer vision (see, for example, [23, 99, 104]), and machine learning (see, for example, [27, 57]), to name a few. Given two graphs, the graph isomorphism problem consists of determining whether these graphs are isomorphic or not, that is, if there exists a bijection between the vertex sets of the graphs which exactly preserves the vertex adjacency. The graph isomorphism problem is very challenging from a computational complexity point of view. Indeed, its complexity is still unresolved: it is not currently classified as NP-complete or P [50]. The graph isomorphism problem is a special case of the (harder) graph matching problem. The graph matching problem consists of finding the exact isomorphism between two graphs if it exists, or, in general, finding the bijection between the vertex sets that minimizes the number of adjacency disagreements. Graph matching is a very challenging and well-studied problem in the literature with applications in such diverse fields as pattern recognition, computer vision, neuroscience, etc. (see [25]). Although polynomial-time algorithms for solving the graph matching problem are known for certain classes of graphs (e.g., trees [88, 90]; planar graphs [55]; and graphs with some spectral properties [1, 43]), there are no known polynomial-time algorithms for solving the general case. Indeed, in its most general form, the graph matching problem is equivalent to the NP-hard quadratic assignment problem.

Formally, for any two graphs on n vertices with respective $n \times n$ adjacency matrices $\mathbf{A}$ and $\mathbf{B}$, the graph matching problem is to minimize $\|\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{P}^T\|_F$ over all $\mathbf{P} \in \mathcal{P}$, where $\mathcal{P}$ denotes the set of $n \times n$ permutation matrices, and $\|\cdot\|_F$ is the Froebenius matrix norm (other graph matching objectives have been proposed in the literature as well, this being a common one). Note that for any permutation matrix $\mathbf{P}$, $\frac{1}{2}\|\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{P}^T\|_F^2 = \frac{1}{2}\|\mathbf{A}\mathbf{P} - \mathbf{P}\mathbf{B}\|_F^2$ counts the number of adjacency disagreements induced by the vertex bijection corresponding to $\mathbf{P}$.

An equivalent formulation of the graph matching problem is to minimize $-\langle \mathbf{AP}, \mathbf{PB} \rangle$ over all $\mathbf{P} \in \mathcal{P}$, where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product, i.e., for all $\mathbf{C}, \mathbf{D} \in \mathbb{R}^{n \times n}$, $\langle \mathbf{C}, \mathbf{D} \rangle := \text{trace}(\mathbf{C}^\mathbf{T}\mathbf{D})$. This can be seen by expanding, for any $\mathbf{P} \in \mathcal{P}$,

$$\begin{aligned} \|\mathbf{A} - \mathbf{PBP}^\mathbf{T}\|_F^2 &= \|\mathbf{AP} - \mathbf{PB}\|_F^2 \\ &= \|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 - 2\langle \mathbf{AP}, \mathbf{PB} \rangle, \end{aligned}$$

and noting that $\|\mathbf{A}\|_F^2$ and $\|\mathbf{B}\|_F^2$ are constants for the optimization problem over $\mathbf{P} \in \mathcal{P}$.

Let $\mathcal{D}$ denote the set of $n \times n$ doubly stochastic matrices, i.e., nonnegative matrices with row and column sums each equal to 1. We define the *convex relaxed graph matching problem* to be minimizing $\|\mathbf{AD} - \mathbf{DB}\|_F^2$ over all $\mathbf{D} \in \mathcal{D}$, and we define the *indefinite relaxed graph matching problem* to be minimizing $-\langle \mathbf{AD}, \mathbf{DB} \rangle$ over all $\mathbf{D} \in \mathcal{D}$. Unlike the graph matching problem, which is an integer programming problem, these relaxed graph matching problems are each continuous optimization problems with a quadratic objective function subject to affine constraints. Since the quadratic objective $\|\mathbf{AD} - \mathbf{DB}\|_F^2$ is also convex in the variables $\mathbf{D}$ (it is a composition of a convex function and a linear function), there is a polynomial-time algorithm for exactly solving the convex relaxed graph matching problem (see [52]). However, $-\langle \mathbf{AD}, \mathbf{DB} \rangle$ is not convex (in fact, the Hessian has trace zero and is therefore *indefinite*), and nonconvex quadratic programming is (in general) NP-hard. Nonetheless the indefinite relaxation can be efficiently approximately solved with Frank-Wolfe (F-W) methodology [46, 95].

It is natural to ask how the (possibly different) solutions to these relaxed formulations relate to the solution of the original graph matching problem. Our main theoretical result, Theorem 4.1, proves, under mild conditions, that convex relaxed graph matching (which is tractable) almost always yields the wrong matching, and indefinite relaxed graph matching (which is intractable) almost always yields the correct matching.

In light of graph matching complexity results (see for example [1, 6, 79]), it is unsurprising that the convex relaxation can fail to recover the true permutation. In our main theorem, we take this a step further and provide an answer from a probabilistic point of view, showing almost sure failure of the convex relaxation for a very rich and general family of graphs. This paints a sharp contrast to the (surprising) almost sure correctness of the solution of the indefinite relaxation. We further illustrate that our theory gives rise to a new state-of-the-art matching strategy.

Correlated random Bernoulli graphs

Our theoretical results will be set in the context of correlated random (simple) Bernoulli graphs,[1] which can be used to model many real-data scenarios. Random Bernoulli graphs are the most general edge independent random graphs, and contain many important random graph families including Erdős-Rényi and the widely

[1] Also known as *inhomogeneous random graphs* in [17].

used stochastic block model of [54] (in the stochastic block model, Λ is a block constant matrix, with the number of diagonal blocks representing the number of communities in the network). Stochastic block models, in particular, have been extensively used to model networks with inherent community structure (see, for example, [2, 75, 78, 85]). As this model is a submodel of the random Bernoulli graph model used here, our main theorem (Theorem 4.1) extends to stochastic block models immediately, making it of highly practical relevance.

These graphs are defined as follows. Given $n \in \mathbb{Z}^+$, a real number $\rho \in [0,1]$, and a symmetric, hollow matrix $\mathbf{\Lambda} \in [0,1]^{n\times n}$, define $\mathcal{E} := \{\{i,j\} : i \in [n], j \in [n], i \neq j\}$, where $[n] := \{1,2,\ldots,n\}$. Two random graphs with respective $n \times n$ adjacency matrices $\mathbf{A}$ and $\mathbf{B}$ are *ρ-correlated Bernoulli($\mathbf{\Lambda}$)* distributed if, for all $\{i,j\} \in \mathcal{E}$, the random variables (matrix entries) $\mathbf{A}_{i,j}, \mathbf{B}_{i,j}$ are Bernoulli($\mathbf{\Lambda}_{i,j}$) distributed, and all of these random variables are collectively independent except that, for each $\{i,j\} \in \mathcal{E}$, the Pearson product-moment correlation coefficient for $\mathbf{A}_{i,j}, \mathbf{B}_{i,j}$ is ρ. It is straightforward to show that the parameters n, ρ, and $\mathbf{\Lambda}$ completely specify the random graph pair distribution, and the distribution may be achieved by first, for all $\{i,j\} \in \mathcal{E}$, having $\mathbf{B}_{ij} \sim$ Bernoulli($\mathbf{\Lambda}_{i,j}$) independently drawn and then, conditioning on $\mathbf{B}$, have $\mathbf{A}_{i,j} \sim \text{Bernoulli}\left((1-\rho)\mathbf{\Lambda}_{i,j} + \rho\mathbf{B}_{i,j}\right)$ independently drawn. While $\rho = 1$ would imply the graphs are isomorphic, this model allows for a natural vertex alignment (namely the identity function) for $\rho < 1$, i.e. when the graphs are not necessarily isomorphic.

4.2.2 The main result

We will consider a sequence of correlated random Bernoulli graphs for $n = 1,2,3,\ldots$, where $\mathbf{\Lambda}$ is a function of n. When we say that a sequence of events, $\{E_m\}_{m=1}^{\infty}$, holds *almost always* we mean that almost surely it happens that the events in the sequence occur for all but finitely many m.

Theorem 4.1. *Suppose $\mathbf{A}$ and $\mathbf{B}$ are adjacency matrices for ρ-correlated Bernoulli($\mathbf{\Lambda}$) graphs, and there is an $\alpha \in (0,1/2)$ such that $\mathbf{\Lambda}_{i,j} \in [\alpha, 1-\alpha]$ for all $i \neq j$. Let $\mathbf{P}^* \in \mathcal{P}$, and denote $\mathbf{A'} := \mathbf{P}^*\mathbf{A}\mathbf{P}^{*\mathbf{T}}$.*
a) If $(1-\alpha)(1-\rho) < 1/2$, then it almost always holds that

$$\arg\min_{\mathbf{D}\in\mathcal{D}} -\langle \mathbf{A'D}, \mathbf{DB}\rangle = \arg\min_{\mathbf{P}\in\mathcal{P}} \|\mathbf{A'} - \mathbf{PBP^T}\|_F = \{\mathbf{P}^*\}.$$

b) If the between graph correlation $\rho < 1$, then it almost always holds that $\mathbf{P}^* \notin \arg\min_{\mathbf{D}\in\mathcal{D}} \|\mathbf{A'D} - \mathbf{DB}\|_F$.

This theorem states that: (part *a*) the unique solution of the indefinite relaxation almost always is the correct permutation matrix, while (part *b*) the correct permutation is almost always not a solution of the commonly used convex relation. Moreover, as shown in Section 3.3, the convex relaxation can lead to a doubly stochastic matrix that is not even in the Voronoi cell of the true permutation. In this case, the convex optimum is closest to an incorrect permutation, hence the correct permutation will not be recovered by projecting the doubly stochastic solution back onto $\mathcal{P}$.

In the above, ρ and α are fixed. However, the proofs follow *mutatis mutandis* if ρ and α are allowed to vary with n. If there exist constants $c_1, c_2 > 0$ such that $\alpha \geq c_1\sqrt{(\log n)/n}$ and $1/2 - c_2\sqrt{(\log n)/n} \geq (1-\rho)(1-\alpha)$, then Theorem 4.1, part *a* will hold. Note that $\alpha \geq c_1\sqrt{(\log n)/n}$ also guarantees the corresponding graphs are almost always connected. For the analogous result for part *b*, let us first define $\sigma(i) = \frac{1}{n-1}\sum_{k\neq i}\mathbf{\Lambda}_{ki}(1-\mathbf{\Lambda}_{ki})$. If there exists an $i \in [n]$ such that $1 - \frac{3}{2\sigma(i)}\sqrt{(8\log n)/n} > \rho$, then the results of Theorem 4.1, part *b* hold as proven below.

Isomorphic versus ρ-correlated graphs

There are numerous algorithms available in the literature for (approximately) solving the graph isomorphism problem (see, for example, [26,37]), as well as for (approximately) solving the subgraph isomorphism problem (see, for example, [91]). All of the graph matching algorithms we explore herein can be used for the graph isomorphism problem as well.

We emphasize that the ρ-correlated random graph model extends our random graphs beyond isomorphic graph pairs; indeed ρ-correlated graphs G_1 and G_2 will almost surely have on the order of $[\alpha, 1-\alpha]\rho n^2$ edge-wise disagreements. As such, these graphs are a.s. *not* isomorphic. In this setting, the goal of graph matching is to align the vertices across graphs whilst simultaneously preserving the adjacency structure as best possible across graphs. However, this model does preserve a very important feature of isomorphic graphs: namely the presence of a latent alignment function (the identity function in the ρ-correlated model).

We note here that in the ρ-correlated Bernoulli($\mathbf{\Lambda}$) model, both G_1 and G_2 are marginally Bernoulli($\mathbf{\Lambda}$) random graphs, which is amenable to theoretical analysis. We note here that real data experiments across a large variety of data sets (see Section 3.3.3) and simulated experiments across a variety of robust random graph settings (see Section 3.3.4) also both support the result of Theorem 4.1. Indeed, we suspect that an analogue of Theorem 4.1 holds over a much broader class of random graphs, and we are presently investigating this extension.

4.2.3 Proof of Theorem 4.1, part a

Without loss of generality, let $\mathbf{P}^* = \mathbf{I}$. We will first sketch the main argument of the proof, and then we will spend the remainder of the section filling in all necessary details of the proof. The proof will proceed as follows. Almost always, $-\langle \mathbf{A}, \mathbf{B}\rangle < -\langle \mathbf{AQ}, \mathbf{PB}\rangle$ for any $\mathbf{P}$, $\mathbf{Q} \in \mathcal{P}$ such that either $\mathbf{P} \neq \mathbf{I}$ or $\mathbf{Q} \neq \mathbf{I}$. To accomplish this, we count the entrywise disagreements between $\mathbf{AQ}$ and $\mathbf{PB}$ in two steps (of course, this is the same as the number of entrywise disagreements between $\mathbf{A}$ and $\mathbf{PBQ^T}$). We first count the entrywise disagreements between $\mathbf{B}$ and $\mathbf{PBQ^T}$ (Lemma 4), and then count the additional disagreements induced by realizing $\mathbf{A}$ conditioning on $\mathbf{B}$. Almost always, this two step realization will result in more errors than simply realizing $\mathbf{A}$ directly from $\mathbf{B}$ without permuting the vertex labels (Lemma 5). This establishes $-\langle \mathbf{A}, \mathbf{B}\rangle < -\langle \mathbf{AQ}, \mathbf{PB}\rangle$, and Theorem 4.1, part *a* is a consequence of the Birkhoff-von Neumann theorem.

We begin with two lemmas used to prove Theorem 4.1. First, Lemma 2 is adapted from [5], presented here as a variation of the form found in [59, Prop. 3.2]. This lemma lets us tightly estimate the number of disagreements between $\mathbf{B}$ and $\mathbf{PBQ^T}$, which we do in Lemma 4.

Lemma 2. *For any integer $N > 0$ and constant $\alpha \in (0, \frac{1}{2})$, suppose that the random variable X is a function of at most N independent Bernoulli random variables, each with Bernoulli parameter in the interval $[\alpha, 1-\alpha]$. Suppose that changing the value of any one of the Bernoulli random variables (and keeping all of the others fixed) changes the value of X by at most γ. Then for any t such that $0 \leq t < \sqrt{\alpha(1-\alpha)}\gamma N$, it holds that $\mathbb{P}\left[|X - \mathbb{E}X| > t\right] \leq 2 \cdot exp\{-t^2/(\gamma^2 N)\}$.*

The next result, Lemma 3, is a special case of the classical Hoeffding inequality (see, for example, [24]), which we use to tightly bound the number of additional entrywise disagreements between $\mathbf{AQ}$ and $\mathbf{PB}$ when we realize $\mathbf{A}$ conditioning on $\mathbf{B}$.

Lemma 3. *Let N_1 and N_2 be positive integers, and q_1 and q_2 be real numbers in $[0, 1]$. If $X_1 \sim \text{Binomial}(N_1, q_1)$ and $X_2 \sim \text{Binomial}(N_2, q_2)$ are independent, then for any $t \geq 0$ it holds that*

$$\mathbb{P}\Big[\Big|X_1 + X_2 - \mathbb{E}\Big(X_1 + X_2\Big)\Big| \geq t\Big] \leq 2 \cdot \exp\left\{\frac{-2t^2}{N_1 + N_2}\right\}.$$

Setting notation for the next lemmas, let n be given. Let $\mathcal{P}$ denote the set of $n \times n$ permutation matrices. Just for now, fix any $\mathbf{P}, \mathbf{Q} \in \mathcal{P}$ such that they are not both the identity matrix, and let τ, ω be their respective associated permutations on $[n]$; i.e. for all $i, j \in [n]$ it holds that $\tau(i) = j$ precisely when $\mathbf{P}_{i,j} = 1$ and, for all $i, j \in [n]$, it holds that $\omega(i) = j$ precisely when $\mathbf{Q}_{i,j} = 1$. It will be useful to define the following sets:

$$\begin{aligned}
\Delta &:= \{(i,j) \in [n] \times [n] : \tau(i) \neq i \text{ or } \omega(j) \neq j\}, \\
\Delta_t &:= \{(i,j) \in \Delta : \tau(i) = j \text{ and } \omega(j) = i\}, \\
\Delta_d &:= \{(i,j) \in \Delta : i = j \text{ or } \tau(i) = \omega(j)\}, \\
\Delta_\tau &:= \{(i,j) \in [n] \times [n] : \tau(i) \neq i\}, \\
\Delta_\omega &:= \{(i,j) \in [n] \times [n] : \omega(j) \neq j\}.
\end{aligned}$$

If we define m to be the maximum of $|\{i \in [n] : \tau(i) \neq i\}|$ and $|\{j \in [n] : \omega(j) \neq j\}|$, then it follows that $mn \leq |\Delta| \leq 2mn$. This is clear from noting that $\Delta_\omega, \Delta_\tau \subseteq \Delta \subseteq \Delta_\tau \cup \Delta_\omega$. Also, $|\Delta_t| \leq m$, since for $(i,j) \in \Delta_t$ it is necessary that $\tau(i) \neq i$ and $\omega(j) \neq j$. Lastly, $|\Delta_d| \leq 4m$, since

$$\Delta_d \subseteq \{(i,i) \in \Delta\} \cup \{(i,j) \in \Delta : i \neq j, \, \tau(i) = \omega(j)\},$$

and $|\{(i,i) \in \Delta\}| \leq 2m$, and $|\{(i,j) \in \Delta : i \neq j, \, \tau(i) = \omega(j)\}| \leq 2m$.

We make the following assumption in all that follows:

Assumption 1: *Suppose that $\mathbf{\Lambda} \in [0,1]^{n \times n}$ is a symmetric, hollow matrix, there* *is a real number* $\rho \in [0,1]$*, and there is a constant* $\alpha \in (0,1/2)$ *such that* $\mathbf{\Lambda}_{i,j} \in [\alpha, 1-\alpha]$ *for all* $i \neq j$*, and* $(1-\alpha)(1-\rho) < 1/2$*. Further, let* $\mathbf{A}$*,* $\mathbf{B}$ *be the adjacency matrices of two random* ρ*-correlated Bernoulli(*$\mathbf{\Lambda}$*) graphs.*

Define the (random) set

$$\Theta' := \{(i,j) \in \Delta : i \neq j, \text{ and } B_{i,j} \neq B_{\tau(i),\omega(j)}\}.$$

Note that $|\Theta'|$ counts the entrywise disagreements induced *within* the off-diagonal part of B by τ and ω.

Lemma 4. *Under Assumption 1, if* n *is sufficiently large then*

$$\mathbb{P}\left(|\Theta'| \notin [\alpha mn/3, \; 2mn]\right) \leq 2e^{-\alpha^2 mn/128}.$$

Proof of Lemma 4: For any $(i,j) \in \Delta$, note that $(\mathbf{B}_{i,j} - \mathbf{B}_{\tau(i),\omega(j)})^2$ has a Bernoulli distribution; if $(i,j) \in \Delta_t \cup \Delta_d$, then the Bernoulli parameter is either 0 or is in the interval $[\alpha, 1-\alpha]$, and if $(i,j) \in \Delta \backslash (\Delta_t \cup \Delta_d)$, then the Bernoulli parameter is $\mathbf{\Lambda}_{i,j}(1 - \mathbf{\Lambda}_{\tau(i),\omega(j)}) + (1 - \mathbf{\Lambda}_{i,j})\mathbf{\Lambda}_{\tau(i),\omega(j)}$, and this Bernoulli parameter is in the interval $[\alpha, 1-\alpha]$ since it is a convex combination of values in this interval. Now, $|\Theta'| = \sum_{(i,j)\in\Delta, i\neq j}(\mathbf{B}_{i,j} - \mathbf{B}_{\tau(i),\omega(j)})^2$, so we obtain that $\alpha\left(|\Delta| - |\Delta_t| - |\Delta_d|\right) \leq \mathbb{E}(|\Theta'|) \leq (1-\alpha)|\Delta|$, and thus

$$\alpha m(n-5) \;\leq\; \mathbb{E}(|\Theta'|) \;\leq\; 2(1-\alpha)mn. \tag{4.1}$$

Next we apply Lemma 2, since $|\Theta'|$ is a function of the at-most $N := 2mn$ Bernoulli random variables $\{\mathbf{B}_{i,j}\}_{(i,j)\in\Delta : i\neq j}$, which as a set (noting that $\mathbf{B}_{i,j} = \mathbf{B}_{j,i}$ is counted at most once for each $\{i,j\}$) are independent, each with Bernoulli parameter in $[\alpha, 1-\alpha]$. Furthermore, changing the value of any one of these random variable would change $|\Theta'|$ by at most $\gamma := 4$, thus Lemma 2 can be applied and, for the choice of $t := \frac{\alpha}{2}mn$, we obtain that

$$\mathbb{P}\left[\big||\Theta'| - \mathbb{E}(|\Theta'|)\big| > \alpha mn/2\right] \leq 2e^{-\alpha^2 mn/128}. \tag{4.2}$$

Lemma 4 follows from (4.1) and (4.2), since

$$\begin{aligned}
&\mathbb{P}\left[\big||\Theta'| - \mathbb{E}(|\Theta'|)\big| > \alpha mn/2\right] \\
&= \mathbb{P}\left[|\Theta'| \notin \left[\mathbb{E}(|\Theta'|) - \alpha mn/2, \mathbb{E}(|\Theta'|) + \alpha mn/2\right]\right] \\
&\geq \mathbb{P}\left[|\Theta'| \notin \left[\alpha m(n-5) - \alpha mn/2, 2(1-\alpha)mn + \alpha mn/2\right]\right] \\
&\geq \mathbb{P}\left[|\Theta'| \notin \left[\alpha m(n-5) - \alpha mn/2, 2mn\right]\right],
\end{aligned}$$

and $5\alpha mn/6 \leq \alpha m(n-5)$ when n is sufficiently large (e.g. $n \geq 30$). ■

With the above bound on the number of (non-diagonal) entrywise disagreements between $\mathbf{B}$ and $\mathbf{PBQ^T}$, we next count the number of additional disagreements introduced by realizing $\mathbf{A}$ conditioning on $\mathbf{B}$. In Lemma 5, we prove that this two step realization will almost always result in more entrywise errors than simply realizing $\mathbf{A}$ from $\mathbf{B}$ without permuting the vertex labels.

Lemma 5. *Under Assumption 1, it almost always holds that, for all* $\mathbf{P}, \mathbf{Q} \in \mathcal{P}$ *such that either* $\mathbf{P} \neq \mathbf{I}$ *or* $\mathbf{Q} \neq \mathbf{I}$*,* $\|\mathbf{A} - \mathbf{PBQ^T}\|_F > \|\mathbf{A} - \mathbf{B}\|_F$*.*

Proof of Lemma 5: Just for now, let us fix any $\mathbf{P}, \mathbf{Q} \in \mathcal{P}$ such that either $\mathbf{P} \neq \mathbf{I}$ or $\mathbf{Q} \neq I$, and say τ and ω are their respective associated permutations on $[n]$. Let Δ and Θ' be defined as before. For every $(i,j) \in \Delta$, a combinatorial argument, combined with A and B being binary valued, yields (where for an event C, $\mathbb{1}_C$ is the indicator random variable for the event C)

$$\begin{aligned}\mathbb{1}_{\mathbf{A}_{i,j}\neq\mathbf{B}_{i,j}} + \mathbb{1}_{\mathbf{B}_{i,j}\neq\mathbf{B}_{\tau(i),\omega(j)}} = \\ \mathbb{1}_{\mathbf{A}_{i,j}\neq\mathbf{B}_{\tau(i),\omega(j)}} + 2 \cdot \mathbb{1}_{\mathbf{A}_{i,j}\neq\mathbf{B}_{i,j}\ \&\ \mathbf{B}_{i,j}\neq\mathbf{B}_{\tau(i),\omega(j)}}.\end{aligned} \tag{4.3}$$

Note that

$$\|\mathbf{A}-\mathbf{PBQ^T}\|_F^2=\sum_{i,j}(\mathbf{A}_{i,j}-\mathbf{B}_{\tau(i),\omega(j)})^2=\sum_{i,j}\mathbb{1}_{\mathbf{A}_{i,j}\neq\mathbf{B}_{\tau(i),\omega(j)}}$$
$$\|\mathbf{A}-\mathbf{B}\|_F^2=\sum_{i,j}(\mathbf{A}_{i,j}-\mathbf{B}_{i,j})^2=\sum_{i,j}\mathbb{1}_{\mathbf{A}_{i,j}\neq\mathbf{B}_{i,j}}.$$

Summing Eq. (4.3) over the relevant indices then yields that

$$\|\mathbf{A}-\mathbf{PBQ^T}\|_F^2 - \|\mathbf{A}-\mathbf{B}\|_F^2 = |\Theta| - 2|\Gamma|, \tag{4.4}$$

where the sets Θ and Γ are defined as

$$\begin{aligned}\Theta &:= \{(i,j) \in [n]\times[n] : \mathbf{B}_{i,j} \neq \mathbf{B}_{\tau(i),\omega(j)}\} \subseteq \Delta,\\ \Gamma &:= \{(i,j) \in \Theta : \mathbf{A}_{i,j} \neq \mathbf{B}_{i,j}\}.\end{aligned}$$

Now, partition Θ into sets Θ_1, Θ_2, Θ_d, and partition Γ into sets Γ_1, Γ_2 where

$$\begin{aligned}\Theta_1 &:= \{(i,j) \in \Theta : i \neq j \text{ and } (j,i) \notin \Theta\},\\ \Theta_2 &:= \{(i,j) \in \Theta : i \neq j \text{ and } (j,i) \in \Theta\},\\ \Theta_d &:= \{(i,j) \in \Theta : i = j\},\\ \Gamma_1 &:= \{(i,j) \in \Theta_1 : \mathbf{A}_{i,j} \neq \mathbf{B}_{i,j}\},\\ \Gamma_2 &:= \{(i,j) \in \Theta_2 : \mathbf{A}_{i,j} \neq \mathbf{B}_{i,j}\}.\end{aligned}$$

Note that all (i,j) such that $i=j$ are not in Γ. Also note that $\Theta' \subseteq \Theta$ can be partitioned into the disjoint union $\Theta' = \Theta_1 \cup \Theta_2$.

Equation (4.4) implies

$$\begin{aligned}|\Gamma_1| + |\Gamma_2| < (|\Theta_1|+|\Theta_2|)/2 \Rightarrow |\Gamma| < |\Theta|/2 \Rightarrow \\ \|\mathbf{A}-\mathbf{B}\|_F^2 < \|\mathbf{A}-\mathbf{PBQ^T}\|_F^2.\end{aligned}$$

In particular,

$$\begin{aligned}\left\{\|\mathbf{A}-\mathbf{B}\|_F \geq \|\mathbf{A}-\mathbf{PBQ^T}\|_F\right\} \Rightarrow \\ \left\{|\Gamma_1| + |\Gamma_2| \geq (|\Theta_1| + |\Theta_2|)/2 = |\Theta'|/2\right\}.\end{aligned} \tag{4.5}$$

Now, conditioning on $\mathbf{B}$ (hence, conditioning on Θ'), we have, for all $i \neq j$, that (see Section 4.2.1), $\mathbf{A}_{i,j} \sim \text{Bernoulli}\left((1-\rho)\mathbf{\Lambda}_{i,j} + \rho\mathbf{B}_{i,j}\right)$. Thus $\mathbb{1}_{\mathbf{A}_{i,j}\neq\mathbf{B}_{i,j}}$

has a Bernoulli distribution with parameter bounded above by $(1-\alpha)(1-\rho)$. Thus, $|\Gamma_1|$ is stochastically dominated by a Binomial$(|\Theta_1|,(1-\alpha)(1-\rho))$ random variable, and the independent random variable $|\Gamma_2|$ is stochastically dominated by a Binomial$(|\Theta_2|,(1-\alpha)(1-\rho))$ random variable. An application of Lemma 3 with $N_1 := |\Theta_1|$, $N_2 := |\Theta_2|$, $q_1 = q_2 := (1-\alpha)(1-\rho)$, and $t := \left(\frac{1}{2}-(1-\alpha)(1-\rho)\right)|\Theta'|$, yields (recall that we are conditioning on $\mathbf{B}$ here)

$$\begin{aligned}
&\mathbb{P}\left[|\Gamma_1|+|\Gamma_2|\geq|\Theta'|/2\right]\\
&=\mathbb{P}\Big[|\Gamma_1|+|\Gamma_2|-(1-\alpha)(1-\rho)|\Theta'|\geq\Big(1/2-(1-\alpha)(1-\rho)\Big)|\Theta'|\Big]\\
&\leq 2\mathrm{exp}\left\{\frac{-2\left(1/2-(1-\alpha)(1-\rho)\right)^2|\Theta'|^2}{|\Theta_1|+|\Theta_2|}\right\}\\
&\leq 2\mathrm{exp}\left\{-2\Big(1/2-(1-\alpha)(1-\rho)\Big)^2|\Theta'|\right\}. \qquad (4.6)
\end{aligned}$$

No longer conditioning (broadly) on $\mathbf{B}$, Lemma 4, equations (4.5) and (4.6), and $(1-\alpha)(1-\rho)<\frac{1}{2}$, imply that

$$\begin{aligned}
&\mathbb{P}\Big[\|\mathbf{A}-\mathbf{PBQ^T}\|_F\leq\|\mathbf{A}-\mathbf{B}\|_F\Big]\\
&\leq\mathbb{P}\left(|\Theta'|\notin\left[\alpha mn/3,\ 2mn\right]\right)\\
&\qquad+\mathbb{P}\Big[|\Gamma_1|+|\Gamma_2|\geq\frac{1}{2}|\Theta'|\ \Big|\ |\Theta'|\in\big[\frac{\alpha}{3}mn,\ 2mn\big]\Big]\\
&\leq 4\exp\left\{-\min\left\{\frac{\alpha^2}{128},\frac{2\alpha}{3}\left(\frac{1}{2}-(1-\alpha)(1-\rho)\right)^2\right\}mn\right\}. \qquad (4.7)
\end{aligned}$$

Until this point, $\mathbf{P}$ and $\mathbf{Q}$—and their associated permutations τ and ω—have been fixed. Now, for each $m\in[n]$, define $\mathcal{H}_m$ to be the event that $\|\mathbf{A}-\mathbf{PBQ^T}\|_F\leq\|\mathbf{A}-\mathbf{B}\|_F$ for *any* $\mathbf{P},\mathbf{Q}\in\mathcal{P}$ with the property that their associated permutations τ,ω are such that the maximum of $|\{i\in[n]:\tau(i)\neq i\}|$ and $|\{j\in[n]:\omega(j)\neq j\}|$ is exactly m. There are at most $\binom{n}{m}m!\binom{n}{m}m!\leq n^{2m}$ such permutation pairs.

By (4.7), for every $m\in[n]$, setting

$$c_1=\min\{\alpha^2/128,2\alpha(1/2-(1-\alpha)(1-\rho))^2/3\},$$

we have $\mathbb{P}(\mathcal{H}_m)\leq n^{2m}\cdot 4\exp\{-c_1mn\}\leq\exp\{-c_2n\}$, for some positive constant c_2 (the last inequality holding when n is large enough). Thus, for sufficiently large n, $\mathbb{P}(\cup_{m=1}^n\mathcal{H}_m)\leq n\cdot\exp\{-c_2n\}$ decays exponentially in n, and is thus finitely summable over $n=1,2,3,\ldots$. Lemma 5 follows from the Borel-Cantelli Lemma. ■

Proof of Theorem 4.1, part a: By Lemma 5, it almost always follows that for every $\mathbf{P},\mathbf{Q}\in\mathcal{P}$ not both the identity, $\langle\mathbf{AQ},\mathbf{PB}\rangle<\langle\mathbf{A},\mathbf{B}\rangle$. By the Birkhoff-von Neuman Theorem, $\mathcal{D}$ is the convex hull of $\mathcal{P}$, i.e., for every $\mathbf{D}\in\mathcal{D}$, there exists constants $\{a_{D,P}\}_{P\in\mathcal{P}}$ such that $\mathbf{D}=\sum_{\mathbf{P}\in\mathcal{P}}a_{D,P}\mathbf{P}$ and $\sum_{\mathbf{P}\in\mathcal{P}}a_{D,P}=1$. Thus, if

$\mathbf{D}$ is not the identity matrix, then almost always

$$\begin{aligned}\langle \mathbf{AD}, \mathbf{DB}\rangle &= \sum_{\mathbf{P}\in\mathcal{P}}\sum_{\mathbf{Q}\in\mathcal{P}} a_{D,P}a_{D,Q}\langle \mathbf{AQ}, \mathbf{PB}\rangle \\ &< \sum_{\mathbf{P}\in\mathcal{P}}\sum_{\mathbf{Q}\in\mathcal{P}} a_{D,P}a_{D,Q}\langle \mathbf{A}, \mathbf{B}\rangle = \langle \mathbf{A}, \mathbf{B}\rangle,\end{aligned}$$

and almost always $\operatorname{argmin}_{\mathbf{D}\in\mathcal{D}} -\langle \mathbf{AD}, \mathbf{DB}\rangle = \{\mathbf{I}\}$. ■

4.2.4 Proof of Theorem 4.1, part b

The proof will proceed as follows: we will use Lemma 6 to prove that the identity is almost always not a KKT (Karush-Kuhn-Tucker) point of the relaxed graph matching problem. Since the relaxed graph matching problem is a constrained optimization problem with convex feasible region and affine constraints, this is sufficient for the proof of Theorem 4.1, part b.

First, we state Lemma 6, a variant of Hoeffding's inequality, which we use to prove Theorem 4.1, part b.

Lemma 6. *Let N be a positive integer. Suppose that the random variable X is the sum of N independent random variables, each with mean 0 and each taking values in the real interval $[-1,1]$. Then for any $t \geq 0$, it holds that*

$$\mathbb{P}[|X| \geq t] \leq 2 \cdot e^{\frac{-t^2}{2N}}.$$

Again, without loss of generality, we may assume $\mathbf{P}^* = \mathbf{I}$. We first note that the convex relaxed graph matching problem can be written as

$$\min \|\mathbf{AD} - \mathbf{DB}\|_F^2, \tag{4.8}$$

$$\text{s.t. } \mathbf{D1} = \mathbf{1}, \tag{4.9}$$

$$\mathbf{1}^T\mathbf{D} = \mathbf{1}^T, \tag{4.10}$$

$$\mathbf{D} \geq 0, \tag{4.11}$$

where (4.8) is a convex function (of $\mathbf{D}$) subject to affine constraints (4.9)-(4.11) (i.e., $\mathbf{D} \in \mathcal{D}$). It follows that if $\mathbf{I}$ is the global (or local) optimizer of the convex relaxed graph matching problem, then $\mathbf{I}$ must be a KKT (Karush-Kuhn-Tucker) point (see, for example, [12, Chapter 4]).

The gradient of $\|\mathbf{AD} - \mathbf{DB}\|_F^2$ (as a function of $\mathbf{D}$) is

$$\boldsymbol{\nabla}(\mathbf{D}) := 2(\mathbf{A^TAD} + \mathbf{DBB^T} - \mathbf{A^TDB} - \mathbf{ADB^T}).$$

Hence, a $\widehat{\mathbf{D}}$ satisfying (4.9)-(4.11) (i.e., $\widehat{\mathbf{D}}$ is primal feasible) is a KKT point if it satisfies

$$\boldsymbol{\nabla}(\widehat{\mathbf{D}}) + \boldsymbol{\mu} + \boldsymbol{\mu}' - \boldsymbol{\nu} = 0, \tag{4.12}$$

where $\boldsymbol{\mu}$, $\boldsymbol{\mu}'$, and $\boldsymbol{\nu}$ are dual variables as follows:

$$\boldsymbol{\mu} := \begin{bmatrix} \mu_1 & \mu_1 & \cdots & \mu_1 \\ \mu_2 & \mu_2 & \cdots & \mu_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mu_n & \mu_n & \cdots & \mu_n \end{bmatrix} \in \mathbb{R}^{n\times n},$$

noting that the dual variables $\mu_1, \mu_2, \ldots, \mu_n$ are not restricted. They correspond to the equality primal constraints (4.9) that the row-sums of a primal feasible $\mathbf{D}$ are all one;

$$\boldsymbol{\mu}' := \begin{bmatrix} \mu_1' & \mu_2' & \cdots & \mu_n' \\ \mu_1' & \mu_2' & \cdots & \mu_n' \\ \vdots & \vdots & \ddots & \vdots \\ \mu_1' & \mu_2' & \cdots & \mu_n' \end{bmatrix} \in \mathbb{R}^{n\times n},$$

noting that the dual variables $\mu_1', \mu_2', \ldots, \mu_n'$ are not restricted. They correspond to the equality primal constraints (4.10) that the column-sums of a primal feasible $\mathbf{D}$ are all one;

$$\boldsymbol{\nu} := \begin{bmatrix} 0 & \nu_{1,2} & \cdots & \nu_{1,n} \\ \nu_{2,1} & 0 & \cdots & \nu_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \nu_{n,1} & \nu_{n,2} & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n\times n},$$

noting that the dual variables $\nu_{i,j}$ are restricted to be nonnegative. They correspond to the inequality primal constraints (4.11) that the entries of a primal feasible $\mathbf{D}$ be nonnegative. Complementary slackness further constrains the $\nu_{i,j}$, requiring that $\widehat{\mathbf{D}}_{i,j}\nu_{i,j} = 0$ for all i, j.

At the identity matrix $\mathbf{I}$, the gradient $\boldsymbol{\nabla}(\mathbf{I})$, denoted $\boldsymbol{\nabla}$, simplifies to $\boldsymbol{\nabla} = [\nabla_{i,j}] = 2\mathbf{A}^2 + 2\mathbf{B}^2 - 4\mathbf{AB} \in \mathbb{R}^{n\times n}$; and $\mathbf{I}$ being a KKT point is equivalent to:

$$\boldsymbol{\nabla} + \boldsymbol{\mu} + \boldsymbol{\mu}' - \boldsymbol{\nu} = 0, \tag{4.13}$$

where $\boldsymbol{\mu}$, $\boldsymbol{\mu}'$, and $\boldsymbol{\nu}$ are as specified above. At the identity matrix, complimentary slackness translates to having $\nu_{1,1} = \nu_{2,2} = \cdots = \nu_{n,n} = 0$.

Now, for Equation (4.13) to hold, it is necessary that there exist $\mu_1, \mu_2, \mu_1', \mu_2'$ such that

$$\nabla_{1,1} + \mu_1 + \mu_1' \;=\; 0, \tag{4.14}$$

$$\nabla_{2,2} + \mu_2 + \mu_2' \;=\; 0, \tag{4.15}$$

$$\nabla_{1,2} + \mu_1 + \mu_2' \;\geq\; 0, \tag{4.16}$$

$$\nabla_{2,1} + \mu_2 + \mu_1' \;\geq\; 0. \tag{4.17}$$

Adding equations (4.16), (4.17) and subtracting equations (4.14), (4.15), we obtain

$$\nabla_{1,2} + \nabla_{2,1} \geq \nabla_{1,1} + \nabla_{2,2}. \tag{4.18}$$

Note that $\frac{1}{2}\boldsymbol{\nabla} + \frac{1}{2}\boldsymbol{\nabla}^T = 2(\mathbf{A}-\mathbf{B})^T(\mathbf{A}-\mathbf{B})$, hence Equation (4.18) is equivalent to (where $\mathbf{X} := (\mathbf{A}-\mathbf{B})^T(\mathbf{A}-\mathbf{B})$)

$$2[\mathbf{X}]_{1,2} \geq [\mathbf{X}]_{1,1} + [\mathbf{X}]_{2,2}. \tag{4.19}$$

Next, referring back to the joint distribution of $\mathbf{A}$ and $\mathbf{B}$ (see Section 4.2.1), we have, for all $i \neq j$,

$$\begin{aligned}\mathbb{P}\big[\mathbf{A}_{i,j} = 0,\, \mathbf{B}_{i,j} = 1\big] &= \mathbb{P}\big[\mathbf{A}_{i,j} = 1,\, \mathbf{B}_{i,j} = 0\big] \\ &= (1-\rho)\mathbf{\Lambda}_{i,j}(1-\mathbf{\Lambda}_{i,j}).\end{aligned}$$

Now, since

$$[\mathbf{X}]_{1,1} + [\mathbf{X}]_{2,2} = \sum_{i\neq 1}(\mathbf{A}_{i,1} - \mathbf{B}_{i,1})^2 + \sum_{i \neq 2}(\mathbf{A}_{i,2} - \mathbf{B}_{i,2})^2,$$

is the sum of $(n-1)+(n-1)$ Bernoulli random variables which are collectively independent—besides the two of them which are equal, namely $(\mathbf{A}_{12} - \mathbf{B}_{12})^2$ and $(\mathbf{A}_{21} - \mathbf{B}_{21})^2$—we have that $[\mathbf{X}]_{1,1} + [\mathbf{X}]_{2,2}$ is stochastically greater than or equal to a Binomial$\big(2n-3, 2(1-\rho)\alpha(1-\alpha)\big)$ random variable. Also note that

$$[\mathbf{X}]_{1,2} = \sum_{i \neq 1,2}(\mathbf{A}_{i,1} - \mathbf{B}_{i,1})(\mathbf{A}_{i,2} - \mathbf{B}_{i,2})$$

is the sum of $n-2$ independent random variables (namely, the $(\mathbf{A}_{i,1} - \mathbf{B}_{i,1})(\mathbf{A}_{i,2} - \mathbf{B}_{i,2})$'s) each with mean 0 and each taking on values in $\{-1, 0, 1\}$. Applying Lemma 3 and Lemma 6, respectively, to $\mathbf{X}_{11} + \mathbf{X}_{22}$ and to $\mathbf{X}_{12}$, with $t := (2n-3)2(1-\rho)\alpha(1-\alpha)/4$, yields

$$\begin{aligned}&\mathbb{P}\big(2[\mathbf{X}]_{1,2} \geq [\mathbf{X}]_{1,1} + [\mathbf{X}]_{2,2}\big) \\ &\qquad \leq \mathbb{P}\big(2[\mathbf{X}]_{1,2} \geq 2t\big) + \mathbb{P}\big([\mathbf{X}]_{1,1} + [\mathbf{X}]_{2,2} \leq 2t\big) \\ &\qquad \leq 2 \cdot e^{\frac{-2t^2}{2n-3}} + 2 \cdot e^{\frac{-t^2}{2(n-2)}} \leq e^{-cn},\end{aligned}$$

for some positive constant c (the last inequality holds when n is large enough). Hence the probability that Equation (4.19) holds is seen to decay exponentially in n, and is finitely summable over $n = 1, 2, 3, \ldots$. Therefore, by the Borel-Cantelli Lemma we have that almost always Equation (4.19) does not hold. Theorem 4.1, part b is now shown, since Equation (4.19) is a necessary condition for $\mathbf{I} \in \arg\min_{\mathbf{D}\in\mathcal{D}} \|\mathbf{AD} - \mathbf{DB}\|_F^2$. ■

4.2.5 On correlation testing after graph matching

If two graphs are not (or very weakly) correlated, it might not be useful to match them. After running a graph matching algorithm and obtaining a permutation matrix, we would like to determine whether the graphs are significantly correlated or not. If they were not, then the graph matching result might not be informative.

Let us consider here two ρ-correlated Erdős-Rényi graphs, with n vertices each and parameter (link probability) p, and let us also consider the statistic $T_n := \min_{\mathbf{P}\in\mathcal{P}} \|\mathbf{A} - \mathbf{PBP^T}\|_F^2$. Also define

$$c_n(\alpha) := \left(4\binom{n}{2}p(1-p)\left[\left(n + \frac{1}{2}\right)\log n - n + 1 + \log\alpha\right]\right)^{1/2}.$$

The following result allows us to decide whether the graphs were correlated from the statistic T_n.

Theorem 4.2. *With notation as above, for any $\alpha > 0$,*
a) If $\rho = 0$, then $P(T_n \leq 4\binom{n}{2}p(1-p) - 2c_n(\alpha)) \leq \alpha$.
b) If there exists a constant $c > 0$ such that if $\rho \geq c\sqrt{\frac{\log n}{n}}$, then for all but finitely many n, it holds

$$P\left(T_n \geq 4\binom{n}{2}p(1-p) - 2c_n(\alpha)\right) \leq exp\left\{-(1+o(1))\frac{n}{2}\log n\right\}.$$

This provides an implementable test of whether or not two graphs are significantly correlated. Estimate p via $\hat{p}$ and T_n via $\widehat{T}_n$. If $\widehat{T}_n > 4\binom{n}{2}\hat{p}(1-\hat{p}) - 2c_n(\alpha)$, then the correlation is insignificant. We experimentally illustrate this in the following section.

Proof. Part a) For the present, fix $\mathbf{P} \in \Pi(n)$. If $\rho = 0$,

$$||\mathbf{A} - \mathbf{PBP^T}||_F^2/2 \sim \text{Bin}\left(\binom{n}{2}, 2p(1-p)\right).$$

With a simple application of Hoeffding's inequality (see Theorem 3.3 of [24] for example) we have that

$$\begin{aligned}\mathbb{P}_{H_0}\left(||\mathbf{A} - \mathbf{PBP^T}||_F^2 \leq 4\binom{n}{2}p(1-p) - 2c_n(\alpha)\right) &\leq \exp\left\{-\frac{c_n(\alpha)^2}{4\binom{n}{2}p(1-p)}\right\}\\ &= \exp\left\{-\left[(n+\frac{1}{2})\log n - n + 1 + \log\alpha\right]\right\}\end{aligned}$$

A simple subadditivity bound combined with Sterling's formula yields

$$\begin{aligned}\mathbb{P}_{H_0}\Big(T_n \leq 4\binom{n}{2}p(1-p) - 2c_n(\alpha)\Big) &\leq n!\exp\left\{-\left[(n+\frac{1}{2})\log n - n + 1 + \log\alpha\right]\right\}\\ &\leq e\sqrt{n}\left(\frac{n}{e}\right)^n\exp\left\{-\left[(n+\frac{1}{2})\log n - n + 1 + \log\alpha\right]\right\}\\ &= \exp\left\{-\left[(n+\frac{1}{2})\log n - n + 1 + \log\alpha\right] + 1 + \frac{1}{2}\log n + n\log n - n\right\} = \alpha,\end{aligned}$$

as desired.
Part b) In [71], the authors proved that for a suitably chosen constant c', if $\rho \geq c'\sqrt{\frac{\log n}{n}}$ then

$$\underset{P\in\Pi(n)}{\text{argmax}}\,||\mathbf{AP} - \mathbf{PB}||_F^2 = \{\mathbf{P^*}\}\text{a.s.}$$

where $\mathbf{P^*}$ is the permutation matrix corresponding to the true but unknown alignment of the vertices. Therefore if ρ satisfies $\rho \geq c'\sqrt{\frac{\log n}{n}}$, then for all but finitely many n,

$$T_n/2 \sim \text{Bin}\Big(\binom{n}{2}, 2p(1-p)(1-\rho)\Big).$$

Therefore for all but finitely many n, again applying Hoeffding's inequality, we have that (where c is a constant satisfying $c > c' \vee 3/\sqrt{p(1-p)}$)

$$\begin{aligned}
&\mathbb{P}\left(T_n \geq 4\binom{n}{2}p(1-p) - 2c_n(\alpha)\middle|\rho \geq c\sqrt{\frac{\log n}{n}}\right)\\
&= \mathbb{P}\left(T_n \geq 4\binom{n}{2}p(1-p)(1-\rho) + 4\binom{n}{2}p(1-p)\rho - 2c_n(\alpha)\middle|\rho \geq c\sqrt{\frac{\log n}{n}}\right)\\
&\leq \exp\left\{-\frac{\left[2\binom{n}{2}p(1-p)\rho - c_n(\alpha)\right]^2}{2\left(2\binom{n}{2}p(1-p)(1-\rho) + \left[2\binom{n}{2}p(1-p)\rho - c_n(\alpha)\right]/3\right)}\right\}\\
&\leq \exp\left\{-\frac{\left[2\binom{n}{2}p(1-p)\rho - (1+o(1))2n^{3/2}\sqrt{p(1-p)\log n}\right]^2}{(1+o(1))2n^2p(1-p)}\right\}\\
&\leq \exp\left\{-\frac{\left[(1+o(1))n^{3/2}\sqrt{p(1-p)\log n}\right]^2}{(1+o(1))2n^2p(1-p)}\right\}\\
&= \exp\left\{-(1+o(1))\frac{1}{2}n\log n\right\}.
\end{aligned}$$

as desired □

We want to reinforce the importance of these two theorems. We demonstrated the validity (or more properly, invalidity) of a popular convex relaxation method in graph matching. The second theorem provides, for the first time, an explicit tool to verify if two graphs are correlated enough for a matching of the graphs to be informative.

Testing the correlation levels of two graphs

Theorem 4.2 provides an implementable test of whether or not two graphs are significantly correlated. Estimate p via $\hat{p}$ and T_n via $\widehat{T}_n$. If $\widehat{T}_n > 4\binom{n}{2}\hat{p}(1-\hat{p}) - 2c_n(\alpha)$, then the correlation is insignificant. This *a posteriori* test may provide insight into how much credence to put into the matching; indeed, if the correlation is insignificant, then the matching is less informative. For the following experiment, the graphs follow the correlated ER(5000,0.1) model. For each one of the different correlation values, we generated 20 pairs of graphs, and the energy $\|\mathbf{A}-\mathbf{B}\|_F^2$ is plotted in Figure 4.1. The horizontal line corresponds to the value in the statement of Theorem 4.2, namely, $4\binom{n}{2}p(1-p) - 2c_n(\alpha)$, for several values of α in $(0,1)$ (as $c_n(\alpha)$ is a lower order term, these lines are very close to each other and only one horizontal line can be seen in the figure). As it can be observed, when the value of T_n is above the threshold in red, the graphs are poorly correlated. As an aside note, the behavior of T_n seems to be linear with ρ.

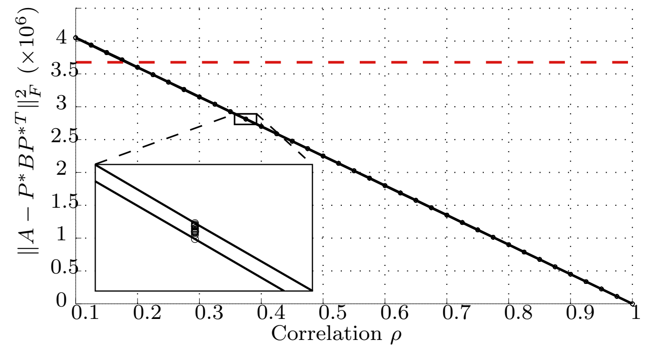


Figure 4.1: Value of the statistic T_n: 20 runs for each correlation (black dots), and solid black lines for maximum and minimum values over all runs. Threshold in Theorem 4.2 in dashed red.

4.3 Deterministic Results for Graph Matching

Problems related to graph matching and isomorphisms are very important both from a theoretical and practical perspective, with applications ranging from image and video analysis to biological and biomedical problems. The graph matching problem is challenging from a computational point of view, and therefore different relaxations are commonly used. Although common relaxations techniques tend to work well for matching perfectly isomorphic graphs, it is not yet fully understood under which conditions the relaxed problem is guaranteed to obtain the correct answer.

In this paper we prove that the graph matching problem and its most common convex relaxation, where the matching domain of permutation matrices is substituted with its convex hull of doubly-stochastic matrices, are equivalent for a certain class of graphs, such equivalence being based on spectral properties of the corresponding adjacency matrices. We also derive results about the automorphism group of a graph, and provide fundamental spectral properties of the adjacency matrix.

4.3.1 Introduction

The theoretical and computational aspects behind graph isomorphisms and graph matching have been a great challenge for the scientific community for a long time. Maybe the easiest problem to state from this category is the graph isomorphism problem, which consists in determining whether two given graphs are isomorphic or not, that is, if there exists a bijection between the vertex sets of the graphs, preserving the edge structure. Besides the theoretical analysis, the graph isomorphism problem is also very interesting from the computational complexity point of view, since its complexity class is still unsolved: it is one of the few problems in NP not yet classified as P nor NP-complete [25].

The concept of graph automorphism, and its related properties, is closely connected to the graph isomorphism problem. An automorphism of a graph is a mapping from its vertex set onto itself, preserving the connectivity structure. The set of automorphisms forms a group under the composition operation. Of course, the identity map is always an automorphism, and when this is the only element in the group, we say that the graph has a trivial automorphism group. From the computational complexity point of view, computing the automorphism group is at least as difficult as solving the graph isomorphism problem.

The last problem we wish to discuss here is the so-called graph matching problem, which consists in finding an isomorphism between two graphs, and it is therefore harder than the graph isomorphism problem. Specifically, let G_A and G_B be two graphs with n vertices, and let $\mathbf{A}$ and $\mathbf{B}$ be their corresponding adjacency matrices. A common statement of the graph matching problem is to find the correspondence between the nodes of G_A and G_B which minimizes some matching error. In terms of the corresponding adjacency matrices $\mathbf{A}$ and $\mathbf{B}$, which encode the graph connectivity, this corresponds to finding a matrix $\mathbf{P}$ in the set of permutation matrices $\mathcal{P}$, such that it minimizes a given distance between $\mathbf{A}$ and $\mathbf{PBP^T}$. A common choice is the Frobenius norm $||\mathbf{A} - \mathbf{PBP^T}||_F^2$, and then the graph matching problem can be formally stated as

$$\min_{\mathbf{P}\in\mathcal{P}} ||\mathbf{A} - \mathbf{PBP^T}||_F^2 = \min_{\mathbf{P}\in\mathcal{P}} ||\mathbf{AP} - \mathbf{PB}||_F^2. \qquad (P_1)$$

Although polynomial algorithms have been developed for a few special types of graphs, like trees or planar graphs for example [25], the combinatorial nature of the permutation search makes this problem NP in general. As such, there are several and diverse techniques addressing the graph matching problem, including spectral methods [92] and relaxations techniques [39, 95, 102].

In this paper we focus on a particular and very common relaxation technique, which consists in relaxing the feasible set (the set of permutation matrices) to its convex hull. By virtue of the Birkhoff-von Neuman theorem, the convex hull of $\mathcal{P}$ is the set of doubly stochastic matrices $\mathcal{D} = \{\mathbf{M} \in \mathbb{R}^{n\times n} : \mathbf{M}_{ij} \geq 0, \mathbf{M1} = \mathbf{1}, \mathbf{M}^T\mathbf{1} = \mathbf{1}\}$, that is, the set of matrices with non-negative entries such that each row and column sum up to one.

The relaxed version of the problem is then

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}\in\mathcal{D}} ||\mathbf{AP} - \mathbf{PB}||_F^2, \qquad (P_2)$$

which is a convex problem. However, the resulting $\hat{\mathbf{P}}$ is a doubly stochastic matrix and not necessarily a permutation matrix, or in general the solution to (P_1).

Indeed, since the feasible set of problem (P_2) is the convex hull of the feasible set of problem (P_1), every solution of the first problem is also a solution of the relaxed graph matching problem. A very important question is under which hypothesis the solution set of these two problems (P_1) and (P_2) coincide. It is easy to see that, if there are two permutation matrices that solve the graph matching problem, then every matrix on the straight line joining them is a solution of problem (P_2), since this problem is convex. Therefore, the least that one should ask for these two problems to be equivalent is for the solution of (P_1) to be unique. When the two graphs are isomorphic, this is equivalent to asking for the automorphism group of the graphs to be the trivial group.

A probabilistic analysis of this equivalence between the original and the relaxed graph matching problems is provided in Section 4.2. Basically, we proved that, when two graph are correlated (but not necessarily isomorphic), then the unique solution of a non-convex relaxation is almost always the correct permutation matrix; while on the other hand, the underlying alignment is almost always not a solution of the commonly used convex relaxation, where the permutation set is replaced by the doubly-stochastic set as above.

On the other hand, in [1] the authors prove the equivalence of the original graph matching problem and a relaxed version for a particular kind of graphs which they call *friendly*, based on spectral properties. In this work, we extend these results, proving the (deterministic) equivalence for a larger set of graphs, and also shedding light on some new spectral graph properties.

4.3.2 Main result

In this section, we consider two isomorphic graphs G_A and G_B with n vertices each, and adjacency matrices $\mathbf{A}$ and $\mathbf{B}$ respectively. Let $\mathbf{P_o} \in \mathcal{P}$ be the permutation matrix associated to the isomorphism between the two graphs, that is, $\mathbf{B} = \mathbf{P_o A P_o^T}$.

Since the graphs considered here are isomorphic, then the minimum (either over $\mathcal{D}$ or $\mathcal{P}$) of $\|\mathbf{AP} - \mathbf{PB}\|_F^2$ is zero, and it is achieved (at least) at $\mathbf{P_o}$. Both problems can be then re-stated as solving the set of linear equations $\mathbf{AP} = \mathbf{PB}$ over $\mathbf{P} \in \mathcal{P}$ or $\mathbf{P} \in \mathcal{D}$.

Now, consider that by the simple change of variables $\mathbf{Q} = \mathbf{PP_o}$. Then for any solution $\mathbf{P}$ to the relaxed problem (P_2), it holds that

$$\mathbf{AP} = \mathbf{PB} \iff \mathbf{AP} = \mathbf{PP_o A P_o^T} \iff \mathbf{APP_o} = \mathbf{PP_o A} \iff \mathbf{AQ} = \mathbf{QA}. \quad (4.20)$$

Note that the change of variables is a multiplication by a permutation matrix, and hence the set of doubly stochastic matrices is invariant under this mapping. Therefore, any solution to $\mathbf{AQ} = \mathbf{QA}$ over $\mathbf{Q} \in \mathcal{D}$ leads, via the change of variables, to a solution of $\mathbf{AP} = \mathbf{PB}$ with $\mathbf{P} \in \mathcal{D}$. This allows us to state the equivalency between both problems (P_1) and (P_2) using only one of the adjacency matrices. Specifically, the problem $\mathbf{AQ} = \mathbf{QA}$ with $\mathbf{Q} \in \mathcal{D}$ has a trivial solution $\mathbf{Q} = \mathbf{I}$, which corresponds to the solution $\mathbf{P_o}$ of the problem $\mathbf{AP} = \mathbf{PB}$ with $\mathbf{P} \in \mathcal{D}$. Then the matrix $\mathbf{P_o}$ will be the unique solution of problem (P_2) if and only if the identity is the unique solution of $\mathbf{AQ} = \mathbf{QA}$ with $\mathbf{Q} \in \mathcal{D}$.

Now, since $\mathbf{A}$ is a symmetric matrix, we can consider its spectral decomposition $\mathbf{A} = \mathbf{UDU^T}$, where $\mathbf{D}$ is a diagonal matrix containing the eigenvalues and $\mathbf{U}$ is an orthonormal matrix containing the eigenvectors as columns, denoted as u_i, for $i = 1 \dots n$.

The main result of [1] states that if $\mathbf{A}$ has no repeated eigenvalues, and no eigenvector u_i is perpendicular to the vector of ones $\mathbf{1}$, then problems (P_1) and (P_2) are equivalent. This is illustrated in Figure 4.2, where some graph properties are represented. Here *asymmetric* means that the authomorphism group of the graph is trivial, *simple spectrum* means that the adjacency matrix has no repeated eigenvalues, *non-orthogonal to* $\mathbf{1}$ means that no eigenvector u_i verifies $u_t^T\mathbf{1} = 0$, and the *regular* circle contains regular graphs, i.e., graphs such that each vertex has the same number of neighbors. The intersection of *simple spectrum* and *non-orthogonal to* $\mathbf{1}$ graphs is what the authors of [1] call *friendly* graphs, and they prove the equivalence of problems (P_1) and (P_2) for this class.

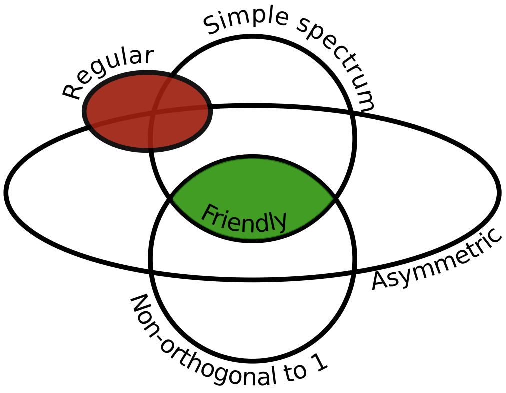


Figure 4.2: Graph classes and equivalence of problems (P_1) and (P_2). Graphs with trivial authomorphism group are represented in the *asymmetric* set (we know that graphs for which (P_1) and (P_2) are equivalent are inside this set), graphs whose adjacency matrices have no repeated eigenvalues are represented as the *simple spectrum* set, *non-orthogonal to* $\mathbf{1}$ means that no eigenvector u_i verifies $u_t^T\mathbf{1} = 0$, and the *regular* circle contains regular graphs. Graphs in the intersection of *simple spectrum* and *non-orthogonal to* $\mathbf{1}$ are called *friendly* graphs, and here the equivalence of problems (P_1) and (P_2) holds [1]. These problems are not equivalent for regular graphs. A key question addressed in this work is how far we can extend the green zone of equivalence inside the asymmetric set.

As observed above, a necessary condition for problems (P_1) and (P_2) to be equivalent, is for the automorphism group of the graph to be the trivial group. However, this condition is not sufficient. Take for instance a regular graph, and denote by $\mathbf{J}$ the barycenter of the set of doubly stochastic matrices, $\mathbf{J} = \frac{1}{n}\mathbf{1}\mathbf{1}^T$. Hence, it is very easy to see that, if $\mathbf{A}$ is the adjacency matrix of a regular graph, then $\mathbf{AJ} = \mathbf{JA}$. Therefore, there is a solution to problem (P_2) which is not a permutation matrix. Since there are regular graphs with trivial automorphism group (like the Frucht graph [49] for instance), then this condition cannot be sufficient. In Figure 4.2, this is represented with the small red circle, which intersects both the asymmetric and simple spectrum sets (see Section 4.3.6 for examples of graphs in each intersection). In summary, we can hope for problems (P_1) and (P_2) to be equivalent inside the *asymmetric* set minus the regular graphs. So far, we know from [1] that this is true for *friendly* graphs, being this until now the largest known class for which the relaxation is equivalent to the original problem.

The next theorems extend the set where these problems are equivalent to a larger set of graphs. Theorem 4.4 is stronger than Theorem 4.3, but we include both proofs for the sake of clarity, since both have pedagogic value.

Theorem 4.3. *If* $\mathbf{A}$ *has no repeated eigenvalues (simple spectrum), and there are* k *eigenvectors* u_i *such that* $u_i^T\mathbf{1} = 0$*, each one of these vectors having at least* $2k+1$ *nonzero entries, then problems* (P_1) *and* (P_2) *are equivalent.*

Proof. We want to prove that the identity is the unique solution to the problem $\mathbf{AQ} = \mathbf{QA}$ for $\mathbf{Q} \in \mathcal{D}$. Let us write the equality $\mathbf{AQ} = \mathbf{QA}$ in terms of the eigenvector decomposition of $\mathbf{A}$:

$$\mathbf{AQ} = \mathbf{QA} \Leftrightarrow \mathbf{UDU^TQ} = \mathbf{QUDU^T} \Leftrightarrow$$
$$\mathbf{U^TUDU^TQU} = \mathbf{U^TQUDU^TU} \Leftrightarrow \mathbf{DU^TQU} = \mathbf{U^TQUD}.$$

Now, let us denote by $\mathbf{F}$ the new unknown matrix $\mathbf{F} = \mathbf{U^TQU}$. The problem can be now stated as

$$\mathbf{DF} = \mathbf{FD} \quad , \quad \mathbf{UFU^T} \in \mathcal{D}, \tag{P_F}$$

and we now want to prove that $\mathbf{F} = \mathbf{I}$ is the unique solution of this last problem.

It is easy to see that, since $\mathbf{D}$ is diagonal with no repeated entries in the diagonal, then $\mathbf{F}$ has to be diagonal as well in order to commute with $\mathbf{D}$.

Let us write the conditions for $\mathbf{UFU^T}$ to be in $\mathcal{D}$:

c1) $\mathbf{UFU^T1} = \mathbf{1}$,

c2) $\mathbf{UF^TU^T1} = \mathbf{1}$,

c3) $\left(\mathbf{UFU^T}\right)_{i,j} \geq 0\,, \forall i, j.$

Since $\mathbf{F}$ is diagonal, and in particular $\mathbf{F} = \mathbf{F^T}$, then the first two conditions are redundant, and one of them can be eliminated. Left-multiplying the first condition by $\mathbf{U^T}$, we obtain $\mathbf{FU^T1} = \mathbf{U^T1}$, and calling $\mathbf{v} = \mathbf{U^T1}$, condition c1) can be written as $\mathbf{Fv} = \mathbf{v}$.

Without loss of generality, we can assume that the k eigenvectors u_i satisfying $u_i^T\mathbf{1} = 0$ are the first k columns in $\mathbf{U}$. Therefore, $\mathbf{v}_i = 0$ for $i = 1 \dots k$, and $\mathbf{v}_i \neq 0$ for $i = k+1 \dots n$. As $\mathbf{F}$ is diagonal, the equations $\mathbf{Fv} = \mathbf{v}$ can be easily written as $\mathbf{F}_{i,i}\mathbf{v}_i = \mathbf{v}_i$. When $\mathbf{v}_i \neq 0$, the only way for this equation to hold is when $\mathbf{F}_{i,i} = 1$. This means that $\mathbf{F}_{i,i} = 1$ for $i = k+1 \dots n$, and this is sufficient to guarantee that the first two conditions hold.

For analyzing the third condition, let us decompose the matrix product using that $\mathbf{F}$ is a diagonal matrix with $\mathbf{F}_{i,i} = 1$ for $i = k+1 \dots n$:

$$\mathbf{UFU^T} = \sum_{i=1}^{n} u_i \mathbf{F}_{i,i} u_i^T = \sum_{i=1}^{k} u_i \mathbf{F}_{i,i} u_i^T + \sum_{i=k+1}^{n} u_i u_i^T .$$

We can now add and subtract $\sum_{i=1}^{k} u_i u_i^T$, leading to

$$\mathbf{UFU^T} = \sum_{i=1}^{k} u_i (\mathbf{F}_{i,i} - 1) u_i^T + \sum_{i=1}^{n} u_i u_i^T = I + \sum_{i=1}^{k} u_i (\mathbf{F}_{i,i} - 1) u_i^T .$$

Let us denote by $\mathbf{L} = \sum_{i=1}^{k}(1 - \mathbf{F}_{i,i})u_i u_i^T$, and therefore $\mathbf{UFU^T} = \mathbf{I} - \mathbf{L}$.

Observe that the matrix $\mathbf{L}$ satisfies $\mathbf{L1} = \mathbf{0}$, since every vector u_i participating in the sum satisfies $u_i^T\mathbf{1} = 0$; and note also that all the elements in the diagonal are $\mathbf{L}_{j,j} \geq 0$, otherwise the corresponding entry of $\mathbf{UFU^T}$ would be $(\mathbf{UFU}^T)_{j,j} > 1$, violating the doubly stochastic condition.

Now, let us assume that there is a solution $\mathbf{F}$ to problem (P_F) different from the identity, and let us analyze the corresponding $\mathbf{L}$ matrix trying to find a contradiction. The condition c3) dictates that $(\mathbf{I} - \mathbf{L})_{i,j} \geq 0$ for all i, j, therefore the $\mathbf{L}$ matrix has no positive elements off the diagonal. On the other hand, since $\mathbf{F}$ is diagonal and we have assumed $\mathbf{F} \neq \mathbf{I}$, then at least one of the values $\mathbf{F}_{i,i}$ $(i \leq k)$ is different from 1. The corresponding eigenvector u_i, which has at least $2k + 1$ non-zero elements by hypothesis, will be actually used in the summation constructing $\mathbf{L}$, and therefore this guarantees that at least $2k + 1$ elements in the diagonal of $\mathbf{L}$ are strictly positive.

Considering then the following just described properties for the $\mathbf{L}$ matrix:

- $\mathbf{L1} = \mathbf{0}$,
- $\mathbf{L}_{i,j} \leq 0$ for all $i \neq j$,
- $\mathbf{L}_{i,i} \geq 0$ for all $i = 1 \dots n$,
- $\mathbf{L} = \mathbf{L}^T$.

we can associate an undirected graph G_L such that $\mathbf{L}$ is its Laplacian matrix.[2] Moreover, since at least $2k + 1$ diagonal elements of $\mathbf{L}$ are non-zero (and strictly positive), at least $2k + 1$ elements off the diagonal are non-zero (and strictly negative), since each row has to add up zero.

Now, each off diagonal element of the laplacian matrix $\mathbf{L}$ corresponds to an edge of the graph G_L. Since the matrix $\mathbf{L}$ is symmetric, the graph G_L is undirected, and each edge appears twice in the Laplacian matrix. Since there are at least $2k+1$ non-zero elements off the diagonal of $\mathbf{L}$, the auxiliary graph G_L has at least $\left\lceil \frac{2k+1}{2} \right\rceil = k + 1$ edges. It is easy to see that, if the number of edges is $e \geq k + 1$, then the auxiliary graph G_L has at most $C \leq n - (k + 1)$ connected components.

Remembering that the number of connected components C is given by the multiplicity of the 0 eigenvalue of the Laplacian matrix $\mathbf{L}$, then the rank of $\mathbf{L}$ has to be at least $rank(\mathbf{L}) = n - C \geq k + 1$.

However, by construction, $\mathbf{L}$ is the sum of k rank-one matrices, and therefore $rank(\mathbf{L}) \leq k$, which is a contradiction. This proves that the only solution to problem (P_F) is the identity, concluding the proof of the theorem. □

[2] Given a graph with adjacency matrix $\mathbf{M}$, its Laplacian matrix is defined as $\mathbf{L} = \mathbf{S} - \mathbf{M}$, where $\mathbf{S}$ is the degree matrix, i.e., a matrix having the degree of each node in the corresponding diagonal element, and zeros elsewhere.

This proof, and therefore the class of equivalence between problems (P_1) and (P_2), can be further extended by noting that we originally asked for every eigenvector orthogonal to $\mathbf{1}$ to have at least $2k+1$ non-zero elements, but in the proof we only used this fact for one of these eigenvectors. It might be the case, for instance, that only one of the elements $\mathbf{F}_{i,i}$ is different from one, and therefore the matrix $\mathbf{L}$ has rank one. In this case, it would be sufficient to ask for the corresponding eigenvector to have at least 3 non-zero elements. The problem is that we do not know in advance which or how many of the eigenvectors will be used in the summation to construct the $\mathbf{L}$ matrix. However, it is possible to weaken the hypothesis as shown next.

Let us consider all the k eigenvectors satisfying $u_i^T\mathbf{1} = 0$, and sort them according to the number of non-zero elements, such that $|u_1|_0 \leq |u_2|_0 \leq \cdots \leq |u_k|_0$, where $|\cdot|_0$ is the ℓ_0 pseudo-norm, which counts the non-zero elements of a vector.

Theorem 4.4. *If* $\mathbf{A}$ *has no repeated eigenvalues, and there are* k *eigenvectors* u_i *such that* $u_i^T\mathbf{1} = 0$*, sorted as above and satisfying* $|u_i|_0 \geq 2i+1$*, then problems* (P_1) *and* (P_2) *are equivalent.*

Proof. This proof follows exactly the same procedure as the previous one, with minor changes.

Let us assume, as before, that there is a solution $\mathbf{F} \neq \mathbf{I}$ to problem (P_F). In order to fulfill condition c1), the last $n-k$ diagonal elements of F have to be one, i.e., $\mathbf{F}_{i,i} = 1$ for $i = k+1, \ldots n$. For the first k diagonal elements, there might be some 0 values. Let M be the greatest index of the eigenvectors actually used in the sum, meaning $M = \max\{i \in 1 \ldots k : \mathbf{F}_{i,i} \neq 0\}$.

We can then write

$$\mathbf{L} = \sum_{i=1}^{M}(1-\mathbf{F}_{i,i})u_i u_i^T.$$

Since $|u_M|_0 \geq 2M+1$, the auxiliary graph G_L has at least $M+1$ edges. Therefore the number of connected components satisfies $C \leq n-(M+1)$, and hence $rank(\mathbf{L}) \geq M+1$. The contradiction, as before, comes from the fact that $\mathbf{L}$ is the sum of M rank one matrices, concluding the proof. □

As noted above, a necessary condition for the problems (P_1) and (P_2) to be equivalent is for the automorphism group of G_A to be the trivial group. Therefore, we have the following corollary:

Corollary 4.5. *If* $\mathbf{A}$ *has no repeated eigenvalues, and there are* k *eigenvectors* u_i *such that* $u_i^T\mathbf{1} = 0$*, sorted according to their* ℓ_0 *norm as above, and satisfying* $|u_i|_0 \geq 2i+1$*, then the automorphism group of the corresponding graph* G_A *is the trivial group.*

4.3.3 Interpretation and additional results

It is clear that the spectral decomposition of an adjacency matrix provides a lot of information about the automorphism group of a graph, and the graph matching problem itself. However, very little is known about how the eigenvalues and eigenvectors of the adjacency matrix affect the graph properties. In this section, we discuss some links between these two fields, paying particular attention to the equivalence of problems (P_1) and (P_2), and also discussing more general novel properties.

As noted in the previous section, asymmetry of a graph is a necessary condition for problems (P_1) and (P_2) to be equivalent, although is not sufficient, with asymmetric regular graphs serving as counter-examples (see red region in Figure 4.2). It is interesting to note that regular graphs have the vector $\mathbf{1}$ as an eigenvector, and since the adjacency matrix is symmetric, its eigenvectors are orthogonal to each other, therefore there are $n-1$ eigenvectors satisfying $u_i^T\mathbf{1} = 0$. Hence, the condition asked in [1] is violated not only by one eigenvector, but by $n-1$ of them.

Besides this observation, it is not clear the interpretation of the non existence of eingenvectors perpendicular to $\mathbf{1}$.

Let us focus now on the properties of eigenvectors orthogonal to $\mathbf{1}$ with restricted support, as in the statement of Theorem 4.3.

4.3.4 The simplest case: one eigenvector u such that $u^T\mathbf{1} = 0$

Let us assume here that $\mathbf{A}$ is the adjacency matrix of a graph G_A with no repeated eigenvalues and only one eigevector u satisfying $u^T\mathbf{1} = 0$. Now, if this vector u has strictly more than two non-zero entries, i.e., $|u|_0 > 2$, then this graph falls into the hypothesis of Theorem 4.3. Therefore, the graph has trivial automorphism group, and if G_B is an isomorphic graph, problems (P_1) and (P_2) are equivalent.

Since the sum of the entries of u is zero, the only remaining case is when u has exactly two non-zero elements. Assuming that the eigenvectors are normalized, the eigenvector u is of the form

$$u = \left(0, \dots, 0, \frac{1}{\sqrt{2}}, 0, \dots, 0, \frac{-1}{\sqrt{2}}, 0, \dots, 0\right).$$

Let s and t be the indices of the non-zero coefficients. Since u is an eigenvalue, then we have $\mathbf{A}u = \lambda u$. Now, denoting by $\mathbf{A}_s$ and $\mathbf{A}_t$ the columns of $\mathbf{A}$ at positions s and t respectively, and taking into account the particular structure of u, the product $\mathbf{A}u$ is simply the difference between these two columns: $\mathbf{A}u = \frac{1}{\sqrt{2}}(\mathbf{A}_s - \mathbf{A}_t) = \lambda u$. Therefore, columns $\mathbf{A}_s$ and $\mathbf{A}_t$ are identical, except for the coordinates s and t. This means that the nodes corresponding to indices s and t have exactly the same connectivity pattern with the rest of the nodes in the graph.

Consider now the rest of the involved entries (nodes). Let $\tilde{\mathbf{A}}$ be the 2×2 sub-matrix formed by entries (s,s), (s,t), (t,s) and (t,t) of matrix $\mathbf{A}$, and let $w = \left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}\right)$. We have then $\tilde{\mathbf{A}}w = \lambda w$. It is easy to see that, since the entries in $\tilde{\mathbf{A}}$ are either 1 or 0, then only three values of λ are possible: -1, 0 and 1, corresponding to the following situations:

- $\lambda = -1$: the matrix is $\tilde{\mathbf{A}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, therefore nodes s and t are connected and they have no loops;

- $\lambda = 0$: the matrix is either $\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ or $\tilde{\mathbf{A}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$. Therefore, nodes s and t are either connected and both have loops, or not connected without loops;
- $\lambda = 1$: the matrix is $\tilde{\mathbf{A}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, therefore nodes s and t are not connected and both of them have loops.

Taking into account that nodes s and t have the same connectivity pattern with the rest of the graph, in any of the situations listed above, nodes s and t are interchangeable, meaning that there exists a non trivial automorphism of the graph G_A, namely, the automorphism which permutes nodes s and t, and leaves the rest of the nodes unchanged. Therefore, for graphs with the corresponding adjacency matrix having a single eigenvector orthogonal to the unity vector, and this eigenvector having exactly two non-zero entries, problems (P_1) and (P_2) are not equivalent. The problems are equivalent if the eigenvector has more than two non-zero entries.

4.3.5 The general case

Let us further analyze the relationship between the group of automorphisms of a graph and the eigenvectors of its adjacency matrix, now considering a more general case in terms of the non-zero elements of the eigenvectors.

First, observe that if the matrix $\mathbf{A}$ has simple spectrum, then each element of the automorphism group has order two, with the exception of the identity (this result appears in [16] and [67]):

Lemma 4.6. *If $\mathbf{A}$ has no repeated eigenvalues and $\mathbf{P}$ is a permutation matrix such that*
$\mathbf{AP} = \mathbf{PA}$, then $\mathbf{P}^2 = \mathbf{I}$.

Proof. In order to prove this, let u be an eigenvector of $\mathbf{A}$ associated with the eigenvalue λ. Then, $\mathbf{AP}u = \mathbf{PA}u = \mathbf{P}\lambda u = \lambda \mathbf{P}u$. Therefore, the vector $\mathbf{P}u$ is an eigenvector associated with the eigenvalue λ as well. Since every eigenspace has dimension 1, and the multiplication by the permutation matrix preserves the norm, then necessarily $\mathbf{P}u = \pm u$, and hence $\mathbf{P}^2 u = u$. Since this is true for every eigenvector u in the basis, then $\mathbf{P}^2 = \mathbf{I}$. □

We are now able to prove the following.

Proposition 4.7. *If $\mathbf{A}$ has no repeated eigenvalues, and the group of automorphisms of G_A is non trivial, then there exist a set of k eigenvectors u_i satisfying $u_i^T \mathbf{1} = 0$, each one of them having at most $2k$ non-zero entries.*

Proof. Let $\mathbf{P} \neq \mathbf{I}$ be a permutation matrix, corresponding to a non-trivial automorphism of G_A. As observed above, since $\mathbf{A}$ has simple spectrum, then $\mathbf{P}^2 = \mathbf{I}$. Since the permutation has order two, we can re-arrange the order of the nodes in such a way that the resulting permutation matrix $\mathbf{P}$ is block diagonal as follows

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & & & & & & \mathbf{0} \\ 1 & 0 & & & & & & \\ & & \ddots & & & & & \\ & & & 0 & 1 & & & \\ & & & 1 & 0 & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ \mathbf{0} & & & & & & & 1 \end{pmatrix}.$$

As in the previous section, consider the eigen-decomposition $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^{\mathbf{T}}$, which transforms the problem $\mathbf{AP} = \mathbf{PA}$ into $\mathbf{DF} = \mathbf{FD}$, where the new unknown matrix $\mathbf{F}$ is defined as $\mathbf{F} = \mathbf{U}^{\mathbf{T}}\mathbf{P}\mathbf{U}$, or equivalently, $\mathbf{P} = \mathbf{U}\mathbf{F}\mathbf{U}^{\mathbf{T}}$. As before, since $\mathbf{A}$ has no repeated eigenvalues, $\mathbf{F}$ is necessarily diagonal, and therefore $\mathbf{P} = \mathbf{U}\mathbf{F}\mathbf{U}^{\mathbf{T}}$ is one possible eigen-decomposition of $\mathbf{P}$.

Now, the matrix $\mathbf{U}$ of normalized eigenvectors of $\mathbf{A}$ is unique, up to changes of sign in each column. This is not true for $\mathbf{P}$, since it has repeated eigenvalues. However, any orthogonal eigen-decomposition of $\mathbf{P}$ can be obtained as an orthogonal transformation (rotation and/or symmetries) of $\mathbf{U}$.

Given that $\mathbf{P}$ is block-diagonal, one possible eigen-decomposition can be obtained by combining the eigen-decompositions of each block. The lower part of $\mathbf{P}$ is an identity block, and hence all eigenvalues are equal to 1, with canonical eigenvectors. The rest are 2×2 blocks with the following decomposition:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}.$$

Therefore, a plausible eigendecomposition for $\mathbf{P}$ is $\mathbf{P} = \mathbf{V}\mathbf{E}\mathbf{V}^{\mathbf{T}}$, with:

$$\mathbf{E} = \begin{pmatrix} -1 & & & & & & & \mathbf{0} \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & -1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ \mathbf{0} & & & & & & & 1 \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & & & & & & \mathbf{0} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & & & & & & \\ & & \ddots & & & & & \\ & & & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & & & \\ & & & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ \mathbf{0} & & & & & & & 1 \end{pmatrix}.$$

Since the columns of both $\mathbf{U}$ and $\mathbf{V}$ are possible basis of the eigenvalues of $\mathbf{P}$, the matrix $\mathbf{U}$ can be thought as an orthogonal transformation of $\mathbf{V}$ that leaves invariant the eigenspaces S_{-1} and S_1 (eigenspaces of $\mathbf{P}$ associated with eigenvalues -1 and 1 respectively). Observe that the eigenspace S_{-1} is composed by vectors orthogonal to $\mathbf{1}$, and therefore, in the orthogonal transformation from $\mathbf{V}$ to $\mathbf{U}$, the whole subspace S_{-1} will be mapped to a subspace orthogonal to $\mathbf{1}$. Let k be the dimension of the subspace S_{-1}, and let us denote by $\tilde{\mathbf{U}}$ the set of the k eigenvectors of $\mathbf{A}$ (columns of $\mathbf{U}$) corresponding to the eigenspace S_{-1} after the linear mapping. These columns of $\mathbf{U}$, as argued above, are orthogonal to $\mathbf{1}$. Analogously, let $\tilde{\mathbf{V}}$ be formed by the columns of $\mathbf{V}$ associated with the eigenvalue -1, so the columns of $\tilde{\mathbf{V}}$ are a basis of S_{-1}.

Given that we assumed $\mathbf{P} \neq \mathbf{I}$, there is at least one 2×2 non identity block like the one described above, and therefore $k \geq 1$. Since S_{-1} is invariant under the orthogonal transformation, then the mapping of this subspace can be written as linear combinations of the elements of the basis, this is, $\tilde{\mathbf{U}} = \tilde{\mathbf{V}}\mathbf{T}$, where $\mathbf{T}$ is an orthogonal matrix. Since, according to the previous description, $\tilde{\mathbf{V}}$ has the form

$$\tilde{\mathbf{V}} = \frac{1}{\sqrt{2}} \underbrace{\begin{pmatrix} -1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & -1 \\ 0 & 0 & \dots & 1 \\ \vdots & \vdots & & \vdots \end{pmatrix}}_{k \text{ columns}},$$

then $\tilde{\mathbf{U}}$ is conformed by k vectors with $2k$ non-zero entries at most. Moreover, each one of these vectors has an even number of non-zero entries. □

We have simulated millions of Erdős-Rényi graphs, and obtained have empirical evidence suggesting that there is always a subset of these k vectors formed by r vectors with exactly $2r$ non-zero entries each, in the same location, which correspond to the $2r$ nodes which are permuted in the automorphism (see the examples in the appendix). However, the arguments used in the previous proof are not sufficient to formally prove this.

On the other hand, the empirical evidence also suggests that a converse of this last statement may be true. We formulate then the following conjecture.

Conjecture 4.8. *If $\mathbf{A}$ has no repeated eigenvalues, and there exist a set of r eigenvectors u_i satisfying $u_i^T\mathbf{1} = 0$, each one of them with exactly $2r$ non-zero entries, in the same location, then the group of automorphisms of G_A is not trivial.*

We know that this is true for $k = 1$, Section 4.3.4. The proof for the general case, as well as other relations between spectral properties and the automorphism group, are part of future work.

4.3.6 Graph examples

Figure 4.2 shows different sets of graphs according to the relevant characteristics for this paper, principally about eigenvectors and eigenvalues. For instance, the class of graphs where theorems 4.3 and 4.4 apply lays on the intersection of asymmetric and simple spectrum graphs, but outside the *non-orthogonal to* $\mathbf{1}$ set. It is important therefore to show that there exist graphs in this subset, and in general that each subset in the diagram is not empty.

As mentioned above, the Frucht graph [49], illustrated in Figure 4.3 (left), serves as an example of regular graphs with trivial automorphism group and simple spectrum. The regular graph in Figure 4.3 (right) also has trivial automorphism group, but the adjacency matrix has repeated eigenvalues.

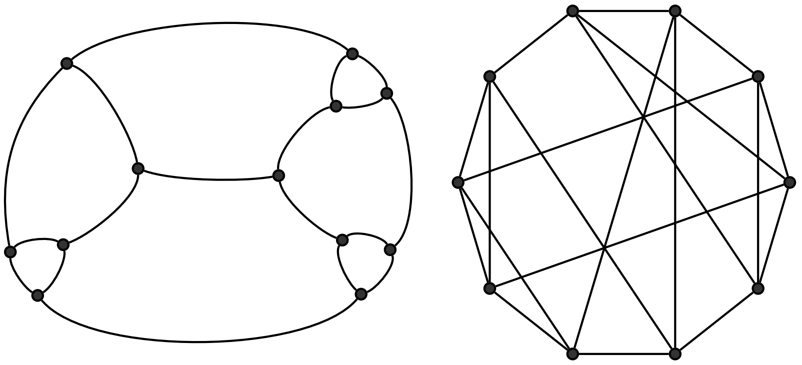


Figure 4.3: Regular graphs with trivial automorphism group. Left: the Frucht graph, with simple spectrum. Right: a regular asymmetric graph with repeated eigenvalues.

The graph of Figure 4.4 is an example of a graph with simple spectrum, but where there is an eigenvector u such that $u^T\mathbf{1} = 0$, and therefore this is not a *friendly* graph. This eigenvector has 4 non-zero elements, and hence Theorem 4.3 applies. As a consequence, for any isomorphic graph, problems (P_1) and (P_2) are equivalent, and in particular the automorphism group of this graph is trivial. Since the graph is not *friendly*, the results of [1] do not hold for this graph.

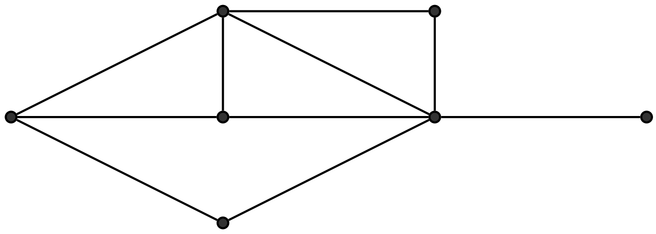


Figure 4.4: A non-friendly graph, with simple spectrum but one eigenvector orthogonal to $\mathbf{1}$.

The graph in Figure 4.5 has simple spectrum but non-trivial automorphism group. Indeed, it has two eigenvectors u_1 and u_2 orthogonal to $\mathbf{1}$, each one with four non-zero elements. Namely, the eigenvectors are

$$u_1 = \left(\frac{1}{2}, \frac{1}{2}, \frac{-1}{2}, \frac{-1}{2}, 0, 0, 0, 0\right)$$

and

$$u_2 = \left(\frac{1}{2}, \frac{-1}{2}, \frac{1}{2}, \frac{-1}{2}, 0, 0, 0, 0\right).$$

Illustrating the Conjecture 4.8, the first four coordinates of the eigenvectors correspond to the four red nodes of the graph. The non-trivial automorphism consist on permuting the two lower nodes between themselves, and the two upper nodes between themselves, as it can be clearly seen in the figure. Of course, theorems 4.3 and 4.4 do not apply for this graph.

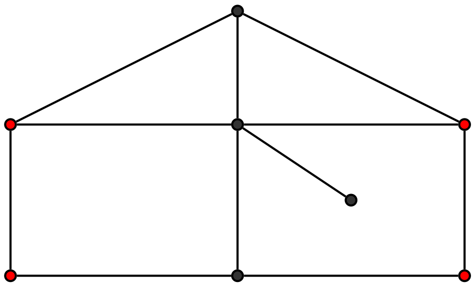


Figure 4.5: A graph with simple spectrum but non-trivial automorphism group.

The following two graphs, illustrated in Figure 4.6, have trivial automorphism group and both have repeated eigenvalues. The first one has one eigenvector orthogonal to $\mathbf{1}$, while the second one has no eigenvector u satisfying $u^T\mathbf{1} = 0$.

Finally, Figure 4.7 shows the same diagram as in Figure 4.2, but with examples of graphs inside each intersection, demonstrating that none of these subsets is empty.

4.3.7 Conclusion

We have addressed the equivalence of the graph matching problem with its most common convex relaxation, generalizing the results in [1], and extending the analysis to graph automorphism properties.

Theorem 4.3 and the stronger version, Theorem 4.4, state conditions on the spectral properties of the adjacency matrix of a graph in order for the graph matching problem and the convex relaxation to be equivalent. Specifically, if the adjacency matrix has simple spectrum, and the eigenvectors orthogonal to vector $\mathbf{1}$ have enough non-zero entries, then the equivalence between the two problems holds. This gives also a set of easily verifiable conditions implying that the automorphism group of a graph is trivial.

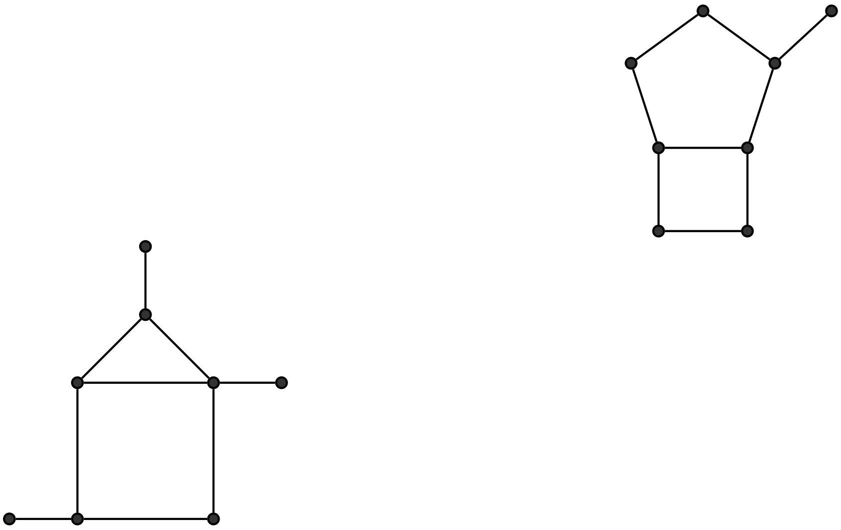


Figure 4.6: Asymmetric graphs with repeated eigenvalues, with (left) and without (right) eigenvectors orthogonal to $\mathbf{1}$.

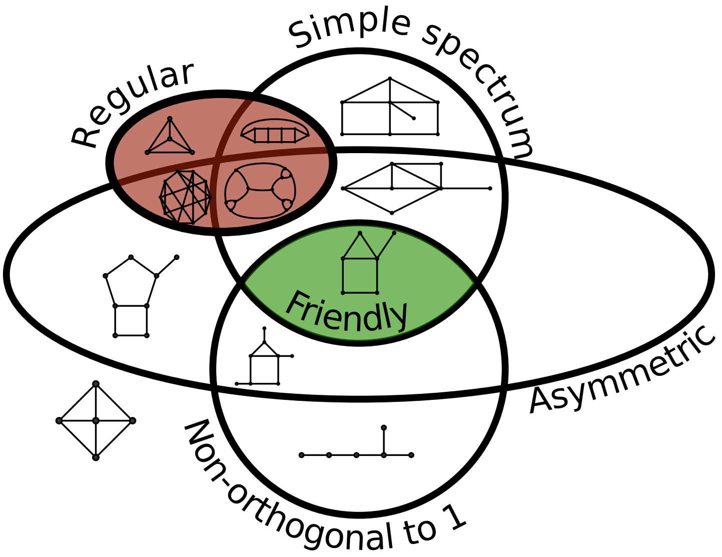


Figure 4.7: Examples of graphs in each class.

The extension of the set where problems (P_1) and (P_2) are known to be equivalent, due to these new results, is shown in Figure 4.8.

In addition to the main theorems, we provided evidence that these particular eigenvectors, orthogonal to $\mathbf{1}$, contain critical information about the symmetries of the graph, specially in their non-zero entries.

During the last decades, important theory was developed on eigenvalues and eigenvectors of the Laplacian matrix of a graph, with very important theoretic results, which brought important applications. The new results here presented shed light on some spectral properties of the adjacency matrix, and leave open some

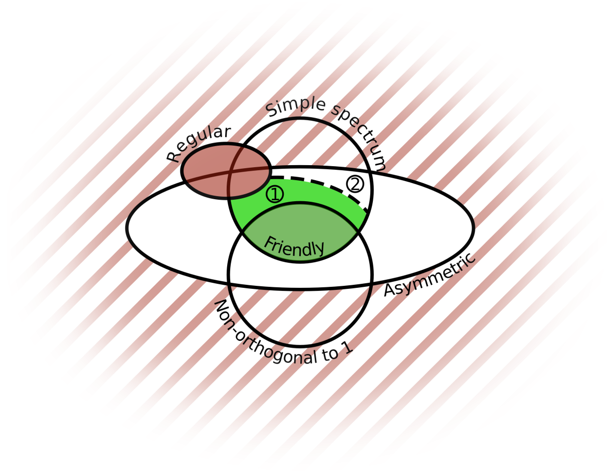


Figure 4.8: Regions where problems (P_1) and (P_2) are known to be equivalent (green) and non-equivalent (red). Outside the *asymmetric* set and inside the *regular* set, the problems are known to be non-equivalent. Problems are equivalent for *friendly* graphs [1], and in the green zone 1 by virtue of the theorems proved in Section 4.3.2. The zone 2 consists of non-regular asymmetric graphs with simple spectrum, but not satisfying the conditions of theorems 4.3 or 4.4; although this subset might be nonempty, since we could not find examples of graphs in this zone.

other questions about the link between these properties and the automorphisms of the graph.

This page was intentionally left blank.

Conclusions

In this thesis we addressed several problems related to graphs, including new algorithms, applications, and theory.

The set of problems tackled in this manuscript can be grouped into two general classes: graph inference problems and graph matching problems.

In Chapter 2 we proposed several algorithms and applications of network estimation. First, we proposed three extensions to ℓ_1 penalized models for graph inference. The first one incorporates topological information to the optimization, allowing to control the graph centrality. We showed how this model can be used to improve the performance of the basic estimation method even when there is no such external information. The second extension favors the appearance of triangles, allowing to better detect motifs in genetic regulatory networks. We combined both models for a better estimation of the *Escherichia coli* genetic regulatory network. Finally, we presented a collaborative model capable of jointly estimating several networks, supposed to share a common structure, and we applied this model to fMRI brain connectivity estimation.

There are several other graph-topological properties that may provide important information, making it interesting to study which kind of structure can be used to guide the optimization problem. On the other hand, the collaborative inference of graphs can be also complemented with other extensions. In particular, an interesting next step would be to collaboratively infer brain connectivity networks of two groups of people, but trying to maximize and spot the main differences between them. This could be used for instance to detect unusual brain activity for some groups, like drug addicts or people suffering from some neuro-biological illness

In the last section of Chapter 2 we presented a mobility graph estimation problem, when only counting information on some nodes is available, the movements are asynchronous, and the time it takes to an entity to go from one site to another depends on the origin and destination, and is unknown. We introduced a formulation based on the dynamics of the system, and we derived its corresponding optimization problem, which was tested for two publicly available real datatets: the New York taxis dataset, and the domestic US flights. The results show that the general topology of the mobility pattern can be recovered, and therefore the system can be analyzed from this inferred network.

Since as far as we know this is the first formulation of this problem, there is a lot of room for improvement, which is part of the future work, as well as different extensions and applications, like analyzing a mobility network at different times of the day to detect patterns of behavior, or detecting outliers in the behavior.

In Chapter 3 we presented methods for graph matching problems. First, we introduced a new formulation for the graph matching problem, inspired by ideas from the sparse modeling community. Since in the problem formulation the weights of the graphs are not compared explicitly, the method can deal with multimodal data, outperforming the other state-of-the-art methods that are used here for comparison. In addition, the proposed formulation naturally fits into the prealignment-free collaborative network inference framework, where the permutation is estimated together with the underlying common network, with promising preliminary results in applications with real data.

In the last section of the chapter, we presented an exhaustive experimental analysis of several graph matching techniques, including a new method inspired by theoretical results that we derived in Chapter 4. The experimental results further emphasize the trade-off between tractability and correctness in relaxing the graph matching problem, with real data experiments and simulations in non edge-independent random graph models suggesting that the theory could be extended to more general random graph settings.

The last chapter is focused on theoretical analyses of the graph matching problem. In the first part, we presented theoretical results showing the surprising fact that the indefinite relaxation (if solved exactly) yields the optimal solution to the graph matching problem with high probability, under mild conditions. Conversely, we also present a novel result which states that the popular convex relaxation of graph matching almost always fails to find the correct (and optimal) permutation. In spite of the apparently negative statements presented here, these results have an immediate practical implication: the usefulness of intelligently initializing the indefinite matching algorithm to obtain a good approximate solution of the indefinite problem, as demonstrated in Chapter 3. The extension of these theoretical results to more general random graph settings will be the subject of the future work.

Finally, in Section 4.3 we studied the equivalence of the graph matching problem with its most common convex relaxation from a deterministic point of view. The main theorem of this section establishes conditions on the spectral properties of the adjacency matrix of a graph under which the graph matching problem and its convex relaxation are equivalent. This gives also a set of easily verifiable conditions implying that the automorphism group of a graph is trivial.

In addition to the consequences for graph matching problems, we provided evidence that the eigenvectors of the adjacency matrix which are orthogonal to $\mathbf{1}$, contain significant information about the symmetries of the graph, specially in their non-zero entries.

Several questions and conjectures were raised along this section, mainly about the link between these spectral properties and the automorphisms of the graph, or the graph matching problem itself. We expect to be able to further understand these connections in the future.

Bibliography

[1] Y. Aflalo, A. Bronstein, and R. Kimmel, "On convex relaxation of graph isomorphism," *Proceedings of the National Academy of Sciences*, pp. 2942–2947, 2015.

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," in *Advances in Neural Information Processing Systems*, 2009, pp. 33–40.

[3] R. Albert, H. Jeong, and A.-L. Barabási, "Internet: Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, 1999.

[4] H. Almohamad and S. Duffuaa, "A linear programming approach for the weighted graph matching problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 5, pp. 522–525, 1993.

[5] N. Alon, J. Kim, and J. Spencer, "Nearly perfect matchings in regular simple hypergraphs," *Israel Journal of Mathematics*, vol. 100, pp. 171–187, 1997.

[6] A. Atserias and E. Maneva, "Sherali–Adams relaxations and indistinguishability in counting logics," *SIAM Journal on Computing*, vol. 42, no. 1, pp. 112–137, 2013.

[7] B. Baingana, G. Mateos, and G. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 8, no. 4, pp. 563–575, Aug 2014.

[8] K. Balińska and L. Quintas, "Algorithms for the random f-graph process," *Communications in Mathematical and in Computer Chemistry/MATCH*, no. 44, pp. 319–333, 2001.

[9] O. Banerjee, L. El Ghaoui, and A. D'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, pp. 485–516, 2008.

[10] O. Banerjee, L. El Ghaoui, A. D'Aspremont, and G. Natsoulis, "Convex optimization techniques for fitting sparse Gaussian graphical models," *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pp. 89–96, 2006.

[11] A. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[12] M. Bazaraa, S. Mokhtar, H. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms.* John Wiley & Sons, 2013.

[13] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[14] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *2005 IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 26–33.

[15] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.* Prentice Hall, 1989.

[16] N. Biggs, *Algebraic Graph Theory*, ser. Cambridge Mathematical Library. Cambridge University Press, 1993.

[17] B. Bollobás, *Random Graphs.* Springer, 1998.

[18] R. E. Burkard, S. E. Karisch, and F. Rendl, "Qaplib–a quadratic assignment problem library," *Journal of Global Optimization*, vol. 10, no. 4, pp. 391–403, 1997.

[19] T. Caelli and S. Kosinov, "An eigenspace projection clustering method for inexact graph matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 4, pp. 515–519, 2004.

[20] E. J. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.

[21] K. Carpenter, P. Sprechmann, M. Fiori, R. Calderbank, H. Egger, and G. Sapiro, "Questionnaire simplification for fast risk analysis of children's mental health," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 6009–6013.

[22] J. Chiquet, Y. Grandvalet, and C. Ambroise, "Inferring multiple graphical structures," *Statistics and Computing*, vol. 21, no. 4, pp. 537–553, 2011.

[23] M. Cho and K. M. Lee, "Progressive graph matching: Making a move of graphs via probabilistic voting," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 398–405.

[24] F. Chung and L. Lu, "Concentration inequalities and Martingale inequalities: A survey," *Internet Mathematics*, vol. 3, pp. 79–127, 2006.

[25] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 03, pp. 265–298, 2004.

[26] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, “A (sub) graph isomorphism algorithm for matching large graphs,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 10, pp. 1367–1372, 2004.

[27] T. Cour, P. Srinivasan, and J. Shi, “Balanced graph matching,” in *Advances in Neural Information Processing Systems*, vol. 19. MIT; 1998, 2007, pp. 313–320.

[28] R. Craddock, G. James, P. Holtzheimer, X. Hu, and H. Mayberg, “A whole brain fMRI atlas generated via spatially constrained spectral clustering,” *Human Brain Mapping*, vol. 33, no. 8, pp. 1914–1928, 2012.

[29] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, “Iteratively reweighted least squares minimization for sparse recovery,” *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.

[30] A. Dempster, “Covariance selection,” *Biometrics*, vol. 28, no. 1, pp. 157–175, 1972.

[31] M. Di Martino, G. Hernández, M. Fiori, and A. Fernández, “A new framework for optimal classifier design,” *Pattern Recognition*, vol. 46, no. 8, pp. 2249–2255, 2013.

[32] D. L. Donoho, “Compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[33] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[34] D. L. Donoho and J. Tanner, “Sparse nonnegative solution of underdetermined linear equations by linear programming,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 27, pp. 9446–9451, 2005.

[35] P. Erdős and A. Rényi, “On random graphs, I,” *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.

[36] M. Faloutsos, P. Faloutsos, and C. Faloutsos, “On power-law relationships of the internet topology,” in *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4. ACM, 1999, pp. 251–262.

[37] S. Fankhauser, K. Riesen, H. Bunke, and P. Dickinson, “Suboptimal graph isomorphism using bipartite matching,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 06, 2012.

[38] M. Fiori, P. Musé, A. Hariri, and G. Sapiro, “Multimodal graphical models via group lasso,” *Signal Processing with Adaptive Sparse Structured Representations*, 2013.

[39] M. Fiori, P. Sprechmann, J. Vogelstein, P. Musé, and G. Sapiro, "Robust multimodal graph matching: Sparse coding meets graph matching," *Advances in Neural Information Processing Systems 26*, pp. 127–135, 2013.

[40] M. Fiori, P. Musé, and G. Sapiro, "Topology constraints in graphical models," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 800–808.

[41] M. Fiori, P. Musé, and G. Sapiro, "Polyps flagging in virtual colonoscopy," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications.* Springer, 2013, pp. 181–189.

[42] M. Fiori, P. Musé, and G. Sapiro, "A complete system for candidate polyps detection in virtual colonoscopy," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 07, 2014.

[43] M. Fiori and G. Sapiro, "On spectral properties for graph matching and graph isomorphism problems," *Information and Inference*, vol. 4, no. 1, pp. 63–76, 2015. doi: 10.1093/imaiai/iav002. [Online]. Available: http://imaiai.oxfordjournals.org/content/4/1/63.abstract

[44] D. E. Fishkind, S. Adali, and C. E. Priebe, "Seeded graph matching," *arXiv:1209.0367*, 2012.

[45] J. P. Formby, W. J. Smith, and B. Zheng, "Mobility measurement, transition matrices and statistical inference," *Journal of Econometrics*, vol. 120, no. 1, pp. 181–205, 2004.

[46] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[47] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso." *Biostatistics*, vol. 9, no. 3, pp. 432–41, Jul. 2008.

[48] J. Friedman, T. Hastie, and R. Tibshirani, "Applications of the lasso and grouped lasso to the estimation of sparse graphical models," Tech. Rep., 2010.

[49] R. Frucht, "Graphs of degree three with a given abstract group," *Canadian J. Math*, vol. 1, pp. 365–378, 1949.

[50] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness.* W.H. Freeman, 1979.

[51] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[52] D. Goldfarb and S. Liu, "An o(n^3l) primal interior point algorithm for convex quadratic programming," *Mathematical Programming*, vol. 49, no. 1-3, pp. 325–340, 1990.

[53] W. R. Gray, J. A. Bogovic, J. T. Vogelstein, B. A. Landman, J. L. Prince, and R. J. Vogelstein, "Magnetic resonance connectome automated pipeline: an overview," *Pulse, IEEE*, vol. 3, no. 2, pp. 42–48, 2012.

[54] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.

[55] J. E. Hopcroft and J.-K. Wong, "Linear time algorithm for isomorphism of planar graphs (preliminary report)," in *Proceedings of the sixth annual ACM symposium on Theory of computing.* ACM, 1974, pp. 172–184.

[56] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack, "Big & quic: Sparse inverse covariance estimation for a million variables," in *Advances in Neural Information Processing Systems 26*, 2013, pp. 3165–3173.

[57] B. Huet, A. D. J. Cross, and E. R. Hancock, "Graph matching for shape retrieval," in *Advances in Neural Information Processing Systems*, 1999, pp. 896–902.

[58] M. Kalisch and P. Bühlmann, "Estimating high-dimensional directed acyclic graphs with the PC-Algorithm," *Journal of Machine Learning Research*, vol. 8, pp. 613–636, 2007.

[59] J. H. Kim, B. Sudakov, and V. H. Vu, "On the asymmetry of random regular graphs and random graphs," *Random Structures and Algorithms*, vol. 21, pp. 216–224, 2002.

[60] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[61] V. Koponen, "Random graphs with bounded maximum degree: asymptotic structure and a logical limit law," *arXiv preprint arXiv:1204.2446*, 2012.

[62] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistic Quarterly*, vol. 2, pp. 83–97, 1955.

[63] S. Lauritzen, *Graphical Models.* Clarendon Press, Oxford, 1996.

[64] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in Neural Information Processing Systems 24*, 2011, pp. 612–620.

[65] Q. Liu and A. Ihler, "Learning scale free networks by reweighted ℓ_1 regularization," *AI & Statistics*, vol. 15, pp. 40–48, Apr. 2011.

[66] P. Loh and M. Wainwright, "Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 2096–2104.

[67] L. Lovász, "Eigenvalues of graphs," *Lecture notes*, 2007, http://www.cs.elte.hu/~lovasz/eigenvals-x.pdf.

[68] E. M. Luks, "Isomorphism of graphs of bounded valence can be tested in polynomial time," *Journal of Computer and System Sciences*, vol. 25, no. 1, pp. 42–65, 1982.

[69] V. Lyzinski, D. Fishkind, M. Fiori, J. Vogelstein, C. Priebe, and G. Sapiro, "Graph matching: Relax at your own risk," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015. doi: 10.1109/TPAMI.2015.2424894

[70] V. Lyzinski, D. L. Sussman, D. E. Fishkind, H. Pao, L. Chen, J. T. Vogelstein, Y. Park, and C. E. Priebe, "Spectral clustering for divide-and-conquer graph matching," *stat*, vol. 1050, p. 22, 2014.

[71] V. Lyzinski, D. E. Fishkind, and C. E. Priebe, "Seeded graph matching for correlated Erdos-Renyi graphs," *Journal of Machine Learning Research*, vol. 15, pp. 3513–3540, 2014. [Online]. Available: http://jmlr.org/papers/v15/lyzinski14a.html

[72] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[73] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the Lasso," *The Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, Jun. 2006.

[74] M. Newman, *Networks: An Introduction.* New York, NY, USA: Oxford University Press, Inc., 2010.

[75] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[76] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.

[77] K. B. Nooner, S. J. Colcombe, R. H. Tobe, M. Mennes, M. M. Benedict, A. L. Moreno, L. J. Panek, S. Brown, S. T. Zavitz, Q. Li *et al.*, "The NKI-Rockland sample: A model for accelerating the pace of discovery science in psychiatry," *Frontiers in Neuroscience*, vol. 6, no. 152, 2012.

[78] K. Nowicki and T. A. B. Snijders, "Estimation and prediction for stochastic blockstructures," *Journal of the American Statistical Association*, vol. 96, no. 455, pp. 1077–1087, 2001.

[79] R. O'Donnell, J. Wright, C. Wu, and Y. Zhou, "Hardness of robust graph isomorphism, Lasserre gaps, and asymmetry of random graphs," *arXiv:1401.2436*, 2014.

[80] J. Peng, P. Wang, N. Zhou, and J. Zhu, "Partial correlation estimation by joint sparse regression models." *Journal of the American Statistical Association*, vol. 104, no. 486, pp. 735–746, Jun. 2009.

[81] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.* IEEE, 2010, pp. 3501–3508.

[82] B. Rolfs, B. Rajaratnam, D. Guillot, I. Wong, and A. Maleki, "Iterative thresholding algorithm for sparse inverse covariance estimation," in *NIPS 2012*, 2012, pp. 1583–1591.

[83] C. Seshadhri, T. Kolda, and A. Pinar, "Community structure and scale-free collections of Erdős-Rényi graphs," *Physical Review E*, vol. 85, no. 5, p. 056109, 2012.

[84] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of Escherichia coli," *Nature Genetics*, vol. 31, no. 1, pp. 64–8, May 2002.

[85] T. A. Snijders and K. Nowicki, "Estimation and prediction for stochastic blockmodels for graphs with latent block structure," *Journal of classification*, vol. 14, no. 1, pp. 75–100, 1997.

[86] P. Sprechmann, I. Ramírez, G. Sapiro, and Y. C. Eldar, "C-hilasso: A collaborative hierarchical sparse modeling framework," *Signal Processing, IEEE Transactions on*, vol. 59, no. 9, pp. 4183–4198, 2011.

[87] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B*, vol. 58, pp. 267–288, 1996.

[88] A. Torsello, D. Hidovic-Rowe, and M. Pelillo, "Polynomial-time metrics for attributed trees," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 7, pp. 1087–1099, 2005.

[89] J. Tropp, "Greed is good: algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, Oct 2004.

[90] J. D. Ullman, A. V. Aho, and J. E. Hopcroft, "The design and analysis of computer algorithms," *Addison-Wesley, Reading*, vol. 4, pp. 1–2, 1974.

[91] J. R. Ullmann, "An algorithm for subgraph isomorphism," *Journal of the ACM (JACM)*, vol. 23, no. 1, pp. 31–42, 1976.

[92] S. Umeyama, “An eigendecomposition approach to weighted graph matching problems,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 5, pp. 695–703, 1988.

[93] G. Varoquaux, A. Gramfort, J. Poline, and B. T., “Brain covariance selection: better individual functional connectivity models using population prior,” in *Advances in Neural Information Processing Systems 23*, 2010, pp. 2334–2342.

[94] L. Varshney, B. Chen, E. Paniagua, D. Hall, and D. Chklovskii, “Structural properties of the caenorhabditis elegans neuronal network,” *PLoS Computational Biology*, vol. 7, no. 2, p. e1001066, 2011.

[95] J. Vogelstein, J. Conroy, V. Lyzinski, L. Podrazik, S. Kratzer, E. Harley, D. Fishkind, R. Vogelstein, and C. Priebe, “Fast approximate quadratic programming for graph matching,” *arXiv:1112.5507*, 2012.

[96] L. Wolf and A. Shashua, “Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach,” *Journal of Machine Learning Research*, vol. 6, pp. 1855–1887, 2005.

[97] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, Feb 2009.

[98] S. Wright, R. Nowak, and M. Figueiredo, “Sparse reconstruction by separable approximation,” *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[99] B. Xiao, E. R. Hancock, and R. C. Wilson, “A generative model for graph matching and embedding,” *Computer Vision and Image Understanding*, vol. 113, no. 7, pp. 777–789, 2009.

[100] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B*, vol. 68, no. 1, pp. 49–67, 2006.

[101] M. Yuan and Y. Lin, “Model selection and estimation in the Gaussian graphical model,” *Biometrika*, vol. 94, no. 1, pp. 19–35, Feb. 2007.

[102] M. Zaslavskiy, F. Bach, and J. Vert, “A path following algorithm for the graph matching problem,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 12, pp. 2227–2242, 2009.

[103] O. Zeitouni, Personal communication, 2012.

[104] F. Zhou and F. De la Torre, “Factorized graph matching,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 127–134.