Convex and Spectral Relaxations for Phase Retrieval, Seriation and Ranking

Fajwel Fogel

ÉCOLE POLYTECHNIQUE

DOCTORAL THESIS

Convex and Spectral Relaxations for Phase Retrieval, Seriation and Ranking

---

FAJWEL FOGEL

Prepared at SIERRA team (D.I. ENS, Inria - CNRS UMR 8548)
& C.M.A.P. (École Polytechnique, CNRS UMR 7641)

A dissertation submitted in fulfillement of the requirements
for the degree of doctor of science, specialized in applied mathematics.

Defended publicly the 18th of November 2015 in front of a jury composed of

Advisors	Alexandre d'Aspremont	CNRS/ENS, Paris, France
	Francis Bach	Inria/ENS, Paris, France
Reviewers	Stéphan Clémençon	Telecom ParisTech, Paris, France
	Anatoli Louditski	Université Joseph Fourier, Grenoble, France
Examiners	Erwan Le Pennec	École Polytechnique, Palaiseau, France
	Milan Vojnovic	Microsoft Research, Cambridge, UK.

Abstract

Optimization is often the computational bottleneck in disciplines such as statistics, biology, physics, finance or economics. Many optimization problems can be directly cast in the well-studied convex optimization framework. For non-convex problems, it is often possible to derive convex or spectral *relaxations*, i.e., derive approximations schemes using spectral or convex optimization tools. Convex and spectral relaxations usually provide guarantees on the quality of the retrieved solutions, which often transcribes in better performance and robustness in practical applications, compared to naive greedy schemes. In this thesis, we focus on the problems of phase retrieval, seriation and ranking from pairwise comparisons. For each of these combinatorial problems we formulate convex and spectral relaxations that are *robust*, *flexible* and *scalable*.

- *Phase retrieval* seeks to reconstruct a complex signal, given a number of observations on the magnitude of linear measurements. In Chapter 2, we focus on problems arising in diffraction imaging, where various illuminations of a single object, e.g., a molecule, are performed through randomly coded masks. We show that exploiting structural assumptions on the signal and the observations, such as sparsity, smoothness or positivity, can significantly speed-up convergence and improve recovery performance.

- The *seriation* problem seeks to reconstruct a linear ordering of items based on unsorted, possibly noisy, pairwise similarity information. The underlying assumption is that items can be ordered along a chain, where the similarity between items decreases with their distance within this chain. In Chapter 3, we first show that seriation can be formulated as a combinatorial minimization problem over the set of permutations, and then derive several convex relaxations that improve the robustness of seriation solutions in noisy settings compared to the spectral relaxation of Atkins et al. (1998). As an additional benefit, these convex relaxations allow to impose a priori constraints on the solution, hence solve semi-supervised seriation problems. We establish new approximation bounds for some of these relaxations and present promising numerical experiments on archeological data, Markov chains and DNA assembly from shotgun gene sequencing data.

- Given pairwise comparisons between n items, the *ranking* problem seeks to find the most consistent global order of these items, e.g., ranking players in a tournament. In practice, the information about pairwise comparisons is usually *incomplete*, especially when the set of items is very large, and the data may also be *noisy*, that is some pairwise comparisons could be incorrectly measured and inconsistent with a total order. In Chapter 4, we formulate this ranking problem as a seriation problem, by constructing an adequate similarity matrix from pairwise comparisons. Intuitively, ordering items based on this similarity

assigns similar rankings to items that compare similarly with all others. We then study how the spectral relaxation of seriation from Atkins et al. (1998) performs on ranking. We first show that this spectral seriation algorithm recovers the true ranking when all pairwise comparisons are observed and consistent with a total order. We then show that ranking reconstruction is still exact when some pairwise comparisons are corrupted or missing, and that seriation based spectral ranking is more robust to noise than classical scoring methods. Finally, we bound the ranking error when only a random subset of the comparisons are observed. This theoretical analysis is supported by experiments on both synthetic and real datasets that lead to competitive and in some cases superior performance compared to classical ranking methods.

Résumé

L'optimisation s'avère souvent essentielle dans de nombreuses disciplines: statistiques, biologie, physique, finance ou encore économie. De nombreux problèmes d'optimisation peuvent être directement formulés dans le cadre de l'optimisation convexe, un domaine très bien étudié. Pour les problèmes non convexes, il est souvent possible d'écrire des *relaxations* convexes ou spectrales, i.e., d'établir des schémas d'approximations utilisant des techniques convexes ou spectrales. Les relaxations convexes et spectrales fournissent en général des garanties sur la qualité des solutions associées. Cela se traduit souvent par de meilleures performances et une plus grande robustesse dans les applications, par rapport à des méthodes gloutonnes naïves. Dans ce manuscript de thèse, nous nous intéressons aux problèmes de reconstruction de phase, de sériation, et de classement à partir de comparaisons par paires. Nous formulons pour chacun de ces problèmes des relaxations convexes ou spectrales à la fois *robustes*, *flexibles*, et *adaptées à de grands jeux de données*.

- Le problème de *reconstruction de phase* consiste à reconstruire un signal complexe, étant donnée l'amplitude de mesures linéaires. Dans le chapitre 2, nous nous intéressons plus particulièrement au problème de diffraction d'image, pour lequel plusieurs illuminations d'un objet unique (par exemple une molécule) sont obtenues à travers des masques codés de manière aléatoire. Nous montrons qu'en exploitant des hypothèses structurelles sur le signal et les observations, comme la parcimonie, la continuité ou la positivité, la vitesse de convergence et la précision de la phase récupérée peuvent être améliorées de manière significative.

- Le problème de *sériation* a pour but de retrouver un ordre linéaire d'éléments, à partir d'une information potentiellement bruitée sur les similarités entre paires d'éléments. L'hypothèse sous-jacente de ce problème est que les éléments peuvent être ordonnés selon une chaîne, de telle manière que la similarité entre deux éléments diminue avec leur distance dans cette chaîne. Dans le chapitre 3, nous montrons tout d'abord que le problème de sériation peut être formulé comme un problème d'optimisation combinatoire sur l'ensemble des permutations, puis nous dérivons plusieurs relaxations convexes qui améliorent la robustesse des solutions au problème de sériation dans des régimes bruités, par rapport à la relaxation spectrale d'Atkins et al. (1998). De plus, ces relaxations convexes permettent d'imposer un a priori sur la solution, et donc de résoudre des problèmes de sériation semi-supervisés. Nous établissons de nouvelles bornes d'approximation pour certaines de ces relaxations et présentons des résultats prometteurs sur des données archéologiques, des chaînes de Markov, et des problèmes d'assemblage d'ADN à partir de données de séquençage génétique.

- Étant données des comparaisons par paires pour un ensemble d'éléments, le problème de classement (*ranking* en anglais) consiste à trouver l'ordre global de ces éléments le plus en accord avec ces comparaisons. Il peut s'agir par exemple d'établir le classement des participants d'un tournoi à partir des résultats de matchs opposant des paires de joueurs. En pratique, ces comparaisons ne sont en général pas disponibles pour toutes les paires d'éléments, en particulier pour les grands jeux de données. Par ailleurs, les données peuvent être bruitées, certaines comparaisons ayant pu être mesurées de manière incorrecte et non consistante avec un ordre global. Dans le chapitre 4, nous formulons le problème de classement comme un problème de sériation, en construisant une matrice de similarité à partir des comparaisons. Nous étudions ensuite les performances de la relaxation spectrale du problème de sériation appliquée à ce problème. Nous montrons tout d'abord que cet algorithme spectral de sériation retrouve le bon classement lorsque les comparaisons sont observées pour toutes les paires d'éléments et sont en accord avec un ordre global sous-jacent. Puis nous montrons que le classement retrouvé est toujours exact lorsqu'un petit nombre de comparaisons sont manquantes ou corrompues. Cela rend cette méthode plus robuste au bruit que les méthodes classiques de "scoring". Enfin, nous bornons l'erreur sur le classement lorsque l'on observe seulement un petit nombre de comparaisons. Cette analyse théorique est confirmée par des expériences numériques sur des jeux de données synthétiques et réels, avec des performances aussi bonnes ou supérieures aux méthodes de classement classiques, selon les contextes.

Acknowledgements

First and foremost, I would like to deeply thank my two Ph.D. advisors, Alexandre d'Aspremont and Francis Bach. Alexandre has been very attentive to my progress, and helped me on many topics, from mathematics to writing, while giving me a lot of liberty in my work. Francis was always available for meetings though his tight schedule, and provided very constructive advice. It has been an invaluable privilege to work with them.

I would like to thank Stéphan Clemençon and Anatoli Juditsky for accepting to review my thesis. I am also grateful to Milan Vojnovic and Erwan Le Pennec for accepting to be part of the jury. Special thanks go to Milan Vojnovic who was my supervisor at Microsoft Research Cambridge for two months in Spring 2014, it was a great pleasure working with him.

I would also like to thank all my professors at ENSAE and Columbia University who put me on the track of Ph.D. studies: Frédéric Pascal, Marc Hoffmann, Alexandre Tsybakov, Donald Goldfarb among others. I am particularly grateful to Edmond Lezmi who was my supervisor at Amundi during a long internship back in 2011, and strongly encouraged me to apply for a Ph.D.

My thanks also go to all the people from the Sierra and Willow team at Inria, who were always very cheerful and friendly, which gave a very nice atmosphere to the lab.

Finally, I want to deeply thank my parents, my sister, my friends, and of course Alice for their inalienable support during these three years.

Financial support. I would like to acknowledge support from the European Research Council grant allocated to my advisor Alexandre d'Aspremont (ERC project SIPA) and from the MSR-Inria Joint Centre.

Contents

Chapter 1

Introduction

Optimization has witnessed a core revolution in the last two decades. The work of Nesterov and Nemirovskii (1994) produced a radical shift in our understanding of problem complexity. It showed that the old dichotomy between linear and nonlinear problems was in fact somewhat irrelevant to computational complexity, and identified convexity (and to a lesser extent smoothness) as the key indicator of computational tractability. Historically, linear programs for example were first solved using some variant of the *simplex* algorithm (Dantzig, 1963). While these algorithms were very efficient in practice, their worst-case complexity was exponential in the dimension n. The ellipsoid method by Nemirovskii and Yudin (1979) was used by Khachiyan (1979) to show that convex programs can be solved in polynomial time, but it was not until the work of Karmarkar (1984) that an efficient polynomial time algorithm for solving linear programs was derived using *interior point methods*. The results of Nesterov and Nemirovskii (1994) showed that interior point algorithms designed for linear programming could be extended to solve very general classes of convex problems, and in particular, quadratic and semidefinite programs.[1]

Reliable numerical packages based on interior point methods now solve medium scale problems (with dimension n up to a few thousands) very efficiently. On large-scale convex optimization problems however, memory constraints quickly arise to the point where forming even a single iteration of interior point algorithms becomes impossible. In order to lift these restrictions on problem size while preserving the striking reliability of interior point solvers, customized optimization methods for the problems at hand can have a dramatic impact, as will be shown in Chapter 2.

Many engineering and statistical problems can be directly cast in the convex optimization framework. For non-convex problems, it is often possible to derive convex or spectral *relaxations*, i.e.,

[1] See for instance (Boyd and Vandenberghe, 2004, chap. 4) for an introduction to convex optimization problems.

derive approximations schemes using spectral or convex optimization tools. Convex and spectral relaxations sometimes provide guarantees on the quality of the retrieved solutions, as for the famous MAXCUT semi-definite programming relaxation (Goemans and Williamson, 1995). These guarantees often imply better performance and robustness in practical applications, compared to naive greedy schemes.

Convex relaxations based on semi-definite programming (SDP) or quadratic programming (QP) tend be more robust than spectral relaxations, and more flexible, e.g., enabling to add structural and a priori constraints, as will be seen in Chapters 2 and 3. On the other hand, spectral relaxations are usually much more scalable to large datasets, as for instance spectral clustering algorithms (see Von Luxburg, 2007, for an introduction).

In this thesis, we demonstrate on real problems how one can formulate and use convex and spectral relaxations that are *robust*, *flexible* and *scalable*. We focus on three important problems of high complexity: phase retrieval, seriation and ranking from pairwise comparisons.

- *Phase retrieval* seeks to reconstruct a complex signal, given a number of observations on the magnitude of linear measurements. We focus on problems arising in diffraction imaging, where various illuminations of a single object, e.g., a molecule, are performed through randomly coded masks.
- The *seriation* problem seeks to reconstruct a linear ordering of items based on unsorted, possibly noisy, pairwise similarity information. The underlying assumption is that items can be ordered along a chain, where the similarity between items decreases with their distance within this chain.
- Given pairwise comparisons between a set of items, the *ranking* problem is to find the most consistent global order of these items, e.g., ranking players in a tournament. In practice, the information about pairwise comparisons is usually *incomplete*, especially when the set of items is large, and the data may also be *noisy*, that is some pairwise comparisons could be incorrectly measured and inconsistent with a total order.

The remaining of this introduction is organized as followed. We first review contributions of this thesis in Section 1.1. We then present the phase retrieval, seriation and ranking problems in Section 1.3. In Section 1.4, we establish the common challenges of these problems that motivate our work. In Section 1.5, we show how these three problems are related to each other. Finally, we recall in Section 1.6 standard convex and spectral relaxation of the MAXCUT and balanced MINCUT, which will be a prerequisite for understanding relaxations presented in following chapters.

1.1 Contributions

- In Chapter 2, we test algorithms to solve convex relaxations of the phase retrieval problem for molecular imaging. We show that exploiting structural assumptions on the signal and the observations, such as sparsity, smoothness or positivity, can significantly speed-up convergence and improve recovery performance. We detail numerical results in molecular imaging experiments simulated using data from the Protein Data Bank (PDB). The material of this part is based on the following publication:
  F. Fogel, I. Waldspurger, A. d'Aspremont, Phase retrieval for imaging problems. To appear in *Mathematical Programming Computation*.

- Chapter 3 presents convex relaxations for the seriation problem, which seeks to reconstruct a linear order between variables using unsorted, pairwise similarity information. We first write seriation as an optimization problem by proving the equivalence between the seriation and combinatorial 2-SUM problems on similarity matrices (2-SUM is a quadratic minimization problem over permutations). The seriation problem can be solved exactly by a spectral algorithm in the noiseless case and we derive several convex relaxations for 2-SUM to improve the robustness of seriation solutions in noisy settings. These convex relaxations also allow us to impose a priori constraints on the solution, hence solve semi-supervised seriation problems. We derive new approximation bounds for some of these relaxations and present numerical experiments on archeological data, Markov chains and DNA assembly from shotgun gene sequencing data.
  The material of this part is based on the following publications:
  F. Fogel, R. Jenatton, F. Bach, A. d'Aspremont, Convex relaxations for permutation problems. In *Advances in Neural Information Processing Systems*, pp. 1016-1024. 2013.
  F. Fogel, R. Jenatton, F. Bach, A. d'Aspremont, Convex relaxations for permutation problems. To appear in *SIAM Journal on Matrix Analysis and Applications (SIMAX)*.

- In Chapter 4, we describe a seriation algorithm for ranking a set of items given pairwise comparisons between these items. Intuitively, the algorithm assigns similar rankings to items that compare similarly with all others. It does so by constructing a similarity matrix from pairwise comparisons, using seriation methods to reorder this matrix and construct a ranking. We first show that this spectral seriation algorithm recovers the true ranking when all pairwise comparisons are observed and consistent with a total order. We then show that ranking reconstruction is still exact when some pairwise comparisons are corrupted or missing, and that seriation based spectral ranking is more robust to noise than classical scoring methods. Finally, we bound the ranking error when only a random subset of the comparisons are observed. An additional benefit of the seriation formulation is that it allows us to solve semi-supervised ranking problems. Experiments on both synthetic and real datasets demonstrate that seriation based spectral ranking achieves competitive and

in some cases superior performance compared to classical ranking methods.

The material of this part is based on the following publications:

F. Fogel, A. d'Aspremont, M. Vojnovic: Serialrank: spectral ranking using seriation. In *Advances in Neural Information Processing Systems*, pp. 900-908. 2014.

F. Fogel, A. d'Aspremont, M. Vojnovic: Spectral ranking using seriation. In submission.

1.2 Notations

Matrices and vectors. We write $\mathbf{S}_n$ (resp. $\mathbf{H}_n$) the cone of symmetric (resp. Hermitian) matrices of dimension n ; $\mathbf{S}_n^+$ (resp. $\mathbf{H}_n^+$) denotes the set of positive symmetric (resp. Hermitian) matrices. We write $A^\dagger$ the (Moore-Penrose) pseudoinverse of a matrix A, and $A \circ B$ the Hadamard (or componentwise) product of the matrices A and B. For $x \in \mathbb{R}^n$ (resp. $x \in \mathbb{C}^n$), $\mathbf{diag}(x)$ is the matrix with diagonal x. When $X \in \mathbf{S}_n$ (resp. $X \in \mathbf{H}_n$) however, $\mathbf{diag}(X)$ is the vector containing the diagonal elements of X. For a matrix X, X^T is the transpose of X, and $\bar{X}$ is the matrix whose elements are the complex conjugate of the elements of X. For $X \in \mathbf{H}_n$, X^* is the Hermitian transpose of X, with $X^* = (\bar{X})^T$. We write b^2 the vector with components b_i^2, $i = 1, \ldots, n$. Here, $e_i \in \mathbb{R}^n$ is i-the Euclidean basis vector and $\mathbf{1}$ is the vector of ones. For a matrix $A \in \mathbb{R}^{m \times n}$, $\mathbf{vec}\, A \in \mathbb{R}^{mn}$ is the vector formed by stacking up the columns of A. We denote by $\mathbf{I}$ the identity matrix.

Norms. We write $\|\cdot\|_p$ the Schatten p-norm of a matrix, that is the p-norm of the vector of its eigenvalues, at the exception of Chapter 3, where $\|\cdot\|_2$ denotes the operator norm. $\|\cdot\|_F$ denotes the Frobenius norm, $\lambda_i(X)$ the i^{th} eigenvalue (in increasing order) of X. We write $|\cdot|$ the absolute value (resp. modulus) of a real (resp. complex) number. When x is a vector in $\mathbb{R}^n$ (resp. $x \in \mathbb{C}^n$), $|x|$ is the vector with coefficients $(|x_1|, \ldots, |x_n|)$.

Permutations. We use the notation $\mathcal{P}$ for both the set of permutations of $\{1, \ldots, n\}$ and the set of permutation matrices. The notation π refers to a permuted vector $(1, \ldots, n)^T$ while the notation Π (in capital letter) refers to the corresponding matrix permutation, which is a $\{0, 1\}$ matrix such that $\Pi_{ij} = 1$ if and only if $\pi(i) = j$. Moreover, for any vector y in $\mathbb{R}^n$, y_π is the vector with coefficients $(y_{\pi(1)}, \ldots, y_{\pi(n)})$ hence $\Pi y = y_\pi$ and $\Pi^T y_\pi = y$. This also means that $A\Pi^T$ is the matrix with coefficients $A_{i\pi(j)}$, and $\Pi A \Pi^T$ is the matrix with coefficients $A_{\pi(i)\pi(j)}$.

If need be, additional and more specific notation may be introduced at the beginning of some chapters.

1.3 Phase retrieval, seriation and ranking problems

We now present the phase retrieval, seriation and ranking problems, illustrating them by practical examples. Related work will also be discussed in following chapters for completeness.

1.3.1 Phase retrieval for molecular imaging problems

Problem statement

Phase retrieval seeks to reconstruct a complex signal, given a number of observations on the *magnitude* of linear measurements, i.e., solve

$$\begin{array}{ll} \text{find} & x \\ \text{such that} & |Ax| = b \end{array} \tag{1.1}$$

in the variable $x \in \mathbb{C}^p$, where $A \in \mathbb{R}^{n\times p}$, $b \in \mathbb{R}^n$ and $|Ax|$ is the vector with coefficients equal to the absolute values of the coefficients of Ax. Phase retrieval has direct applications in imaging problems where physical limitations imply detectors usually capture the intensity of observations but cannot recover their phase, for instance in X-ray and crystallography imaging, diffraction imaging, Fourier optics or microscopy. In what follows, we focus on problems arising in diffraction imaging, where various illuminations of a single object are performed through randomly coded masks, as illustrated in Figure 1.1 from (Candes et al., 2014). Hence the matrix A is usually formed using a combination of random masks and Fourier transforms, and we have significant structural information on both the signal we seek to reconstruct (regularity, etc.) and the observations (power law decay in frequency domain, etc.). Many of these additional structural hints can be used to speedup numerical operations, convergence and improve phase retrieval performance, as will be detailed in Chapter 2.

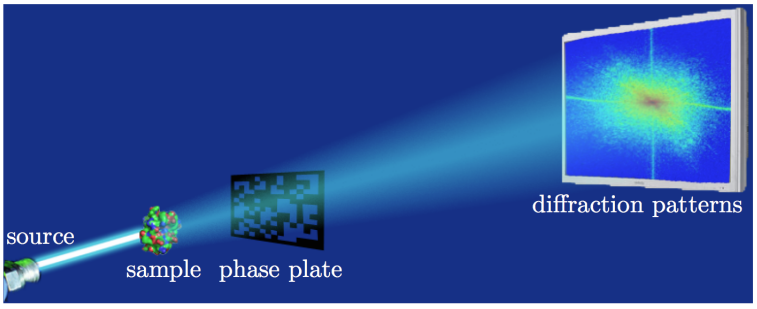


FIGURE 1.1: An illustrative setup for acquiring coded diffraction patterns (Candes et al., 2014).

Related work

Because the phase constraint $|Ax| = b$ is non-convex, the phase recovery problem (1.1) is non-convex. Several greedy algorithms have been developed (see for instance Gerchberg and Saxton, 1972; Fienup, 1982; Griffin and Lim, 1984; Bauschke et al., 2002), which alternate projections on the range of A and on the non-convex set of vectors y such that $|y| = |Ax|$. While empirical performance is often good, these algorithms can stall in local minima. A convex relaxation was introduced in (Chai et al., 2011) and (Candes et al., 2015a), called PhaseLift, by observing that $|Ax|^2$ is a linear function of $X = xx^*$, which is a rank-one Hermitian matrix, using the classical lifting argument for non-convex quadratic programs developed in (Shor, 1987; Lovász and Schrijver, 1991). The recovery of x is thus expressed as a rank minimization problem over positive semidefinite Hermitian matrices X satisfying some linear conditions, i.e., a matrix completion problem. This last problem has received a significant amount of attention because of its link to compressed sensing and the NETFLIX collaborative filtering problem (SIGKDD, 2007). This minimum rank matrix completion problem is approximated by a semidefinite program which has been shown to recover x for several (random) classes of observation operators A (Candes et al., 2013, 2014, 2015a).

On the algorithmic side, Waldspurger et al. (2015) showed that the phase retrieval problem (1.1) can be reformulated in terms of a single phase variable, which can be read as an extension of the MAXCUT combinatorial graph partitioning problem over the unit complex torus, allowing fast algorithms designed for solving semidefinite relaxations of MAXCUT to be applied to the phase retrieval problem. Besides, Candes et al. (2015b) have recently proposed a non-convex algorithm with spectral initialization for phase retrieval, with theoretical guarantees.

On the experimental side, phase recovery is a classical problem in Fourier optics for example (Goodman, 2008), where a diffraction medium takes the place of a lens. This has direct applications in X-ray and crystallography imaging, diffraction imaging or microscopy (Harrison, 1993; Bunk et al., 2007; Johnson et al., 2008; Miao et al., 2008; Dierolf et al., 2010).

Standard approaches

We now present an overview of several basic algorithmic approaches to solve the phase retrieval problem (1.1). Early methods were all based on extensions of an alternating projection algorithm. However, recent results showed that phase retrieval could be interpreted as a matrix completion problem similar to the NETFLIX problem, a formulation which yields both efficient convex relaxations and recovery guarantees.

Greedy algorithms. The phase retrieval problem (1.1) can be rewritten as

$$\begin{array}{ll} \text{minimize} & \|Ax - y\|_2^2 \\ \text{subject to} & |y| = b, \end{array} \tag{1.2}$$

where we now optimize over both phased observations $y \in \mathbb{C}^n$ and signal $x \in \mathbb{C}^p$. Several greedy algorithms attempt to solve this problem using variants of alternating projections, one iteration minimizing the quadratic error (the objective of (1.2)), the next normalizing the moduli (to satisfy the constraint). We detail some of the most classical examples in the paragraphs that follow.

The algorithm Gerchberg-Saxton in (Gerchberg and Saxton, 1972) for instance seeks to reconstruct $y = Ax$ and alternates between orthogonal projections on the range of A and normalization of the magnitudes $|y|$ to match the observations b. The cost per iteration of this method is minimal but convergence (when it happens) is often slow.

Algorithm 1 Gerchberg-Saxton.

Input: An initial $y^1 \in \mathbf{F}$, i.e., such that $|y^1| = b$.

1: **for** $k = 1, \dots, N-1$ **do**

2: Set

$$y_i^{k+1} = b_i \frac{(AA^\dagger y^k)_i}{|(AA^\dagger y^k)_i|}, \quad i = 1, \dots, n. \tag{Gerchberg-Saxton}$$

3: **end for**

Output: $y_N \in \mathbf{F}$.

A classical "input-output" variant, detailed here as algorithm Fienup, introduced by Fienup (1982), adds an extra penalization step which usually speeds up convergence and improves recovery performance when additional information is available on the support of the signal. Oversampling the Fourier transform forming A in imaging problems usually helps performance as well. Of course, in all these cases, convergence to a global optimum cannot be guaranteed but empirical recovery performance is often quite good.

Algorithm 2 Fienup

Input: An initial $y^1 \in \mathbf{F}$, i.e., such that $|y^1| = b$, a parameter $\beta > 0$.

1: **for** $k = 1, \dots, N-1$ **do**

2: Set

$$w_i = \frac{(AA^\dagger y^k)_i}{|(AA^\dagger y^k)_i|}, \quad i = 1, \dots, n.$$

3: Set

$$y_i^{k+1} = y_i^k - \beta(y_i^k - b_i w_i) \tag{Fienup}$$

4: **end for**

Output: $y_N \in \mathbf{F}$.

PhaseLift: semidefinite relaxation in signal. Using a classical lifting argument by (Shor, 1987), and writing

$$|a_i^* x|^2 = b_i^2 \Longleftrightarrow \mathbf{Tr}(a_i a_i^* x x^*) = b_i^2$$

(Chai et al., 2011; Candes et al., 2015a) reformulate the phase recovery problem (2.1) as a matrix completion problem, written

$$\begin{array}{ll} \text{minimize} & \mathbf{Rank}(X) \\ \text{subject to} & \mathbf{Tr}(a_i a_i^* X) = b_i^2, \quad i = 1, \ldots, n \\ & X \succeq 0, \end{array}$$

in the variable $X \in \mathbf{H}_p$, where $X = xx^*$ when exact recovery occurs. This last problem can be relaxed as

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(X) \\ \text{subject to} & \mathbf{Tr}(a_i a_i^* X) = b_i^2, \quad i = 1, \ldots, n \\ & X \succeq 0, \end{array} \qquad \text{(PhaseLift)}$$

which is a semidefinite program (called PhaseLift by Candes et al. (2015a)) in the variable $X \in \mathbf{H}_p$. This problem is solved in (Candes et al., 2015a) using first-order algorithms implemented in (Becker et al., 2011). This semidefinite relaxation has been shown to recover the true signal x exactly for several classes of observation operators A (Candes et al., 2015a, 2013, 2014).

PhaseCut: semidefinite relaxation in phase. As in (Waldspurger et al., 2015) we can rewrite the phase reconstruction problem (1.1) in terms of a phase variable u (such that $|u| = 1$) instead of the signal x. In the noiseless case, we then write the constraint $|Ax| = b$ as $Ax = \mathbf{diag}(b)u$, where $u \in \mathbb{C}^n$ is a phase vector, satisfying $|u_i| = 1$ for $i = 1, \ldots, n$, so problem (1.1) becomes

$$\begin{array}{ll} \text{minimize} & \|Ax - \mathbf{diag}(b)u\|_2^2 \\ \text{subject to} & |u_i| = 1 \end{array} \qquad (1.3)$$

where we optimize over both phase $u \in \mathbb{C}^n$ and signal $x \in \mathbb{C}^p$. While the objective of this last problem is jointly convex in (x, u), the phase constraint $|u_i| = 1$ is not.

Now, given the phase, signal reconstruction is a simple least squares problem, i.e., given u we obtain x as

$$x = A^\dagger \, \mathbf{diag}(b)u \qquad (1.4)$$

where $A^\dagger$ is the pseudo inverse of A. Replacing x by its value in problem (1.3), the phase recovery problem becomes

$$\begin{array}{ll} \text{minimize} & u^* M u \\ \text{subject to} & |u_i| = 1, \quad i = 1, \ldots, n, \end{array} \qquad (1.5)$$

in the variable $u \in \mathbb{C}^n$, where the Hermitian matrix

$$M = \mathbf{diag}(b)(\mathbf{I} - AA^\dagger)\,\mathbf{diag}(b)$$

is positive semidefinite. This problem is non-convex in the phase variable u. Waldspurger et al. (2015) detailed greedy algorithm Greedy to locally optimize (1.5) in the phase variable.

Algorithm 3 Greedy algorithm in phase.

Input: An initial $u \in \mathbb{C}^n$ such that $|u_i| = 1, i = 1, \ldots, n$. An integer $N > 1$.

1: **for** $k = 1, \ldots, N$ **do**
2: **for** $i = 1, \ldots, n$ **do**
3: Set

$$u_i = \frac{-\sum_{j \neq i} M_{ji} \bar{u}_j}{\left|\sum_{j \neq i} M_{ji} \bar{u}_j\right|} \qquad \text{(Greedy)}$$

4: **end for**
5: **end for**

Output: $u \in \mathbb{C}^n$ such that $|u_i| = 1, i = 1, \ldots, n$.

A convex relaxation to (1.5) was also derived in (Waldspurger et al., 2015) using the classical lifting argument for non-convex quadratic programs developed in (Shor, 1987; Lovász and Schrijver, 1991). This relaxation is written

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(UM) \\ \text{subject to} & \mathbf{diag}(U) = 1,\ U \succeq 0, \end{array} \qquad \text{(PhaseCut)}$$

which is a semidefinite program (SDP) in the matrix $U \in \mathbf{H}_n$. This problem has a structure similar to the classical MAXCUT relaxation and instances of reasonable size can be solved using specialized implementations of interior point methods designed for that problem (Helmberg et al., 1996). Larger instances are solved in (Waldspurger et al., 2015) using the block-coordinate descent algorithm BlockPhaseCut.

Ultimately, algorithmic choices heavily depend on problem structure, and these are discussed in detail in Chapter 2. In particular, we study how to exploit structural information on the signal (nonnegativity, sparse 2D Fast Fourier Transform, etc.), to solve realistically large instances formed in diffraction imaging applications (i.e., images with several thousands of pixels).

Algorithm 4 Block Coordinate Descent Algorithm for **PhaseCut**.

Input: An initial $U^0 = \mathbf{I}_n$ and $\nu > 0$ (typically small). An integer $N > 1$.

1: **for** $k = 1, \ldots, N$ **do**

2: Pick $i \in [1, n]$. i^c refers to the set $\{1, \ldots, i-1, i+1, \ldots, n\}$

3: Compute

$$u = U^k_{i^c, i^c} M_{i^c, i} \quad \text{and} \quad \gamma = u^* M_{i^c, i} \qquad \text{(BlockPhaseCut)}$$

4: If $\gamma > 0$, set

$$U^{k+1}_{i^c, i} = U^{k+1*}_{i, i^c} = -\sqrt{\frac{1-\nu}{\gamma}} x$$

else

$$U^{k+1}_{i^c, i} = U^{k+1*}_{i, i^c} = 0.$$

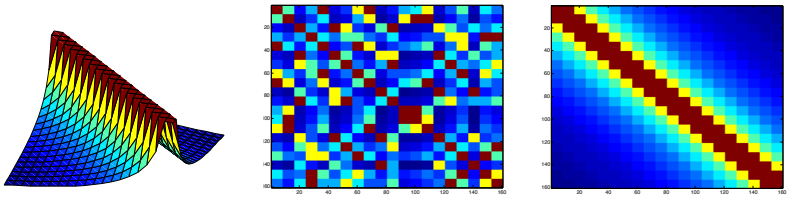
5: **end for**

Output: A matrix $U \succeq 0$ with $\mathbf{diag}(U) = 1$.

1.3.2 Seriation: ordering from a similarity

Problem statement

In the seriation problem, we are given pairwise similarities between n variables and assume that variables can be ordered along a chain, where the similarity between variables decreases with their distance within this chain. The seriation problem seeks to reconstruct this linear ordering based on unsorted, possibly noisy, pairwise similarity information. This amounts to finding a permutation that reorders the rows and columns of the similarity matrix such that coefficients are decreasing when going away from the diagonal (*cf.* Figure 1.2). As an illustrative example, let us assume that we are given the frames of a movie in a random order, as in Figure 1.3 photos of a teapot from different angles, and compute a similarity between frames, based on the ℓ_2-distance bewteen relative pixels. The seriation problem seeks to retrieve the correct order of the frames; in our case the teapot should turn on itself.



Similarity matrix Input Reconstructed

FIGURE 1.2: The seriation problem amounts to finding a permutation that reorders the rows and columns of the similarity matrix such that coefficients are decreasing when going away from the diagonal.

Randomly ordered movie.

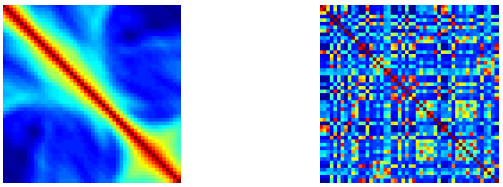


Image similarity matrix (true & observed)

Correctly ordered movie.

FIGURE 1.3: Illustrative example of the seriation problem: retrieve a coherent order of the frames of a movie.

The seriation problem has its roots in archeology (Robinson, 1951), where given undated objects and some of their characteristics, the goal is to find a relative temporal order of objects, based on the assumption that objects from a close period in time should share similar artifacts.

Seriation also has direct applications in DNA de novo assembly, where a single strand of genetic material is reconstructed from many cloned shorter reads (i.e., small, fully sequenced sections of DNA) (Garriga et al., 2011; Meidanis et al., 1998). As illustrated in Figure 1.4 (taken from Commins et al., 2009), genomes are cloned multiple times and randomly cut into shorter reads (a few hundreds base pairs A/C/T/G), which are fully sequenced. We need to reorder the reads to recover the genome, which amounts to solving a seriation problem.

Related work

The seriation problem also has direct applications in e.g., envelope reduction algorithms for sparse linear algebra (Barnard et al., 1995), in identifying interval graphs for scheduling (Fulkerson and Gross, 1965). With DNA assembly applications in mind, many references focused on the *consecutive ones problem* (C1P) which seeks to permute the rows of a binary matrix so that all the ones in each column are contiguous. In particular, Fulkerson and Gross (1965) studied further connections to interval graphs and Kendall (1971) crucially showed that a solution to C1P can be obtained by solving the seriation problem on the squared data matrix. We refer the

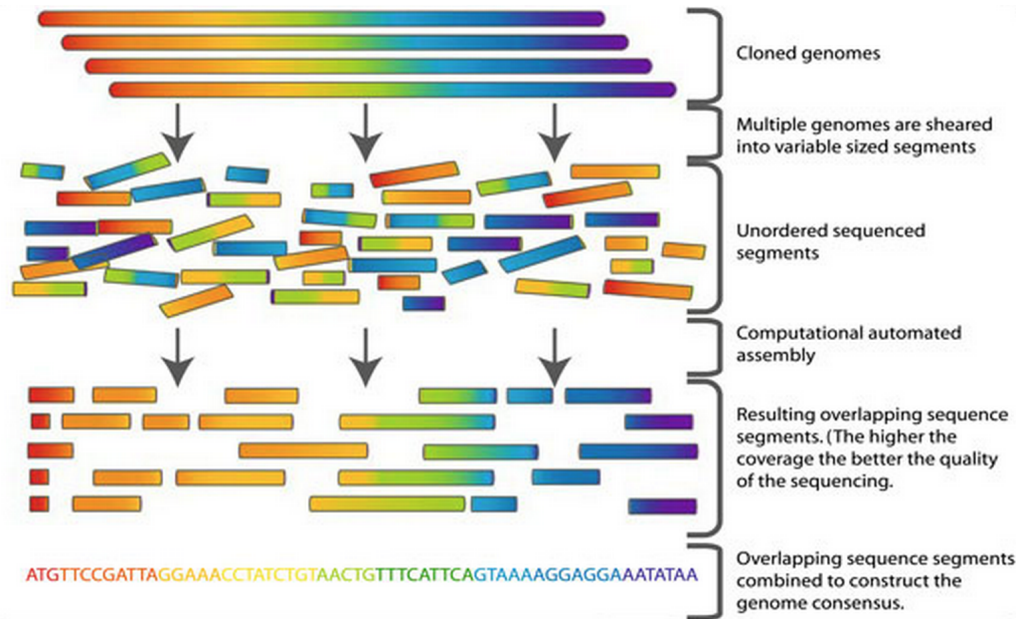


FIGURE 1.4: The seriation problem has direct applications in DNA de novo assembly, where a single strand of genetic material is reconstructed from many cloned shorter reads (Commins et al., 2009).

reader to (Ding and He, 2004; Vuokko, 2010; Liiv, 2010) for a much more complete survey of applications.

On the algorithmic front, the seriation problem was shown to be NP-complete by George and Pothen (1997). Archeological examples are usually small scale and earlier references such as (Robinson, 1951) used greedy techniques to reorder matrices. Similar techniques were, and are still used to reorder genetic data sets. More general ordering problems were studied extensively in operations research, mostly in connection with the quadratic assignment problem (QAP), for which several convex relaxations were derived in e.g., (Lawler, 1963; Zhao et al., 1998). Since a matrix is a permutation matrix if and only if it is both orthogonal and doubly stochastic, much work also focused on producing semidefinite relaxations to orthogonality constraints (Nemirovski, 2007; So, 2011). These programs are convex and can be solved using conic programming solvers, but relaxations have usually very large dimensions and scale poorly. More recently however, Atkins et al. (1998) produced a spectral algorithm that exactly solves the seriation problem in a noiseless setting. They show that for similarity matrices computed from serial variables (for which a total order exists), the ordering of the second eigenvector of the Laplacian (a.k.a. the Fiedler vector) matches that of the variables. A lot of work has focused on the minimum linear arrangement problem or 1-SUM, with (Even et al., 2000; Feige, 2000; Blum et al., 2000) and (Rao and Richa, 2005; Feige and Lee, 2007; Charikar et al., 2010) producing semidefinite relaxations with nearly dimension independent approximation ratios. While these relaxations form semidefinite programs that have an exponential number of constraints, they admit a polynomial-time separation oracle and can be solved using the ellipsoid method. The

later algorithm being extremely slow, these programs have very little practical impact. Finally, seriation is also directly related to the manifold learning problem (Weinberger and Saul, 2006), which seeks to reconstruct a low dimensional manifold based on local metric information. Seriation can be seen as a particular instance of that problem, where the manifold is unidimensional but the similarity information is not metric.

Standard approaches

2-SUM formulation. Given a symmetric, binary matrix A, we focus on variations of the following 2-SUM combinatorial minimization problem, studied in e.g., (George and Pothen, 1997), and written

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^{n} A_{ij}(\pi(i)-\pi(j))^2 \\ \text{subject to} & \pi \in \mathcal{P}, \end{array} \tag{1.6}$$

where $\mathcal{P}$ is the set of permutations of the vector $(1,\dots,n)^T$. This problem is used for example to reduce the envelope of sparse matrices and is shown in George and Pothen (1997, Th. 2.2) to be NP-complete. In Chapter 3, we show that solving the seriation problem is equivalent to solving 2-SUM for a large class of similarity matrices. A more general result was recently published by (Laurent and Seminaroti, 2015). Intuitively, minimizing 2-SUM results in assigning nearby positions, i.e., low distance $(\pi(i)-\pi(j))^2$, to items that have high similarity A_{ij}.

Similarity matrices. Solving seriation relies on the underlying assumption that the similarity matrix is consistent with a global order. In the ideal setting (with no noise), the similarity matrix should respect the following Robinson property:

Definition 1.1. (R-matrices) We say that the matrix $A \in \mathbf{S}_n$ is an R-matrix (or Robinson matrix) if and only if it is symmetric and satisfies $A_{i,j} \leq A_{i,j+1}$ and $A_{i+1,j} \leq A_{i,j}$ in the lower triangle, where $1 \leq j < i \leq n$.

Another way to write the R-matrix conditions is to impose $A_{ij} \geq A_{kl}$ if $|i-j| \leq |k-l|$ off-diagonal, i.e., the coefficients of A decrease as we move away from the diagonal (cf. Figure 1.2). In that sense, R-matrices are similarity matrices between variables organized on a *chain*, i.e., where the similarity A_{ij} is monotonically decreasing with the distance between i and j on this chain.

As in (Atkins et al., 1998), we will say that A is *pre-R* if and only if there is a permutation Π such that $\Pi A \Pi^T$ is an R-matrix.

Naive greedy. Suppose we know which item comes first (or last) in the order we seek to reconstruct. A very naive approach to solve the seriation problem is to pick the item that is

most similar to the first item, and repeat the procedure until no item is left. Even if we do not know which item comes first, it seems this procedure could be easily extended by first picking randomly an item, and then "growing" two branches in the same manner as before. We will not go into details, but this greedy approach has obvious limitations. In particular, as soon as the *pre-R* property is not exactly satisfied by the input similarity matrix, this method provides very poor reordering, because only local similarities are taken into account. In the following we consider methods that aim at optimizing the global objective 2-SUM and have therefore better performance for similarity matrices which are not exactly *pre-R*.

Spectral ordering. The basis of Chapters 3 and 4 is a spectral relaxation of 2-SUM that provides an exact solution to the seriation problem in the noiseless setting, i.e., for *pre-R* matrices (Atkins et al., 1998), and gives good approximations in the presence of bounded noise.

Let us first define the Fiedler vector of a (irreducible) matrix.

Definition 1.2. The Fiedler value of a symmetric, nonnegative matrix A is the smallest nonzero eigenvalue of its Laplacian $L_A = \mathbf{diag}(A\mathbf{1}) - A$. The corresponding eigenvector is called Fiedler vector and is the optimal solution to

$$\begin{array}{ll} \text{minimize} & y^T L_A y \\ \text{subject to} & y^T \mathbf{1} = 0, \|y\|_2 = 1. \end{array} \tag{1.7}$$

in the variable $y \in \mathbb{R}^n$.

We now recall the main result from (Atkins et al., 1998), which shows how to reorder *pre-R* matrices in a noise free setting.

Proposition 1.3. Atkins et al. (1998, Th. 3.3) *Suppose $A \in \mathbf{S}_n$ is a pre-R-matrix, with a simple Fiedler value whose Fiedler vector v has no repeated values. Suppose that Π is a permutation matrix such that the permuted Fielder vector Πv is strictly monotonic, then $\Pi A \Pi^T$ is an R-matrix.*

The results in (Atkins et al., 1998) thus provide a polynomial time solution to the R-matrix ordering problem in a noise free setting.[2] While Atkins et al. (1998) also show how to handle cases where the Fiedler vector is degenerate, these scenarios are highly unlikely to arise in settings where observations on A are noisy and we refer the reader to Atkins et al. (1998, §4) for details.

[2] Extremal eigenvalues of dense matrices can be computed by randomized polynomial time algorithms with complexity $O(n^2 \log n)$ (Kuczynski and Wozniakowski, 1992).

In Chapter 3, we produce more refined convex relaxations of 2-SUM. Many of them can be directly adapted to other objective functions. Our goal is to improve robustness to noise and add a priori constraints into the optimization problem, with DNA applications in mind.

1.3.3 Ranking from pairwise comparisons

Problem statement

In Chapter 4, we study the problem of ranking a set of n items given pairwise comparisons between these items, and relate it to the seriation problem studied in Chapter 3. The problem of aggregating binary relations has been formulated more than two centuries ago, in the context of emerging social sciences and voting theories (de Borda, 1781; de Condorcet, 1785). The setting we consider goes back at least to (Kendall and Smith, 1940). In this case, the directed graph of all pairwise comparisons, where every pair of vertices is connected by exactly one of two possible directed edges, is usually called a *tournament* graph in the theoretical computer science literature, or a "round robin" in sports, where every player plays every other player once and each preference marks victory or defeat. The motivation for this formulation often stems from the fact that in many applications, e.g., music, images, and movies, preferences are easier to express in relative terms (e.g., a is better than b) rather than absolute ones (e.g., a should be ranked fourth, and b seventh). In practice, the information about pairwise comparisons is usually *incomplete*, especially in the case of a large set of items, and the data may also be *noisy*, that is some pairwise comparisons could be incorrectly measured and inconsistent with a total order.

Related work

We present here a representative panel of existing methods, though not exhaustive due to the very rich literature in this field.

Ranking is a classical problem but its formulations vary widely. Website ranking methods such as PageRank (Page et al., 1998) and HITS (Kleinberg, 1999) seek to rank web pages based on the hyperlink structure of the web, where links do not necessarily express consistent preference relationships (e.g., a can link to b and b can link c, and c can link to a). Assumptions about how the pairwise preference information is obtained also vary widely. A subset of preferences is measured adaptively in (Ailon, 2011; Jamieson and Nowak, 2011), while (Freund et al., 2003; Negahban et al., 2012) extract them at random. In other settings, the full preference matrix is observed, but is perturbed by noise: in e.g., (Bradley and Terry, 1952; Luce, 1959; Herbrich et al., 2006), a parametric model is assumed over the set of permutations, which reformulates ranking as a maximum likelihood problem.

Loss functions, performance metrics and algorithmic approaches vary as well. Kenyon-Mathieu and Schudy (2007), for example, derive a PTAS for the minimum feedback arc set problem on tournaments, i.e., the problem of finding a ranking that minimizes the number of upsets (a pair of players where the player ranked lower on the ranking beats the player ranked higher). In practice, the complexity of this method is relatively high, and other authors (see for instance Keener, 1993; Negahban et al., 2012) have been using spectral methods to produce more efficient algorithms (each pairwise comparison is understood as a link pointing to the preferred item). In other cases, such as the classical Analytic Hierarchy Process (AHP) (Saaty, 1980; Barbeau, 1986) preference information is encoded in a "reciprocal" matrix whose Perron-Frobenius eigenvector provides the global ranking. Simple scoring methods such as the point difference rule (Huber, 1963; Wauthier et al., 2013) produce efficient estimates at very low computational cost. Ranking has also been approached as a prediction problem, i.e., learning to rank (Schapire et al., 1998; Rajkumar and Agarwal, 2014), with (Joachims, 2002) for example using support vector machines to learn a score function. Recent work by Sibony et al. (2015) develop a multi-resolution analysis in order to produce rankings from incomplete data. Finally, in the Bradley-Terry-Luce framework, where multiple observations on pairwise preferences are observed and assumed to be generated by a generalized linear model, the maximum likelihood problem is usually solved using fixed point algorithms or EM-like majorization-minimization techniques (Hunter, 2004).

Standard approaches

We now briefly recall the standard methods to which we compare our proposed algorithm SerialRank in Chapter 4. We refer to the forthcoming book of Milan Vojnovic on contest theory for a complete survey (Vojnovic, 2015).

Point score. Suppose we want to rank n items based on pairwise comparisons. Denote by $C \in \{-1, 0, 1\}^{n \times n}$ the matrix of pairwise comparisons, such that $C_{i,j} = -C_{j,i}$ for every $i \neq j$ and

$$C_{i,j} = \begin{cases} 1 & \text{if } i \text{ is ranked higher than } j, \\ 0 & \text{if } i \text{ and } j \text{ are not compared or in a draw,} \\ -1 & \text{if } j \text{ is ranked higher than } i, \end{cases} \tag{1.8}$$

setting $C_{i,i} = 1$ for all $i \in \{1, \dots, n\}$.[3]

Many scoring methods have been proposed to produce rankings from pairwise comparisons. The well-known point difference rule, also called point score, or Borda Count simply counts the number of "victories" minus the number of "defeats" for each item, i.e., computes $C\mathbf{1}$, and then sorts items by score.

[3] Note that algorithms presented here can be adapted to settings where comparisons have continuous values, i.e., $C_{ij} \in [-1, 1]$, for instance in tournaments where outcomes of several games are averaged for each pair of players.

Although it is quite robust to missing comparisons (*cf.* the following paragraph on ranking approximations), when some comparisons are not consistent with a global ranking, i.e., form cycles ($a > b$, $b > c$, but $a < c$), the point score rule is often less accurate that the more refined methods that follow.

Bradley-Terry-Luce model. The Bradley-Terry-Luce model (BTL) assumes there are parameters $\theta_1, \dots, \theta_n$ such that for any given pair of items $i \neq j$, item i is compared favorably to item j with probability

$$p_{ij} = \frac{\theta_i}{\theta_i + \theta_j}.$$

In the context of a tournament, the underlying parameters θ_i can be seen as the inherent *skills* of the players. Estimation can be made via maximum log-likelihood estimation

$$\hat{\theta} \in \arg\min_{\theta \in \mathbb{R}^n} \sum_{i<j} \Big(\log(1 + \exp(\theta_j - \theta_i)) - \hat{P}_{ij}(\theta_j - \theta_i) \Big),$$

where $\hat{P}_{ij} = \frac{C_{ij}+1}{2}$. BTL model then outputs a ranking by sorting the estimated score vector $\hat{\theta}$.

We refer to (Hunter, 2004) for EM-like majorization-minimization techniques. Simple fixed point algorithms are also detailed in (Vojnovic, 2015).

PageRank, Rank Centrality. Website ranking methods such as PageRank (Page et al., 1998) and HITS (Kleinberg, 1999) seek to rank web pages based on the hyperlink structure of the web, where links do not necessarily express consistent preference relationships (e.g., a can link to b and b can link c, and c can link to a). A transition matrix between web pages is computed based on links, and a global ranking is retrieved by reordering the stationary probabilities of the associated Markov chain.

In recent work, Negahban et al. (2012) analyzed a ranking algorithm based on the same principles. In the setting they consider, k comparisons are observed for a subset of all pairs of items, which enables them to define an empirical transition matrix A for observed pairs

$$A_{ij} = \frac{1}{k} \sum_{l=1}^{n} \frac{C_{ij}^{(l)} + 1}{2},$$

where $C_{ij}^{(l)} \in \{-1, 1\}$ is the l^{th} comparison between i and j. Entries of A with no observed comparisons are set to 0. In order to obtain a probability matrix P associated with a random walk, A is then scaled by the maximum out degree of a node d_{max}, i.e.,

$$P_{ij} = \begin{cases} \frac{1}{d_{max}} A_{ij} & \text{if } i \neq j, \\ 1 - \frac{1}{d_{max}} \sum_{k \neq i} A_{ik} & \text{if } i = j. \end{cases}$$

This ensures that rows of P sum to one. The stationary distribution is then computed by extracting the top left eigenvector of P, which upon sorting, induces a ranking of the n items.

The intuition behind *network centrality* algorithms is that comparisons define a random walk over the graph of items (similarly as links between webpages in PageRank algorithm). In the case of a tournament, at each iteration of the random walk, the probability of transitioning from vertex i to vertex j is directly proportional to how often player j beat player i across all games between the two players, and is 0 if they never confronted. Hence, the random walk has a higher probability of transitioning to a more skillful player. The stationary distribution of the associated Markov chain, which reflects the frequencies of visits for each node, gives a scoring of the players, hence a ranking.

New spectral methods. In Chapter 4, we present a new ranking algorithm called SerialRank, based on the seriation spectral relaxation presented in Section 1.3.2 [4]. We notably show how to construct a similarity matrix based on pairwise comparisons.[5] When ranking players, the similarity measure between two players is the number of similar outcomes against other opponents. Intuitively, players that beat the same players and are beaten by the same players should have a similar ranking.

The presentation of SerialRank algorithm is complemented by a careful analysis of the method in the presence of missing and corrupted comparisons as well as numerical experiments whose results often match or outperform existing approaches, depending on the setting.

Other spectral algorithms which can be seen as variations of SerialRank are the object of ongoing work. We notably refer the interested reader to the "SVD ranking" briefly presented in (Cucuringu, 2015).

Measuring the quality of a ranking. A natural question that arises is how to assess the quality of a retrieved ranking. When the true ranking is known, it is common to count the number of inverted pairs of the retrieved ranking $\hat{\pi}$, relative to the true ranking π^*, known as the Kendall τ distance

$$d_{\text{Kendall}}(\pi^*, \hat{\pi}) = \sum_{\pi^*(i)<\pi^*(j)} \mathbf{1}(\hat{\pi}(j) < \hat{\pi}(i)).$$

Another common measure is the Spearman's footrule

$$d_{\text{Spearman}}(\pi^*, \hat{\pi}) = \sum_{j=1}^{n} |\hat{\pi}(j) - \pi(j)^*|.$$

[4] A short tutorial including python code is also available at http://www.di.ens.fr/~fogel/SerialRank/tutorial.html.

[5] SerialRank algorithm is also briefly introduced in Section 1.5.

As shown by Diaconis and Graham (1977), Spearman's footrule is related to Kendall τ distance as $d_{\text{Kendall}} < d_{\text{Spearman}} < 2d_{\text{Kendall}}$. Therefore bounds on the Kendall τ can translate on the Spearman's footrule.

For practical applications the true ranking is unknown. As in the computation of the Kendall τ distance, the quality of a ranking $\hat{\pi}$ is often measured by counting the numbers of inverted pairs of the ranking relative to the input comparisons

$$dis(\hat{\pi}) = \sum_{C(i,j)<0} \mathbf{1}(\hat{\pi}(j) < \hat{\pi}(i)).$$

Sample complexity bounds. A number of authors have derived approximation guarantees for ranking algorithms. In particular, assuming there exists a true ranking, it is of much interest to know how many comparisons are needed in order to retrieve a ranking that is not too distant from the true ranking. A common approach to answer this question is to associate pairwise comparisons with directed edges in an Erdös-Rényi graphs, i.e., pairwise comparisons are observed independently with a given probability q. Since Erdös-Rényi graphs are connected with high probability only when the total number of pairs sampled scales as $\Omega(n \log n)$ (Erdös and Rényi, 1960), we need at least that many comparisons in order to retrieve a ranking.

While many authors derive bounds on the ℓ_2 distance between the retrieved ranking and the true ranking, i.e., bound $\|\hat{\pi} - \pi^*\|_2$, some bound the largest displacement, i.e., the ℓ_∞ distance $\max_j |\hat{\pi}(j) - \pi^*(j)|$, which gives stronger and more interpretable results. Formulations of sample complexity bounds vary among authors. For instance, Wauthier et al. (2013) show that for $0 < \nu < 1$, sampling $\Omega\left(\frac{n \log n}{\nu^2}\right)$ comparisons guarantees that $\max_j |\hat{\pi}(j) - \pi^*(j)| < \nu n$ with high probability for n large enough.

We refer to the recent work of Rajkumar and Agarwal (2014) for a survey of sample complexity bounds for Rank centrality, BTL and an SVM based ranking.

Note that empirical evaluation is also needed in order to detect more subtle differences between competing methods.

1.4 Challenges

Through the study of the phase retrieval, seriation and ranking problems, this thesis attempts to take up three major challenges in modern optimization: how to design algorithms to solve problems with high complexity that are *robust*, *scalable* and *flexible*.

1.4.1 Robustness

The first issue that we have encountered in trying to solve the phase retrieval, seriation and ranking problems is robustness. These three problems can be solved quite easily, e.g., using greedy schemes, when observations are consistent with model assumptions, without noise, and come in sufficiently large number, which we call "nice" settings. However, their complexity increases drastically in the presence of noise or incomplete information [6], and performance of greedy schemes deteriorates very quickly. Ideally we would like to provide robust algorithms, i.e., that compute the optimal solution in "nice" settings, and good approximations in the presence of bounded noise and incomplete information.

We now detail more specifically the issues of noise on measurements, model assumptions' violation, and lack of information for our three problems.

1.4.1.1 Noise on measurements.

In most applications, measurements come with some noise.

- In phase retrieval, the precision of the physical instruments used in the experiments is limited.
- In DNA sequencing, sequencing machines produce short segments (reads) that can contain deletions, insertions or substitutions of some base pairs A, C, T, G.
- When ranking players, outcomes of games can be influenced by external random events.

The level of noise varies among applications, but is in most cases high enough to break the most naive greedy schemes. Fortunately we will see in next chapters that our proposed methods are quite robust to (bounded) noise on measurements.

1.4.1.2 Ill-posed model

A more difficult issue to deal with is the discrepancy between data and model assumptions.

- For molecular imaging, the difficulty lies in the fact that an experimental setting with several coded diffraction patterns is not yet fully available, and we therefore had to use

[6] The seriation problem was shown to be NP-complete by George and Pothen (1997), finding a Kemeny optimal ranking was shown to be NP-hard by Dwork et al. (2001b), the complexity of the phase retrieval problem decreases with the number of observations (see (Waldspurger and Mallat, 2012; Candes et al., 2015a; Candes and Li, 2014) for conditions on the uniqueness of the recovery).

simulations, which have of course their own limitations. At the time of writing, a real experimental setup reproducing the simulations described in Chapter 2 is being implemented and tested by Matthew Seaberg at SLAC (Stanford University).

The seriation and ranking problems are generally formulated with the goal of finding a global order that is most consistent with the data. However this underlying assumption of a global order is intrinsically not correct in many applications, even when measurements come with no noise.

- For instance, in DNA de novo sequencing it often occurs that the DNA strain contains repetitions, inducing ambiguities when reordering reads (a read can be highly similar to a read that is very far away from its true location in the original DNA strain).
- Similarly, in ranking applications, comparisons often comprise a high number of cycles, i.e., $a > b$, $b > c$ but $c > a$ that are not compatible with a global ranking.

1.4.1.3 Lack of information

The last robustness requirement is to design methods that can provide reasonable results when the number of measurements is very small.

- For molecular imaging, the number of illuminations is limited by the experimental framework feasibility. The less illuminations the more realistic, since molecules are deteriorated by each successive illumination.
- For ranking, it is very common that only a small fraction of all pairwise comparisons are available, especially in large-scale datasets, and the recovery of the ranking in such settings is of primary concern.

1.4.2 Scalability

There is a natural tradeoff between using methods that are simple, but scalable, and methods that can give better approximations, but at a higher computational cost. As a consequence, solving large-scale problems often requires to use problem-specific optimization procedures.

- For the phase retrieval problem, PhaseCut SDP relaxation from Waldspurger et al. (2015) turns out to be too expensive for realistically large problems, if relying on standard solvers using interior point methods. However we show in Chapter 2 how using specific structural assumptions on the data as well as a fast block coordinate descent algorithm can drastically push forward size limitations.

- For the seriation problem, we propose several convex relaxations, the most sophisticated ones having better approximation guarantees, but more prohibitive computational cost. Large-scale DNA sequencing experiments are performed thanks to the initial use of a spectral relaxation to first reduce problem size, followed by more refined relaxations.
- For ranking, we focus on a spectral relaxation which scales very well for sparse data.

1.4.3 Flexibility

In addition to robustness and scalability, flexibility is here the key property for algorithms to perform well on applications. By flexibility we mean the possibility to add problem-specific support and a priori constraints into the optimization.

- For the phase retrieval problem, we have tested several support constraints.
- For the DNA sequencing problem, implementing in the algorithm a priori constraints coming from additional information from sequencing machines (mate-reads) has turned out to be primordial.

1.4.4 Pros and cons of convex and spectral relaxations

Having defined our goals, we can now argue on which class of methods to develop for the three studied problems. More specifically, why use convex or spectral relaxations?

- Convex relaxations have the appealing feature to rely on the well-studied and understood convex optimization tools. Deriving approximation guarantees for convex relaxations has proven to be very efficient, especially for SDP relaxations (see for instance MAXCUT approximation from Goemans and Williamson (1995), briefly recalled in Section 1.6). Moreover, convex relaxations are usually very stable compared to greedy algorithms. On the other hand, one may argue that convex approximations techniques are too sophisticated for real problems, the empirical gain over simpler methods not being very significative, and not adapted to large-scale applications.
- Spectral relaxations have the advantage to be usually more scalable than convex relaxations, at the cost of looser approximation ratios. There are many examples of successful usage of spectral algorithms, e.g., PageRank (Page et al., 1998) and spectral clustering (Ng et al., 2002; Bach and Jordan, 2004; Von Luxburg, 2007).

In the following, we will try to keep a pragmatic point of view, following Occam's razor principle, and privileging methods that both perform well in practice and can be analyzed theoretically.

1.4.5 Three essential steps

Besides classical "guarantees vs. applicability" tradeoffs, we would like to emphasize three steps that have proven essential for solving the problems described in this manuscript: pre-processing, initialization and post-processing.

Pre-processing. *Pre-processing* traditionally refers to filtering techniques used for instance to remove outliers from data. We give here to *pre-processing* the broader meaning of improving the quality of available data using problem-specific techniques. For instance, in the ranking problem, as will be seen in Chapter 4, computing a similarity matrix based on comparisons can be seen as a way to make data more consistent with a global order. For DNA assembly, the design of the similarity used to retrieve the order of the reads is also a crucial step and is the object of current work with Antoine Recanati (E.N.S.), Alexandre d'Aspremont (E.N.S.) and Thomas Bruls (Génoscope).

Initialization. For non-convex optimization problems, *initialization* is the key step for algorithms to converge to a local optimum not too far from the global optimum. As will be seen in following Chapters, convex and spectral relaxations can be seen as methods that provide good *initializations*.

Post-processing. Naturally, if convex and spectral relaxations provide good *initialization*, this implies that they must be followed by refinements, or *post-processing*. For instance, in the case of phase retrieval, using the greedy Fienup algorithm initialized by PhaseCut SDP relaxation turns out to be a good combination compared to the use of Fienup alone or PhaseCut alone.

Of course, one may argue that the most important step is the *acquisition of data*, but this issue is out of scope for this manuscript.

1.5 Connection between phase retrieval, seriation and ranking

In this section, we first show how to formulate the seriation problem as a phase retrieval problem (for similarities that reflect Euclidian distances). Then we detail how to construct a similarity matrix from pairwise comparisons and cast the ranking problem as a seriation problem. The connection between the ranking problem and the seriation problem will be further investigated in Chapter 4.

1.5.1 Seriation as a phase retrieval problem

Recall that the seriation problem seeks to recover a linear ordering of n items, given pairwise similarities between these items, while the phase retrieval problem seeks to reconstruct a complex signal, given m observations on the magnitude of linear measurements, i.e., solve

$$\begin{aligned} &\text{find} && x \in \mathbb{C}^n \\ &\text{such that} && |Ax| = b, \end{aligned} \tag{1.9}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

Given a seriation problem with similarity matrix S, let us define a distance matrix Δ as the opposite of the similarity matrix S (plus a constant term to ensure non-negativity)

$$\Delta_{i,j} = \max_{(k,l)} S_{k,l} - S_{i,j}.$$

Suppose that items can be represented on a line, with Euclidian distances between items given by Δ. Formally, there exists a vector $x \in \mathbb{R}^n$ respecting

$$|x_i - x_j| = \Delta_{i,j} \text{ for all pairs } (i,j) \in \{1, \dots, n\}^2.$$

Finding such a linear representation x is an instance of the phase retrieval problem (1.9), with the additional support constraint that $x \in \mathbb{R}^n$.[7] A linear representation of items x directly translates into a linear ordering by sorting x.

One advantage of this formulation is that it allows us to solve seriation problems even if only a subset of pairwise similarities are given. Specifically we can rewrite the seriation problem as

$$\begin{aligned} &\text{find} && x \in \mathbb{R}^n \\ &\text{such that} && |x_i - x_j| = \Delta_{i,j} \text{ for all pairs } (i,j) \text{ s.t. } S_{i,j} \text{ is given.} \end{aligned} \tag{1.10}$$

This formulation of the seriation problem as a phase retrieval problem is limited by the assumption that similarities reflect Euclidian distances. Our ongoing work tries to extend this formulation to a broader class of similarities. In particular, we would like to use phase retrieval algorithms studied in Chapter 2 to solve seriation problems, with similar guarantees on both the number of observations required for recovery of the order, and robustness to noise.

[7] Allowing x to have complex values would yield a representation of items in the complex plane.

1.5.2 Ranking as a seriation problem

We now reformulate the problem of ranking from pairwise comparisons as a seriation problem. Both seriation and ranking seek to recover a linear order of a set of items. The only difference is that ranking is based on pairwise comparisons, while seriation is based on pairwise similarities.

Given an ordered input pairwise comparison matrix, we now show how to construct a similarity matrix which is *pre-R* [8] when all comparisons are given and consistent with the identity ranking (i.e., items are ranked in increasing order of indices). This means that the similarity between two items decreases with the distance between their ranks. We will then be able to use the spectral seriation algorithm in (Atkins et al., 1998) described in Section 4.3 to reconstruct the true ranking from a disordered similarity matrix.

We show here how to compute a pairwise similarity from binary comparisons between items by counting the number of matching comparisons. Another formulation allows to handle the generalized linear model and will be detailed in Chapter 4. These two examples are only two particular instances of a broader class of ranking algorithms. Any method which produces R-matrices from pairwise preferences yields a valid ranking algorithm.

Suppose we are given a matrix of pairwise comparisons $C \in \{-1, 0, 1\}^{n \times n}$ such that

$$C_{i,j} = -C_{j,i} \quad \text{for every } i \neq j$$

and

$$C_{i,j} = \begin{cases} 1 & \text{if } i \text{ is ranked higher than } j, \\ 0 & \text{if } i \text{ and } j \text{ are not compared or in a draw}, \\ -1 & \text{if } j \text{ is ranked higher than } i, \end{cases} \tag{1.11}$$

setting $C_{i,i} = 1$ for all $i \in \{1, \dots, n\}$. We define the pairwise similarity matrix S^{match} as

$$S^{\text{match}}_{i,j} = \sum_{k=1}^{n} \left(\frac{1 + C_{i,k} C_{j,k}}{2} \right). \tag{1.12}$$

Since $C_{i,k}C_{j,k} = 1$, if $C_{i,k}$ and $C_{j,k}$ have matching signs, and $C_{i,k}C_{j,k} = -1$ if they have opposite signs, $S^{\text{match}}_{i,j}$ counts the number of matching comparisons between i and j with other reference items k. If i or j is not compared with k, then $C_{i,k}C_{j,k} = 0$ and the term $(1 + C_{i,k}C_{j,k})/2$ has an average effect on the similarity of $1/2$. Note that we also have

$$S^{\text{match}} = \frac{1}{2} \left(n \mathbf{1}\mathbf{1}^T + CC^T \right). \tag{1.13}$$

The intuition behind the similarity S^{match} is easy to understand in a tournament setting: players that beat the same players and are beaten by the same players should have a similar ranking.

[8] See Section 1.3.2 Definition 1.1.

The next result shows that when all comparisons are given and consistent with the identity ranking, then the similarity matrix S^{match} is an R-matrix. Without loss of generality, we assume that items are ranked in increasing order of their indices. In the general case, we can simply replace the *R* property by the *pre-R* property.

Proposition 1.4. *Given all pairwise comparisons $C_{i,j} \in \{-1, 0, 1\}$ between items ranked according to the identity permutation (with no ties), the similarity matrix S^{match} constructed in (1.12) is an R-matrix and*

$$S^{\text{match}}_{i,j} = n - |i - j| \tag{1.14}$$

for all $i, j = 1, \ldots, n$.

Proof. Since items are ranked as $1, 2, \ldots, n$ with no ties and all comparisons given, $C_{i,j} = -1$ if $i < j$ and $C_{i,j} = 1$ otherwise. Therefore we obtain from definition (1.12)

$$\begin{aligned} S^{\text{match}}_{i,j} &= \sum_{k=1}^{\min(i,j)-1} \left(\frac{1+1}{2}\right) + \sum_{k=\min(i,j)}^{\max(i,j)-1} \left(\frac{1-1}{2}\right) + \sum_{k=\max(i,j)}^{n} \left(\frac{1+1}{2}\right) \\ &= n - (\max\{i,j\} - \min\{i,j\}) \\ &= n - |i - j| \end{aligned}$$

This means in particular that S^{match} is strictly positive and its coefficients are strictly decreasing when moving away from the diagonal, hence S^{match} is an R-matrix. ∎

We will see in Chapter 4 that these definitions can be directly extended to settings where multiple comparisons are available for each pair and aggregated in comparisons that take fractional values (e.g., tournament setting where participants play several times against each other).

1.6 Standard convex and spectral relaxation techniques

We now briefly recall standard convex and spectral relaxation of the MAXCUT and balanced MINCUT problems. These relaxations have a lot in common with those developed in following chapters and are a pre-requisite for a good comprehension of this manuscript. We refer the reader to (d'Aspremont and Boyd, 2003) for a brief survey of quadratically constrained quadratic programs (QCQP) relaxations, from which this section is very much inspired, and to (Von Luxburg, 2007) for an introduction to spectral clustering. Readers already familiar with these techniques can skip this section.

1.6.1 Cuts

The techniques we use to solve the phase retrieval and seriation problems are very much similar to relaxations of two partitioning problems involving cuts: MAXCUT, and balanced MINCUT. Note however that the phase retrieval and seriation problems are different from the MAXCUT and MINCUT problems. In particular, the phase retrieval relaxation PhaseCut presented in chapter 2 does not involve any graph and associated Laplacian matrix.

1.6.1.1 Partitioning problems

Let us first review the general partitioning problem. We consider here the two-way partitioning problem

$$\begin{array}{ll} \text{minimize} & x^T L x \\ \text{subject to} & x_i^2 = 1, \ i = 1, \ldots, n. \end{array} \qquad \text{(Partitioning)}$$

with variable $x \in \mathbb{R}^n$, where $L \in \mathbf{S}^n$. This problem is a non-convex QCQP. A feasible x corresponds to the partition

$$\{1, ..., n\} = \{i \mid x_i = -1\} \cup \{i \mid x_i = 1\},$$

and the matrix coefficient L_{ij} can be interpreted as the cost of having the elements i and j in the same group, with $-L_{ij}$ the cost of having i and j in different groups. The objective in (Partitioning) is the total cost, over all pairs of elements, and problem (Partitioning) seeks to find the partition with least total cost. Since the feasible set is finite (it contains 2^n points), the problem can in principle be solved by checking the objective value of all feasible points. However, since the number of feasible points grows exponentially, this is possible only for small problems (say, with $n \leq 30$) and problem (Partitioning) is in general very difficult to solve.

1.6.1.2 MAXCUT

We now present the MAXCUT problem, which is a special case of the partitioning problem. Given a graph G with n nodes, we define nonnegative weights a_{ij} associated with each edge (i, j), where $a_{ij} = 0$ if no edge connects nodes i and j. The MAXCUT problem seeks to find a cut of the graph with the largest possible weight, i.e., a partition of the set of nodes in two parts G_1, G_2 such that the total weight of all edges linking these parts is maximized. MAXCUT is a classic problem in network optimization. The weight of a particular cut x is given by

$$\frac{1}{2} \sum_{\{i,j \mid x_i x_j = -1\}} a_{ij}$$

which is also equal to

$$\frac{1}{4}\sum_{i,j=1}^{n} a_{ij}(1 - x_i x_j),$$

Defining the (unnormalized) Laplacian matrix $L = \mathbf{diag}\,(A\mathbf{1}) - A$, i.e., with entries $L_{ij} = -a_{ij}$ if $i \neq j$ and $L_{ii} = \sum_{j=1}^{n} a_{ij}$, the problem is then formulated as

$$\begin{array}{ll} \text{maximize} & x^T L x \\ \text{subject to} & x_i^2 = 1,\ i = 1, \ldots, n, \end{array} \qquad \text{(MAXCUT)}$$

with variable $x \in \mathbb{R}^n$. Hence we retrieve the formulation of a partitioning problem . Note that here L is positive semidefinite.

1.6.1.3 MINCUT

Inversely to the MAXCUT problem, the MINCUT problem seeks to find a cut of the graph with the smallest possible weight, i.e., a partition of the set of nodes in two parts G_1, G_2 such that the total weight of all edges linking these parts is mnimized. MINCUT is a classic problem in network optimization and is equivalent to the maximum flow problem, which can be solved in polynomial time (Ford and Fulkerson, 1962). See for instance (Stoer and Wagner, 1997) for the description of a fast algorithm.

1.6.1.4 Balanced MINCUT

When the goal is to partition a graph into two clusters of nodes with minimal cut weight, it is usually required that the two clusters have the same number of elements. This problem is known as Balanced MINCUT and is formulated as

$$\begin{array}{ll} \text{minimize} & x^T L x \\ \text{subject to} & x_i^2 = 1,\ i = 1, \ldots, n \\ & \sum_{i=0}^{n} x_i = 0, \end{array} \qquad \text{(Balanced MINCUT)}$$

with variable $x \in \mathbb{R}^n$.

Unlike the MINCUT problem, the Balanced MINCUT problem is known to be NP-hard, see (Wagner and Wagner, 1993) for a discussion. Other related cut problems have been studied, notably ratio cut (Hagen and Kahng, 1992) and normalized cut (Shi and Malik, 2000).

We now present the classical relaxations of the MAXCUT and Balanced MINCUT problems.

1.6.2 Relaxations of MAXCUT

1.6.2.1 SDP relaxation

MAXCUT problem can be rewritten

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(Lxx^T) \\ \text{subject to} & x_i^2 = 1,\ i = 1, \dots, n. \end{array} \tag{MAXCUT}$$

Defining the lifted matrix $X = xx^T$, we obtain

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(LX) \\ \text{subject to} & X \succeq 0 \\ & \text{rank}(X) = 1 \\ & X_{ii} = 1,\ i = 1, \dots, n. \end{array} \tag{1.15}$$

Dropping the rank constraint, MAXCUT problem can be relaxed into the SDP

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(LX) \\ \text{subject to} & X \succeq 0 \\ & X_{ii} = 1,\ i = 1, \dots, n. \end{array} \tag{MAXCUT-SDP}$$

1.6.2.2 Projection and randomization

Given the optimal solution X to (MAXCUT-SDP), we can retrieve the eigenvector x corresponding to the largest eigenvalue of X and then use the sign function to project it on $\{-1, 1\}^n$. In order to improve further the solution of the SDP, we can use randomization: sample points x_i with a normal distribution $\mathcal{N}(0, X)$, get feasible points by taking $\hat{x}_i = \text{sign}(x_i)$, then keep the feasible point with highest objective value.

1.6.2.3 Approximation bounds

When $\hat{x}$ is sampled using that procedure, the expected value of the objective $\mathbb{E}(x^T Lx)$ can be computed explicitly (Goemans and Williamson, 1995):

$$\mathbb{E}(x^T Lx) = \frac{2}{\pi} \sum_{i,j=1}^{n} L_{ij} \arcsin X_{ij} = \frac{2}{\pi} \mathbf{Tr}(L \arcsin X).$$

We are guaranteed to reach this expected value $\frac{2}{\pi} \mathbf{Tr}(L \arcsin X)$ after sampling a few (feasible) points $\hat{x}$, hence we know that the optimal value OPT of the MAXCUT problem is between

$\frac{2}{\pi}\,\mathbf{Tr}(L \arcsin X)$ and $\mathbf{Tr}(LX)$. Furthermore, with $\arcsin(X) \succeq X$ (see Ben-Tal and Nemirovski, 2001, p. 174), we can simplify (and relax) the above expression to get

$$\frac{2}{\pi}\,\mathbf{Tr}(LX) \leq OPT \leq \mathbf{Tr}(LX).$$

This means that the procedure detailed above guarantees that we can find a feasible point that is at most $\frac{2}{\pi}$ suboptimal (after taking a certain number of samples from a Gaussian distribution).

1.6.2.4 Spectral relaxation of MAXCUT

Replacing in MAXCUT the constraint $x_1^2 = 1, \ldots, x_n^2 = 1$ with $\sum_{i=1}^n x_i^2 = n$, we obtain the relaxed problem

$$\begin{array}{ll} \text{maximize} & x^T L x \\ \text{subject to} & \sum_{i=1}^n x_i^2 = n. \end{array}$$

This just amounts to finding the eigenvector of the Laplacian associated with the largest eigenvalue. We can then use the sign function for projection. Note that this relaxation gives another (weaker) upper bound for the MAXCUT problem.

1.6.3 Spectral relaxation of balanced MINCUT

1.6.3.1 Spectral relaxation

Allowing the vector x to have continuous values, the Balanced MINCUT problem can be relaxed into the spectral clustering problem

$$\begin{array}{ll} \text{minimize} & x^T L x \\ \text{subject to} & \sum_{i=1}^n x_i^2 = n \\ & \sum_{i=1}^n x_i = 0. \end{array}$$

Without loss of generality, we can normalize the vector x and obtain

$$\begin{array}{ll} \text{minimize} & x^T L x \\ \text{subject to} & \|x\|_2 = 1 \\ & x^T \mathbf{1} = 0. \end{array} \qquad \text{(Spectral clustering)}$$

Since the first eigenvalue of the Laplacian matrix L is by definition always 0 (with corresponding eigenvector the vector of all ones $\mathbf{1}$), Spectral clustering amounts to finding the eigenvector associated with the second smallest eigenvalue of the Laplacian, a.k.a. the Fiedler vector. We can then use the sign function to get a bi-partition.

Remark 1: connected components. We have implicitly assumed in preceding paragraphs that cuts were applied to connected graphs, i.e., graphs for which there is a path between every pair of nodes. For disconnected graphs, connected components, i.e., sets of connected nodes, can be easily retrieved using either breadth-first search or depth-first search.

Moreover, a very useful property of graph Laplacians is that the number of connected components exactly translates into the multiplicity of the smallest eigenvalue of the Laplacian, which is always zero. Corresponding eigenvectors have binary values encoding the sets of nodes associated with each connected component. This property is the basis of more general spectral clustering algorithms, which seek to cluster nodes based on the representation extracted from eigenvectors of the Laplacian associated with smallest eigenvalues. We refer to the turorial of Von Luxburg (2007) for an introduction to spectral clustering.

Remark 2: perturbation analysis. The formal basis for the perturbation analysis of such spectral relaxations is the Davis-Kahan theorem from matrix perturbation theory (see for instance Stewart and Sun, 1990). This theorem bounds the difference between eigenspaces of symmetric matrices under perturbations, and relate it to the norm of the perturbation and the "eigengap". We will see in Chapter 4 an example of such an analysis.

Chapter 2

Phase Retrieval for Imaging Problems

Chapter abstract: We study convex relaxation algorithms for phase retrieval on imaging problems. We show that exploiting structural assumptions on the signal and the observations, such as sparsity, smoothness or positivity, can significantly speed-up convergence and improve recovery performance. We detail numerical results in molecular imaging experiments simulated using data from the Protein Data Bank (PDB).

The material of this part is based on the following publication:
F. Fogel, I. Waldspurger, A. d'Aspremont, Phase retrieval for imaging problems. To appear in *Mathematical Programming Computation*.

2.1 Introduction

Phase retrieval seeks to reconstruct a complex signal, given a number of observations on the *magnitude* of linear measurements, i.e., solve

$$\begin{array}{ll} \text{find} & x \\ \text{such that} & |Ax| = b \end{array} \tag{2.1}$$

in the variable $x \in \mathbb{C}^p$, where $A \in \mathbb{R}^{n\times p}$ and $b \in \mathbb{R}^n$. This problem has direct applications in X-ray and crystallography imaging, diffraction imaging, Fourier optics or microscopy for example, in problems where physical limitations mean detectors usually capture the intensity of observations but cannot recover their phase. In what follows, we will focus on problems arising in diffraction imaging, where A is usually a Fourier transform, often composed with one or multiple masks. The Fourier structure, through the FFT, considerably speeds up basic linear operations, which allows us to solve large scale convex relaxations on realistically large imaging problems. We will also observe that in many of the imaging problems we consider, the Fourier

transform is very sparse, with *known support* (we lose the phase but observe the magnitude of Fourier coefficients), which allows us to considerably reduce the size of our convex phase retrieval relaxations.

Because the phase constraint $|Ax| = b$ is non-convex, the phase recovery problem (2.1) is non-convex. Several greedy algorithms have been developed (see Gerchberg and Saxton, 1972; Fienup, 1982; Griffin and Lim, 1984; Bauschke et al., 2002, among others), which alternate projections on the range of A and on the non-convex set of vectors y such that $|y| = |Ax|$. While empirical performance is often good, these algorithms can stall in local minima. A convex relaxation was introduced in (Chai et al., 2011) and (Candes et al., 2015a) (who call it PhaseLift) by observing that $|Ax|^2$ is a linear function of $X = xx^*$, which is a rank one Hermitian matrix, using the classical lifting argument for non-convex quadratic programs developed in (Shor, 1987; Lovász and Schrijver, 1991). The recovery of x is thus expressed as a rank minimization problem over positive semidefinite Hermitian matrices X satisfying some linear conditions, i.e., a matrix completion problem. This last problem has received a significant amount of attention because of its link to compressed sensing and the NETFLIX collaborative filtering problem. This minimum rank matrix completion problem is approximated by a semidefinite program which has been shown to recover x for several (random) classes of observation operators A (Candes et al., 2013, 2014, 2015a).

On the algorithmic side, (Waldspurger et al., 2015) showed that the phase retrieval problem (2.1) can be reformulated in terms of a single phase variable, which can be read as an extension of the MAXCUT combinatorial graph partitioning problem over the unit complex torus, allowing fast algorithms designed for solving semidefinite relaxations of MAXCUT to be applied to the phase retrieval problem.

On the experimental side, phase recovery is a classical problem in Fourier optics for example (Goodman, 2008), where a diffraction medium takes the place of a lens. This has direct applications in X-ray and crystallography imaging, diffraction imaging or microscopy (Harrison, 1993; Bunk et al., 2007; Johnson et al., 2008; Miao et al., 2008; Dierolf et al., 2010).

Here, we implement and study several efficient convex relaxation algorithms for phase retrieval on imaging problem instances where A is based on a Fourier operator. We show in particular how structural assumptions on the signal and the observations (e.g., sparsity, smoothness, positivity, known support, oversampling, etc.) can be exploited to both speed-up convergence and improve recovery performance. While no experimental data is available from diffraction imaging problems with multiple randomly coded illuminations, we simulate numerical experiments of this type using molecular density information from the protein data bank (Berman et al., 2002). Our results show in particular that the convex relaxation is stable and that in some settings, as few as two random illuminations suffice to reconstruct the image.

This chapter is organized as follows. Section 3.4 briefly recalls the structure of some key algorithms used in phase retrieval. Section 2.3 describes applications to imaging problems and how structural assumptions can significantly reduce the cost of solving large-scale instances and improve recovery performance. Section 4.6 details some numerical experiments while Section 2.6 describes the interface to the numerical library developed for these problems.

Notations

We write $\mathbf{S}_p$ (resp. $\mathbf{H}_p$) the cone of symmetric (resp. Hermitian) matrices of dimension p ; $\mathbf{S}_p^+$ (resp. $\mathbf{H}_p^+$) denotes the set of positive symmetric (resp. Hermitian) matrices. We write $\|\cdot\|_p$ the Schatten p-norm of a matrix, that is the p-norm of the vector of its eigenvalues (in particular, $\|\cdot\|_\infty$ is the spectral norm). We write $A^\dagger$ the (Moore-Penrose) pseudoinverse of a matrix A, and $A \circ B$ the Hadamard (or componentwise) product of the matrices A and B. For $x \in \mathbb{R}^p$, we write $\mathbf{diag}(x)$ the matrix with diagonal x. When $X \in \mathbf{H}_p$ however, $\mathbf{diag}(X)$ is the vector containing the diagonal elements of X. For $X \in \mathbf{H}_p$, X^* is the Hermitian transpose of X, with $X^* = (\bar{X})^T$. $|\cdot|$ refers to the complex modulus. When x is a vector in $\mathbb{C}^p$, $|x|$ is the vector with coefficients $(|x_1|, \ldots, |x_p|)$. Finally, we write b^2 the vector with components b_i^2, $i = 1, \ldots, n$.

2.2 Algorithms

In this section, we briefly recall several basic algorithmic approaches to solve the phase retrieval problem (2.1). Early methods were all based on extensions of an alternating projection algorithm. However, recent results showed that phase retrieval could be interpreted as a matrix completion problem similar to the NETFLIX problem, a formulation which yields both efficient convex relaxations and recovery guarantees.

2.2.1 Greedy algorithms

The phase retrieval problem (2.1) can be rewritten

$$\begin{array}{ll} \text{minimize} & \|Ax - y\|_2^2 \\ \text{subject to} & |y| = b, \end{array} \tag{2.2}$$

where we now optimize over both phased observations $y \in \mathbb{C}^n$ and signal $x \in \mathbb{C}^p$. Several greedy algorithms attempt to solve this problem using variants of alternating projections, one iteration minimizing the quadratic error (the objective of (2.2)), the next normalizing the moduli (to satisfy the constraint). We detail some of the most classical examples in the paragraphs that follow.

The algorithm Gerchberg-Saxton in (Gerchberg and Saxton, 1972) for instance seeks to reconstruct $y = Ax$ and alternates between orthogonal projections on the range of A and normalization of the magnitudes $|y|$ to match the observations b. The cost per iteration of this method is minimal but convergence (when it happens) is often slow.

Algorithm 5 Gerchberg-Saxton.

Input: An initial $y^1 \in \mathbf{F}$, i.e., such that $|y^1| = b$.

1: **for** $k = 1, \dots, N-1$ **do**

2: Set

$$y_i^{k+1} = b_i \frac{(AA^\dagger y^k)_i}{|(AA^\dagger y^k)_i|}, \quad i = 1, \dots, n. \qquad \text{(Gerchberg-Saxton)}$$

3: **end for**

Output: $y_N \in \mathbf{F}$.

A classical "input-output" variant, detailed here as algorithm Fienup, introduced in (Fienup, 1982), adds an extra penalization step which usually speeds up convergence and improves recovery performance when additional information is available on the support of the signal. Oversampling the Fourier transform forming A in imaging problems usually helps performance as well. Of course, in all these cases, convergence to a global optimum cannot be guaranteed but empirical recovery performance is often quite good.

Algorithm 6 Fienup

Input: An initial $y^1 \in \mathbf{F}$, i.e., such that $|y^1| = b$, a parameter $\beta > 0$.

1: **for** $k = 1, \dots, N-1$ **do**

2: Set

$$w_i = \frac{(AA^\dagger y^k)_i}{|(AA^\dagger y^k)_i|}, \quad i = 1, \dots, n.$$

3: Set

$$y_i^{k+1} = y_i^k - \beta(y_i^k - b_i w_i) \qquad \text{(Fienup)}$$

4: **end for**

Output: $y_N \in \mathbf{F}$.

2.2.2 PhaseLift: semidefinite relaxation in signal

Using a classical lifting argument by (Shor, 1987), and writing

$$|a_i^* x|^2 = b_i^2 \iff \mathbf{Tr}(a_i a_i^* x x^*) = b_i^2$$

(Chai et al., 2011; Candes et al., 2015a) reformulate the phase recovery problem (2.1) as a matrix completion problem, written

$$\begin{array}{ll} \text{minimize} & \mathbf{Rank}(X) \\ \text{subject to} & \mathbf{Tr}(a_i a_i^* X) = b_i^2, \quad i = 1, \ldots, n \\ & X \succeq 0, \end{array}$$

in the variable $X \in \mathbf{H}_p$, where $X = xx^*$ when exact recovery occurs. This last problem can be relaxed as

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(X) \\ \text{subject to} & \mathbf{Tr}(a_i a_i^* X) = b_i^2, \quad i = 1, \ldots, n \\ & X \succeq 0, \end{array} \qquad \text{(PhaseLift)}$$

which is a semidefinite program (called PhaseLift by Candes et al. (2015a)) in the variable $X \in \mathbf{H}_p$. This problem is solved in (Candes et al., 2015a) using first-order algorithms implemented in (Becker et al., 2011). This semidefinite relaxation has been shown to recover the true signal x exactly for several classes of observation operators A (Candes et al., 2015a, 2013, 2014).

2.2.3 PhaseCut: semidefinite relaxation in phase

As in (Waldspurger et al., 2015) we can rewrite the phase reconstruction problem (2.1) in terms of a phase variable u (such that $|u| = 1$) instead of the signal x. In the noiseless case, we then write the constraint $|Ax| = b$ as $Ax = \mathbf{diag}(b)u$, where $u \in \mathbb{C}^n$ is a phase vector, satisfying $|u_i| = 1$ for $i = 1, \ldots, n$, so problem (2.1) becomes

$$\begin{array}{ll} \text{minimize} & \|Ax - \mathbf{diag}(b)u\|_2^2 \\ \text{subject to} & |u_i| = 1 \end{array} \qquad (2.3)$$

where we optimize over both phase $u \in \mathbb{C}^n$ and signal $x \in \mathbb{C}^p$. While the objective of this last problem is jointly convex in (x, u), the phase constraint $|u_i| = 1$ is not.

Now, given the phase, signal reconstruction is a simple least squares problem, i.e., given u we obtain x as

$$x = A^\dagger \, \mathbf{diag}(b)u \qquad (2.4)$$

where $A^\dagger$ is the pseudo inverse of A. Replacing x by its value in problem (2.3), the phase recovery problem becomes

$$\begin{array}{ll} \text{minimize} & u^* M u \\ \text{subject to} & |u_i| = 1, \quad i = 1, \ldots, n, \end{array} \qquad (2.5)$$

in the variable $u \in \mathbb{C}^n$, where the Hermitian matrix

$$M = \mathbf{diag}(b)(\mathbf{I} - AA^\dagger)\,\mathbf{diag}(b)$$

is positive semidefinite. This problem is non-convex in the phase variable u. Waldspurger et al. (2015) detailed greedy algorithm Greedy to locally optimize (2.5) in the phase variable.

Algorithm 7 Greedy algorithm in phase.

Input: An initial $u \in \mathbb{C}^n$ such that $|u_i| = 1$, $i = 1, \ldots, n$. An integer $N > 1$.

1: **for** $k = 1, \ldots, N$ **do**
2: **for** $i = 1, \ldots, n$ **do**
3: Set

$$u_i = \frac{-\sum_{j \neq i} M_{ji} \bar{u}_j}{\left|\sum_{j \neq i} M_{ji} \bar{u}_j\right|} \qquad \text{(Greedy)}$$

4: **end for**
5: **end for**

Output: $u \in \mathbb{C}^n$ such that $|u_i| = 1$, $i = 1, \ldots, n$.

A convex relaxation to (2.5) was also derived in (Waldspurger et al., 2015) using the classical lifting argument for non-convex quadratic programs developed in (Shor, 1987; Lovász and Schrijver, 1991). This relaxation is written

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(UM) \\ \text{subject to} & \mathbf{diag}(U) = 1,\ U \succeq 0, \end{array} \qquad \text{(PhaseCut)}$$

which is a semidefinite program (SDP) in the matrix $U \in \mathbf{H}_n$. This problem has a structure similar to the classical MAXCUT relaxation and instances of reasonable size can be solved using specialized implementations of interior point methods designed for that problem (Helmberg et al., 1996). Larger instances are solved in (Waldspurger et al., 2015) using the block-coordinate descent algorithm BlockPhaseCut.

Ultimately, algorithmic choices heavily depend on problem structure, and these will be discussed in detail in the section that follows. In particular, we will study how to exploit structural information on the signal (nonnegativity, sparse 2D Fast Fourier Transform, etc.), to solve realistically large instances formed in diffraction imaging applications.

2.3 Imaging problems

In the imaging problems we study here, various illuminations of a single object are performed through randomly coded masks, hence the matrix A is usually formed using a combination

Algorithm 8 Block Coordinate Descent Algorithm for **PhaseCut**.

Input: An initial $U^0 = \mathbf{I}_n$ and $\nu > 0$ (typically small). An integer $N > 1$.

1: **for** $k = 1, \ldots, N$ **do**

2: Pick $i \in [1, n]$.

3: Compute

$$u = U^k_{i^c,i^c} M_{i^c,i} \quad \text{and} \quad \gamma = u^* M_{i^c,i} \qquad \text{(BlockPhaseCut)}$$

4: If $\gamma > 0$, set

$$U^{k+1}_{i^c,i} = U^{k+1*}_{i,i^c} = -\sqrt{\frac{1-\nu}{\gamma}}x$$

else

$$U^{k+1}_{i^c,i} = U^{k+1*}_{i,i^c} = 0.$$

5: **end for**

Output: A matrix $U \succeq 0$ with $\mathbf{diag}(U) = 1$.

of random masks and Fourier transforms, and we have significant structural information on both the signal we seek to reconstruct (regularity, etc.) and the observations (power law decay in frequency domain, etc.). Many of these additional structural hints can be used to speedup numerical operations, convergence and improve phase retrieval performance. The paragraphs that follow explore these points in more detail.

2.3.1 Fourier operators

In practical applications, because of the structure of the linear operator A, we may often reduce numerical complexity, using the Fourier structure of A to speedup the single matrix-vector product in algorithm BlockPhaseCut. We detail the case where A corresponds to a Fourier transform combined with k random masks, writing $I_1, ..., I_k \in \mathbb{C}^p$ the illumination masks. The image by A of some signal $x \in \mathbb{C}^p$ is then

$$Ax = \begin{pmatrix} \mathcal{F}(I_1 \circ x) \\ \vdots \\ \mathcal{F}(I_k \circ x) \end{pmatrix},$$

and the pseudo-inverse of A also has a simple structure, with

$$A^\dagger \begin{pmatrix} y_1 \\ \vdots \\ y_k \end{pmatrix} = \sum_{l=1}^{k} \mathcal{F}^{-1}(y_l) \circ I'_l$$

where I'_l is the dual filter of I_l, which is

$$I'_l = \overline{I}_l / \left(\sum_s |I_s|^2 \right).$$

With the fast Fourier transform, computing the image of a vector by A or $A^\dagger$ only requires $O(kp\log(p))$ floating-point operations. For any $v \in \mathbb{C}^n$, $Mv = \mathbf{diag}(b)(\mathbf{I} - AA^\dagger)\,\mathbf{diag}(b)v$ may then be computed using $O(kp\log p)$ operations instead of $O(k^2p^2)$ for naive matrix-vector multiplications.

In algorithms Greedy and BlockPhaseCut, we also need to extract quickly columns from M without having to store the whole matrix. Extracting the column corresponding to index i in block $l \leq k$ reduces to the computation of $AA^\dagger\delta_{i,l}$ where $\delta_{i,l} \in \mathbb{C}^{kp}$ is the vector whose coordinates are all zero, except the i-th one of l-th block. If we write $\delta_i \in \mathbb{C}^p$ the Dirac in i, the preceding formulas yields

$$AA^\dagger\delta_{i,l} = \begin{pmatrix} \delta_i \star \mathcal{F}(I_1 \circ I_l') \\ \vdots \\ \delta_i \star \mathcal{F}(I_k \circ I_l') \end{pmatrix}.$$

Convolution with δ_i is only a shift and vectors $\mathcal{F}(I_s \circ I_l')$ may be precomputed so this operation is very fast.

2.3.2 Low rank iterates

In instances where exact recovery occurs, the solution to the semidefinite programming relaxation (PhaseCut) has rank one. It is also likely to have low rank in a neighborhood of the optimum. This means that we can often store a compressed version of the iterates U in algorithm BlockPhaseCut in the form of their low rank approximation $U = VV^*$ where $V \in \mathbb{C}^{n\times k}$. Each iteration updates a single row/column of U which corresponds to a rank two update of U, hence updating the SVD means computing a few leading eigenvalues of the matrix $VV^* + LL^*$ where $L \in \mathbb{C}^{n\times 2}$. This update can be performed using Lanczos type algorithms and has complexity $O(kn\log n)$. Compressed storage of U saves memory and also speeds-up the evaluation of the vector matrix product $U_{i^c,i^c}M_{i^c,i}$ which costs $O(nk)$ given a decomposition $U_{i^c,i^c} = VV^*$, instead of $O(n^2)$ using a generic representation of the matrix U. We refer to table 2.3 for experimental comparison between full rank and low rank iterates.

2.3.3 Bounded support

In many inverse problems the signal we are seeking to reconstruct is known to be sparse in some basis and exploiting this structural information explicitly usually improves signal recovery performance. This is for example the basis of compressed sensing where ℓ_1 penalties encourage sparsity and provide recovery guarantees when the true signal is actually sparse.

The situation is a lot simpler in some of the molecular imaging problems we are studying below since the electron density we are trying to recover is often smooth, which means that its Fourier

transform will be sparse, *with known support*. While we lose the phase, we do observe the magnitude of the Fourier coefficients so we can rank them by magnitude. This allows us to considerably reduce the size of the SDP relaxation without losing much reconstruction fidelity, i.e., in many cases we observe that a significant fraction of the coefficients of b are close to zero. From a computational point of view, sparsity in b allows us to solve a truncated semidefinite relaxation (PhaseCut). See Figure 2.1 for an illustration of this phenomenon on the caffeine molecule.

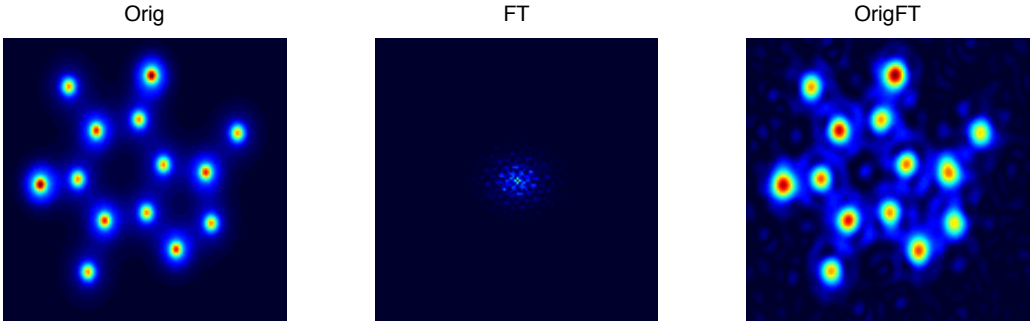


FIGURE 2.1: Electronic density for the caffeine molecule (left), its 2D FFT transform (diffraction pattern, center), the density reconstructed using 2% of the coefficients with largest magnitude in the FFT (right).

Indeed, without loss of generality, we can reorder the observations b such that we approximately have $b = (b_1^T, 0)^T$. Similarly, we note

$$u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \qquad A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}, \qquad A^\dagger = \begin{pmatrix} (A^\dagger)_1 & (A^\dagger)_2 \end{pmatrix}.$$

Using the fact that $b_2 = 0$, the matrix M in the objective of (2.5) can itself be written in blocks, that is

$$M = \begin{pmatrix} M_1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Since $b_2 = 0$, any complex vector with coefficients of magnitude one can be taken for u_2 and the optimization problem (2.5) is equivalent to

$$\begin{array}{ll} \text{minimize} & u_1^* M_1 u_1 \\ \text{subject to} & |u_{1_i}| = 1, \quad i = 1, \dots n, \end{array} \tag{2.6}$$

in the variable $u_1 \in \mathbb{C}^{n_1}$, where the Hermitian matrix

$$M_1 = \mathbf{diag}(b_1)(\mathbf{I} - A_1(A^\dagger)_1)\,\mathbf{diag}(b_1)$$

is positive semidefinite. This problem can in turn be relaxed into a PhaseCut problem which is usually considerably smaller than the original (PhaseCut) problem since M_1 is typically a fraction of the size of M.

2.3.4 Real, positive densities

In some cases, such as imaging experiments where a random binary mask is projected on an object for example, we know that the linear observations are formed as the Fourier transform of a *positive* measure. This introduces additional restrictions on the structure of these observations, which can be written as convex constraints on the phase vector. We detail two different ways of accounting for this positivity assumption.

2.3.4.1 Direct nonnegativity constraints on the density

In the case where the signal is real and nonnegative, (Waldspurger et al., 2015) show that problem (2.3) can modified to specifically account for the fact that the signal is real, by writing it

$$\min_{\substack{u\in\mathbb{C}^n,\,|u_i|=1,\\ x\in\mathbb{R}^p}} \|Ax - \mathbf{diag}(b)u\|_2^2,$$

using the operator $\mathcal{T}(\cdot)$ defined as

$$\mathcal{T}(Z) = \begin{pmatrix} \mathrm{Re}(Z) & -\mathrm{Im}(Z) \\ \mathrm{Im}(Z) & \mathrm{Re}(Z) \end{pmatrix} \tag{2.7}$$

we can rewrite the phase problem on real valued signal as

$$\begin{array}{ll} \text{minimize} & \left\| \mathcal{T}(A) \begin{pmatrix} x \\ 0 \end{pmatrix} - \mathbf{diag} \begin{pmatrix} b \\ b \end{pmatrix} \begin{pmatrix} \mathrm{Re}(u) \\ \mathrm{Im}(u) \end{pmatrix} \right\|_2^2 \\ \text{subject to} & u \in \mathbb{C}^n,\ |u_i| = 1 \\ & x \in \mathbb{R}^p. \end{array}$$

The optimal solution of the inner minimization problem in x is given by $x = A_2^\dagger B_2 v$, where

$$A_2 = \begin{pmatrix} \mathrm{Re}(A) \\ \mathrm{Im}(A) \end{pmatrix}, \quad B_2 = \mathbf{diag} \begin{pmatrix} b \\ b \end{pmatrix}, \quad \text{and} \quad v = \begin{pmatrix} \mathrm{Re}(u) \\ \mathrm{Im}(u) \end{pmatrix}$$

hence the problem is finally rewritten

$$\begin{array}{ll} \text{minimize} & \|(A_2 A_2^\dagger B_2 - B_2)v\|_2^2 \\ \text{subject to} & v_i^2 + v_{n+i}^2 = 1, \quad i = 1, \ldots, n, \end{array}$$

in the variable $v \in \mathbb{R}^{2n}$. This can be relaxed as above by the following problem

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(VM_2) \\ \text{subject to} & V_{ii} + V_{n+i,n+i} = 1, \quad i = 1, \ldots, n, \\ & V \succeq 0, \end{array} \tag{PhaseCutR}$$

which is a semidefinite program in the variable $V \in \mathbf{S}_{2n}$, where

$$M_2 = (A_2 A_2^\dagger B_2 - B_2)^T (A_2 A_2^\dagger B_2 - B_2) = B_2^T (\mathbf{I} - A_2 A_2^\dagger) B_2.$$

Because $x = A_2^\dagger B_2 v$ for real instances, we can add a nonnegativity constraint to this relaxation, using

$$xx^T = (A_2^\dagger B_2) u u^T (A_2^\dagger B_2)^T$$

and the relaxation becomes

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(VM_2) \\ \text{subject to} & (A_2^\dagger B_2) V (A_2^\dagger B_2)^T \geq 0, \\ & V_{ii} + V_{n+i,n+i} = 1, \quad i = 1, \ldots, n, \\ & V \succeq 0, \end{array}$$

which is a semidefinite program in $V \in \mathbf{S}_{2n}$.

2.3.4.2 Bochner's theorem and the Fourier transform of positive measures

Another way to include nonnegativity constraints on the signal, which preserves some of the problem structure, is to use Bochner's theorem. Recall that a function $f : \mathbb{R}^s \mapsto \mathbb{C}$ is *positive semidefinite* if and only if the matrix B with coefficients $B_{ij} = f(x_i - x_j)$ is Hermitian positive semidefinite for any sequence $x_i \in \mathbb{R}^s$. Bochner's theorem then characterizes Fourier transforms of positive measures.

Theorem 2.1. (Bochner) *A function $f : \mathbb{R}^s \mapsto \mathbb{C}$ is positive semidefinite if and only if it is the Fourier transform of a (finite) nonnegative Borel measure.*

Proof. See (Berg et al., 1984) for example. ■

For simplicity, we first illustrate this in dimension one. Suppose that we observe the magnitude of the Fourier transform of a discrete nonnegative signal $x \in \mathbb{R}^p$ so that

$$|\mathcal{F}x| = b$$

with $b \in \mathbb{R}^n$. Our objective now is to reconstruct a phase vector $u \in \mathbb{C}^n$ such that $|u| = 1$ and

$$\mathcal{F}x = \mathbf{diag}(b)u.$$

If we define the Toeplitz matrix

$$B_{ij}(y) = y_{|i-j|+1}, \quad 1 \le j \le i \le p,$$

so that

$$B(y) = \begin{pmatrix} y_1 & y_2^* & & \cdots & & y_n^* \\ y_2 & y_1 & y_2^* & & \cdots & \\ & y_2 & y_1 & y_2^* & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \\ & \cdots & & y_2 & y_1 & y_2^* \\ y_n & & \cdots & & y_2 & y_1 \end{pmatrix}$$

then when $\mathcal{F}x = \mathbf{diag}(b)u$, Bochner's theorem states that $B(\mathbf{diag}(b)u) \succeq 0$ if and only if $x \ge 0$. The contraint $B(\mathbf{diag}(b)u) \succeq 0$ is a linear matrix inequality in u, hence is convex.

Suppose that we observe multiple illuminations and that the k masks $I_1, \ldots, I_k \in \mathbb{R}^{p \times p}$ are also nonnegative (e.g., random coherent illuminations), we have

$$Ax = \begin{pmatrix} \mathcal{F}(I_1 \circ x) \\ \vdots \\ \mathcal{F}(I_k \circ x) \end{pmatrix},$$

and the phase retrieval problem (2.5) for positive signals x is now written

$$\begin{array}{ll} \text{minimize} & u^* M u \\ \text{subject to} & B_j(\mathbf{diag}(b)u) \succeq 0, \quad j = 1, \ldots, k \\ & |u_i| = 1, \quad i = 1, \ldots n, \end{array}$$

where $B_j(y)$ is the matrix $B(y^{(j)})$, where $y^{(j)} \in \mathbb{C}^p$ is the j^{th} subvector of y (one for each of the k masks). We can then adapt the PhaseCut relaxation to incorporate the positivity requirement. In the one dimensional case, using again the classical lifting argument in (Shor, 1987; Lovász and Schrijver, 1991), it becomes

$$\begin{array}{ll} \text{min.} & \mathbf{Tr}(UM) \\ \text{subject to} & \mathbf{diag}(U) = 1,\ u_1 = 1, \\ & B_j(\mathbf{diag}(b)u) \succeq 0, \quad j = 1, \ldots, k \\ & \begin{pmatrix} U & u \\ u^* & 1 \end{pmatrix} \succeq 0 \end{array} \qquad \text{(PhaseCut+)}$$

in the variables $U \in \mathbf{S}_n$ and $u \in \mathbb{C}^n$. The phase vector u is fixed up to an arbitrary global shift, and the additional constraint $u_1 = 1$ allows us to exclude degenerate solutions with $u = 0$. Similar results apply in multiple dimensions, since the 2D Fourier transform is simply computed by applying the 1D Fourier transform first to columns then to rows.

The SDP relaxation PhaseCut+ cannot be solved using block coordinate descent. Without positivity constraints, the relaxation PhaseCutR designed for real signals can be solved efficiently using the algorithm in (Helmberg et al., 1996). The constraint structure in PhaseCutR means that the most expensive step at each iteration of the algorithm in (Helmberg et al., 1996) is computing the inverse of a symmetric matrix of dimension n (or less, exploiting sparsity in b). Sparse instances of the more complex relation PhaseCut+ were solved using SDPT3 (Toh et al., 1999) in what follows.

2.4 Numerical Experiments

We study molecular imaging problems based on electronic densities obtained from the Protein Data Bank (Berman et al., 2002). From a 3D image, we obtain a 2D projection by integrating the third dimension. After normalizing these images, we simulate multiple diffraction observations for each molecule, using several random masks. Here, our masks consist of randomly generated binary filters placed before the sample, but other settings are possible (Candes et al., 2014). Our vector of observations then corresponds to the magnitude of the Fourier transform of the componentwise product of the image and the filter. As in the SPSIM package (Maia, 2013) simulating diffraction imaging experiments, random Poisson noise is added to the observations, modeling sensor and electronic noise. More specifically, the noisy intensity measurements are obtained using the following formula,

$$I = \sqrt{\max\left\{0, \alpha \cdot \text{Poisson}\left(\frac{|Ax|^2}{\alpha}\right)\right\}},$$

where α is the input level of noise, and $\text{Poisson}(\lambda)$ is a random Poisson sample of mean λ. We ensure that all points of the electronic density are illuminated at least once by the random masks (the first mask lets all the signal go through) and call mask “resolution" the number of pixels in a square unit of the mask. For instance masks of resolution 4×4 pixels in a 16×16 pixels image will consist of sixteen square blocks of size 4×4 pixels, each block being either all zeros or all ones.

We present numerical experiments on two molecules from the Protein Data Bank (PDB), namely caffeine and cocaine, with very different structure (properly projected, caffeine is mostly circular, cocaine has a star shape). Images of the caffeine and cocaine molecules at low and high resolutions are presented in Figure 2.2. We first use “high" 128×128 pixels resolutions to

evaluate the sensitivity of PhaseCut to noise and number of masks using the fast BlockPhaseCut algorithm (see Section 2.4.1). We then use a "low" 16×16 pixels image resolution to compare PhaseCut formulations using structural constraints, i.e., complex PhaseCut, real PhaseCutR, and PhaseCut+ (with positivity constraints, see Section 2.4.2) on a large number of random experiments.

FIGURE 2.2: Two molecules, caffeine (left) and cocaine (right), at two resolutions: 16×16 and 128×128.

	Caffeine	Cocaine	Lysozyme
Nb. atoms	14	43	1309
16×16 res.	58 %	40 %	11 %
32×32 res.	44 %	40 %	20 %
64×64 res.	15 %	55 %	14 %
128×128 res.	4 %	55 %	4 %

TABLE 2.1: Percentage of 2D FFT coefficients required to reach $10^{-1.5}$ relative MSE, without oversampling.

	Caffeine	Cocaine	Lysozyme
Nb. atoms	14	43	1309
16×16 res.	48 %	34 %	10 %
32×32 res.	37 %	35 %	17 %
64×64 res.	13 %	48 %	12 %
128×128 res.	4 %	49 %	4 %

TABLE 2.2: Percentage of 2D FFT coefficients required to reach $10^{-1.5}$ relative MSE, with 2x oversampling.

2.4.1 High resolution experiments using BlockPhaseCut

We first compare the results obtained by the Fienup and BlockPhaseCut algorithms while varying the number of masks and the noise level. For the PhaseCut relaxation, in order to deal with the large size of the lifted matrix, we use the low rank approximation described in §2.3.2 to store iterates and exploit sparsity in the magnitude of the observations vector described as described in §2.3.3. Tables 2.1 and 2.2 illustrate the impact of image resolution and oversampling on the

fraction of coefficients required to approximately reconstruct a molecular density up to a given quality threshold, for various molecules. We observe that the sparsity of 2D FFTs increases with resolution and oversampling, but varies from one molecule to another. We then retrieve the phase vector as the first eigenvector in the final low rank approximation, then refine it with the greedy algorithms Greedy or Fienup. Table 2.3 shows the small impact of using low rank iterates instead of full rank iterates in the block coordinate descent algorithm.

rank=1	**rank=2**	**rank=3**	**rank=4**	**rank=5**
5.0±18.5	0.8±0.5	0.7±0.5	0.6±0.4	0.5±0.4

TABLE 2.3: $10^3 \times |MSE(BCD) - MSE(BCDLR)|$. Figures after the sign $\pm$ correspond to standard deviation when varying the random illuminations. We have compared the two algorithms on a 128×128 caffeine image with two random illuminations (filter resolution of 1 pixel), keeping only the 1000 observations with highest magnitude, and adding some small Poisson noise ($\alpha = 10^{-3}$). While there is no guarantee of convergence for the low-rank block coordinate descent algorithm (BCDLR), after 20 cycles both the full rank and low rank algorithms seem to converge to a very close value (up to 10^{-4} accuracy in terms of MSE), even when the number of computed eigenvectors is very small.

2.4.1.1 Parameters

More specifically, in the experiments that follow, the image was of size 128×128, we used a rank of two for the low rank approximation, kept the largest 1000 observations, did 5000 iterations of algorithm Fienup, and 20 cycles of algorithm BlockPhaseCut (one cycle corresponds to optimizing once over all rows/columns of the lifted matrix). The Fourier transform was oversampled by a factor 2. We compared the results of the phase recovery using one to four masks, and three different levels of Poisson noise (no noise, “small" noise, “large" noise). In all settings, all points of the electronic density were illuminated at least once by the random masks (the first mask lets all the signal go through). The noisy (Poisson) intensity measurements were obtained using the formula described above. Experiments were performed on a recent Macbook pro laptop using Matlab for the greedy algorithms and a C implementation of the block coordinate algorithm for PhaseCut. Reported CPU times are in seconds.

2.4.1.2 Results

In most cases both algorithm Fienup and BlockPhaseCut seem to converge to the (global) optimal solution, though Fienup is much faster. In some cases however, such as the experiment with two filters and no noise in Figure 2.3, initializing algorithm Fienup with the solution from BlockPhaseCut significantly outperforms the solution obtained by algorithm Fienup alone, which appears to be stuck in a local minimum. The corresponding MSE values are listed in Table 2.4.

In Figure 2.5 we plot the histogram of MSE for the noiseless case with only two illuminations, using either algorithm Fienup, or BlockPhaseCut followed by greedy refinements, over many random illumination configurations. We observe that in many samples, algorithm Fienup gets stuck in a local optimum, while the SDP always converges to a global optimum.

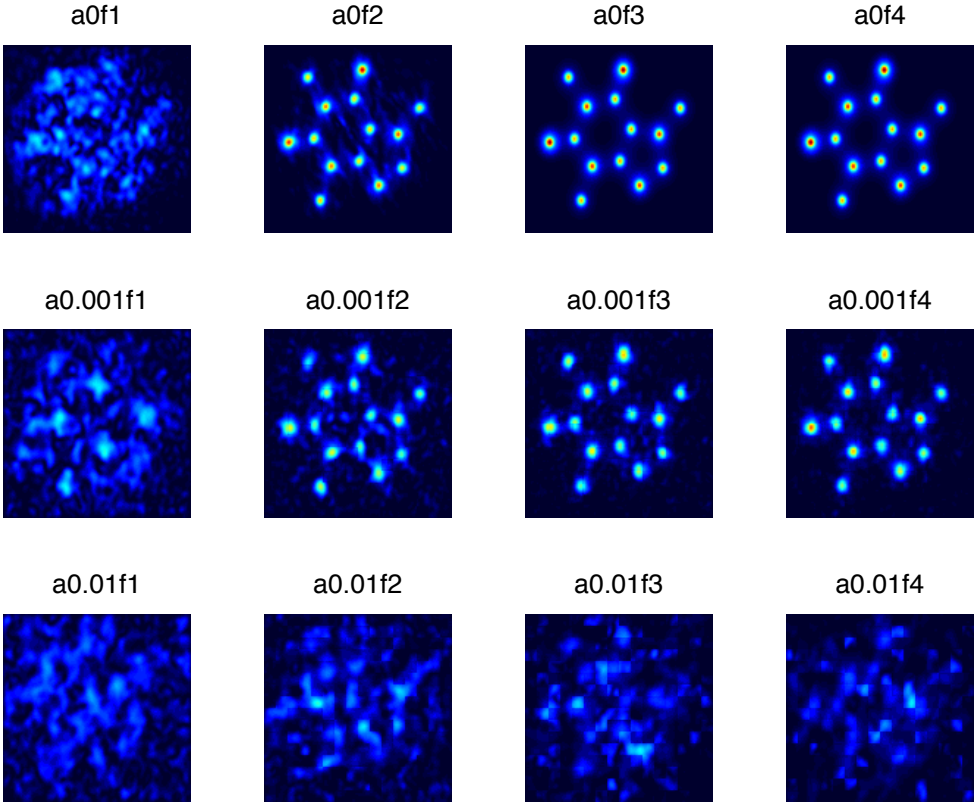


FIGURE 2.3: Solution of the semidefinite relaxation algorithm BlockPhaseCut followed by greedy refinements, for various values of the number of filters and noise level α.

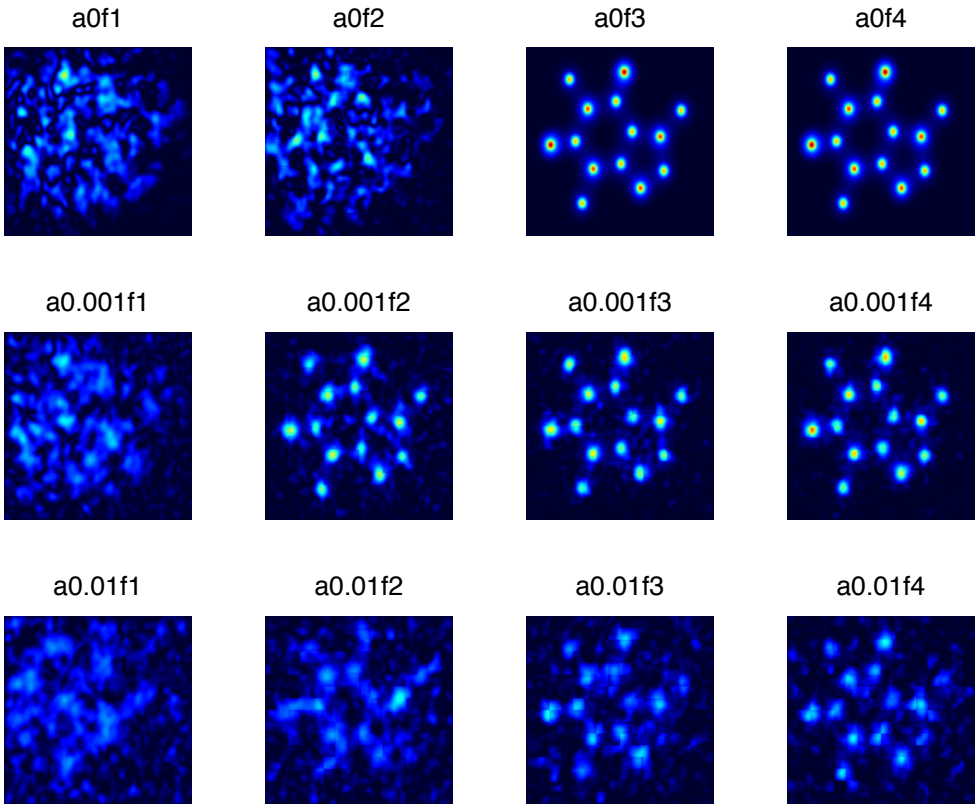


FIGURE 2.4: Solution of the greedy algorithm Fienup, for various values of the number of filters and noise level α.

Nb. masks	α	**SDP MSE**	**SDP refined MSE**	**Fienup MSE**	**SDP time**	**Fienup time**
1.000	0.000	0.023	0.002	0.003	53.802	30.780
2.000	0.000	0.128	0.001	0.004	58.074	53.840
3.000	0.000	0.164	0.000	0.000	58.547	81.332
4.000	0.000	0.177	0.000	0.000	61.135	104.691
1.000	0.001	0.046	0.040	0.042	52.414	26.684
2.000	0.001	0.150	0.241	0.244	55.700	55.277
3.000	0.001	0.183	0.338	0.337	58.372	93.948
4.000	0.001	0.194	0.392	0.392	60.843	111.059
1.000	0.010	0.171	0.168	0.168	53.138	27.648
2.000	0.010	0.320	0.411	0.411	57.659	63.456
3.000	0.010	0.319	0.539	0.540	60.554	100.262
4.000	0.010	0.299	0.599	0.598	63.559	111.435

TABLE 2.4: Performance comparison between algorithms Fienup and BlockPhaseCut for various values of the number of filters and noise level α. CPU times are in seconds.

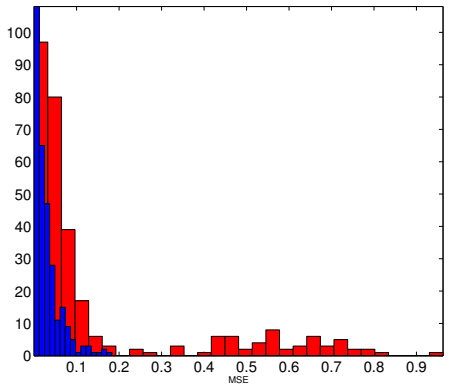


FIGURE 2.5: Histogram of MSE for the noiseless case with only two illuminations, using either algorithm Fienup (red), or BlockPhaseCut (blue) followed by greedy refinements, over many random starts.

2.4.2 Performance of PhaseCut relaxations with respect to number of masks, noise and filter resolution

We now compare PhaseCut formulations with structural constraints, i.e., complex PhaseCut, real PhaseCutR, and PhaseCut+ (with positivity constraints, see Section 2.4.2) on a large number of random experiments formed using "low" 16×16 pixels image resolution.

2.4.2.1 Varying the number of masks

Masks are of resolution 1×1 and no noise is added. As shown in Figures 2.6 and 2.7, PhaseCut, PhaseCutR and PhaseCut+ with Fienup post processing (respectively "SDP + Fienup HIO", "SDP + real + Fienup HIO", "SDP + real + toeplitz + Fienup HIO" and "Fienup HIO" curves on the figure) all outperform Fienup alone. For PhaseCut, in most cases, two to three masks seem enough to exactly recover the phase. Moreover, as expected, PhaseCutR performs a little bit better than PhaseCut, but surprisingly, positivity constraints of PhaseCut+ do not seem to improve the solution of PhaseCutR in these experiments. Finally, as shown in Figures 2.8 and 2.9, oversampling the Fourier transform seems to have a positive impact on the reconstruction. Results on caffeine and cocaine are very similar.

2.4.2.2 Varying mask resolution

Here, two or three masks are used and no noise is added. As shown in Figures 2.10, 2.11, 2.12 and 2.13, we can see that the MSE of reconstructed images increase with the resolution of

masks. Moreover PhaseCutR is more robust to lower mask resolution than PhaseCut. Finally, as expected, with more randomly masked illuminations, we can afford to lower mask resolution.

2.4.2.3 Varying noise levels

Here two masks are used (the minimum), with resolution 1×1. Poisson noise is added (parameterized by α). As shown in Figures 2.14, and 2.15, we can see that PhaseCut and PhaseCutR are stable with regards to noise, i.e., we obtain a linear increase of the log MSE with respect to the log noise.

2.5 Discussion

In this chapter, we have experimented algorithms to solve convex relaxation of the phase retrieval problem for molecular imaging. We have shown that exploiting structural assumptions on the signal and the observations, such as sparsity, smoothness or positivity, can significantly speed-up convergence and improve recovery performance. Extensive molecular imaging experiments were performed using simulated data from the Protein Data Bank (PDB). Candes et al. (2015b) have recently proposed a non-convex algorithm with spectral initialization for phase retrieval that deserves much attention. The iterative structure of their method makes it much more scalable than SDP relaxations, while preserving the same theoretical guarantees on the number of measurements needed for recovery. We also refer to the recent review of phase retrieval in optical imaging by Shechtman et al. (2014) for more background. Ongoing work with Matthew Seaberg and Alexandre d'Aspremont (ENS Paris & SLAC) tries to reproduce experiments on molecular imaging in a real physical setting (no simulations). It will be very interesting to see which algorithms perform best in practice.

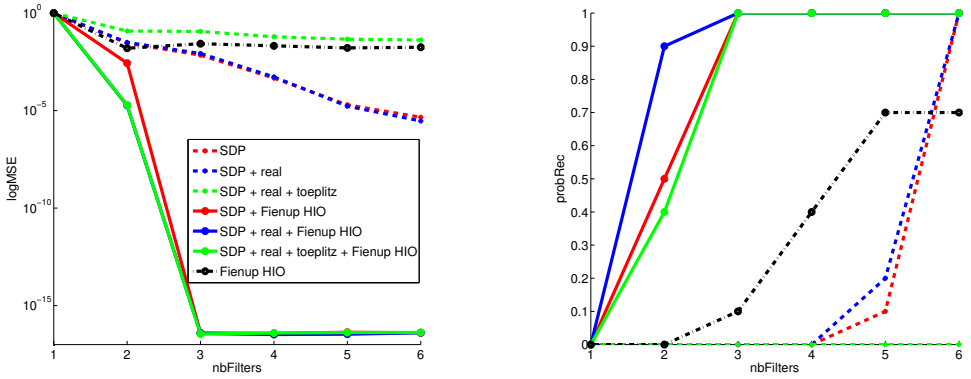


FIGURE 2.6: 16×16 caffeine image. No oversampling. *Left:* MSE (relative to $\|b\|$) vs. number of random masks. *Right:* Probability of recovering molecular density ($MSE < 10^{-4}$) vs. number of random masks.

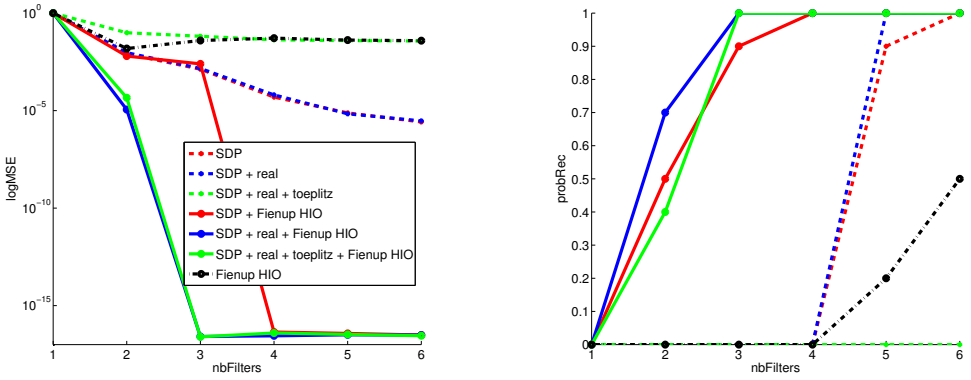


FIGURE 2.7: 16×16 cocaine image. No oversampling. *Left:* MSE (relative to $\|b\|$) vs. number of random masks. *Right:* Probability of recovering molecular density ($MSE < 10^{-4}$) vs. number of random masks.

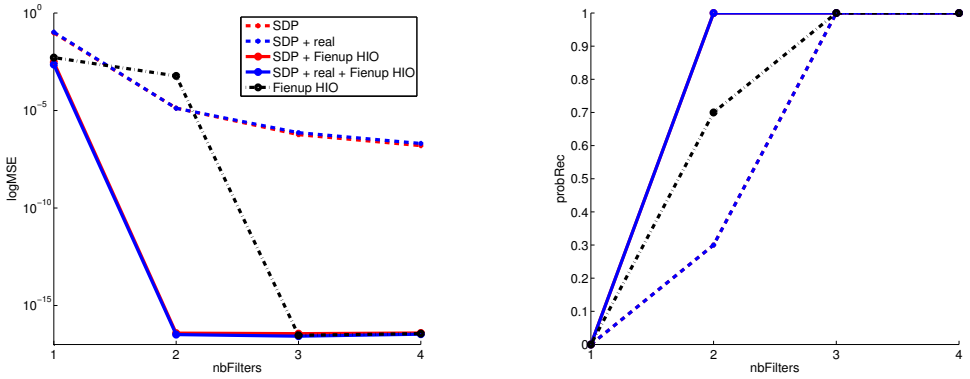


FIGURE 2.8: 16×16 caffeine image. 2x oversampling. *Left:* MSE vs. number of random masks. *Right:* Probability of recovering molecular density ($MSE < 10^{-4}$) vs. number of random masks.

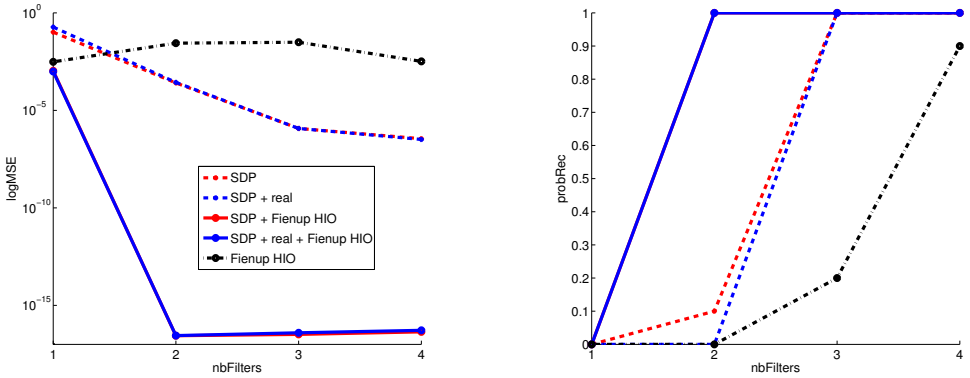


FIGURE 2.9: 16×16 cocaine image. 2x oversampling. *Left:* MSE vs. number of random masks. *Right:* Probability of recovering molecular density ($MSE < 10^{-4}$) vs. number of random masks.

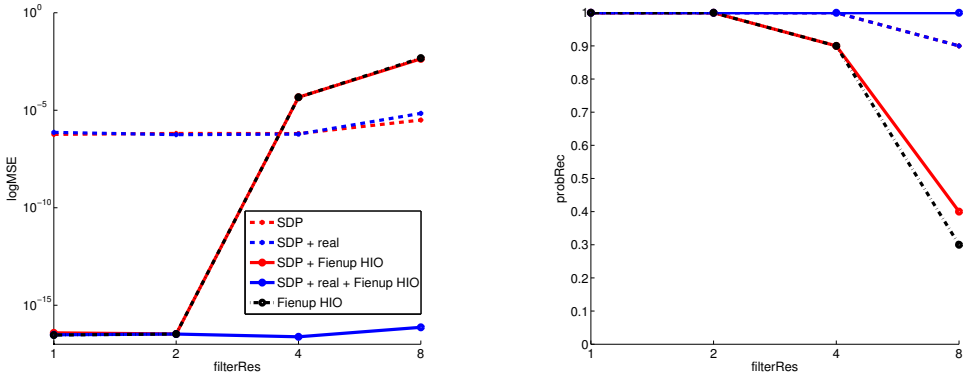


FIGURE 2.10: 16×16 caffeine image. Mask resolution (1x1 to 8x8 pixels). *Left:* MSE vs. mask resolution. (2x oversampling, no noise, 3 masks). *Right:* Probability of recovering molecular density ($MSE < 10^{-4}$) vs. number of random masks.

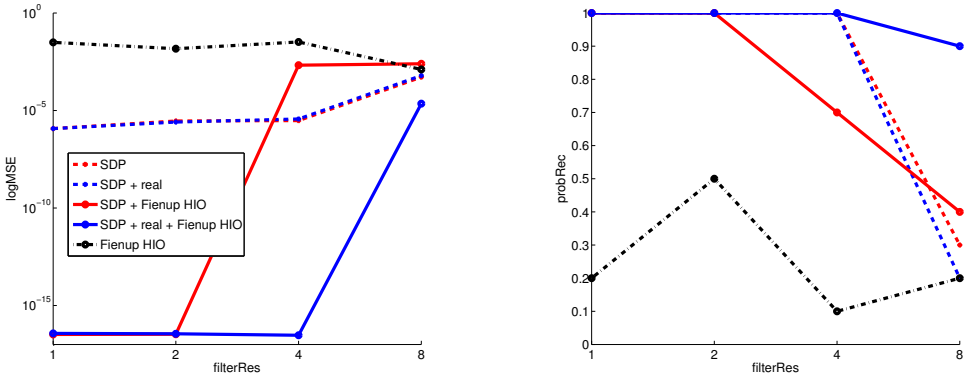


FIGURE 2.11: 16×16 cocaine image. Mask resolution (1x1 to 8x8 pixels). *Left:* MSE vs. mask resolution. (2x oversampling, no noise, 3 masks). *Right:* Probability of recovering molecular density ($MSE < 10^{-4}$) vs. number of random masks.

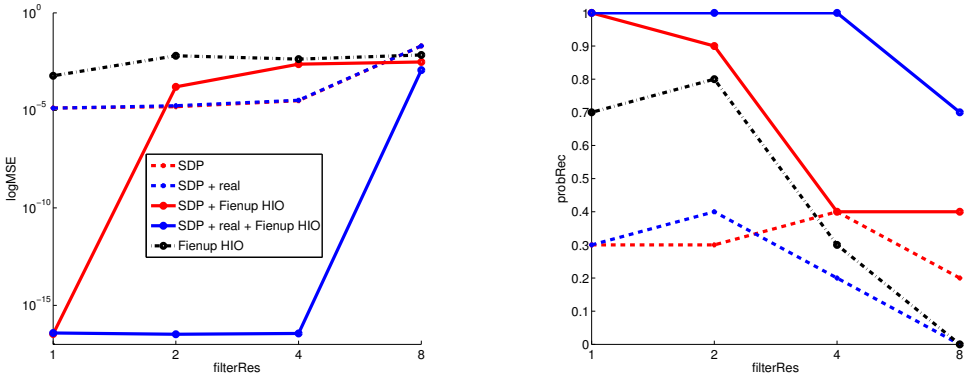


FIGURE 2.12: 16×16 caffeine image. Mask resolution (1x1 to 8x8 pixels). *Left:* MSE vs. mask resolution. (2x oversampling, no noise, 2 masks). *Right:* Probability of recovering molecular density ($MSE < 10^{-4}$) vs. number of random masks.

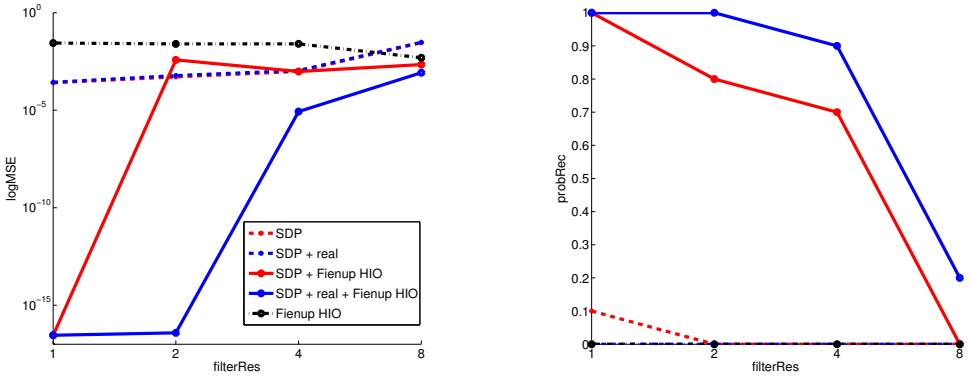


FIGURE 2.13: 16×16 cocaine image. Mask resolution (1x1 to 8x8 pixels). *Left:* MSE vs. mask resolution. (2x oversampling, no noise, 2 masks). *Right:* Probability of recovering molecular density ($MSE < 10^{-4}$) vs. number of random masks.

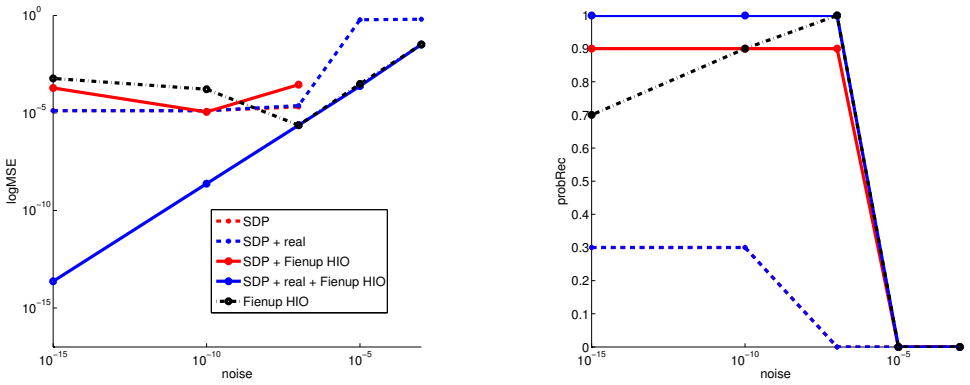


FIGURE 2.14: 16×16 caffeine image. Noise. *Left:* MSE vs. noise level α (2x oversampling, 2 masks). *Right:* Probability of recovering molecular density ($MSE < 10^{-4}$) vs. number of random masks.

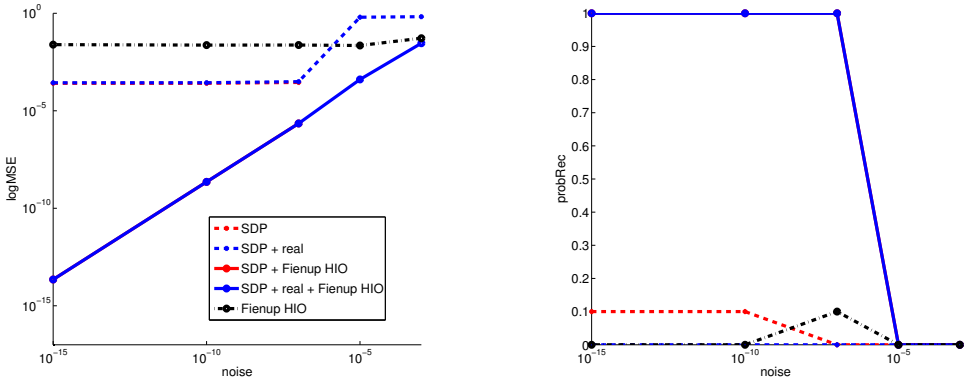


FIGURE 2.15: 16×16 cocaine image. Noise. *Left:* MSE vs. noise level α (2x oversampling, 2 masks). *Right:* Probability of recovering molecular density ($MSE < 10^{-4}$) vs. number of random masks.

2.6 User guide

We provide here the instructions to artificially recover the image of a molecule from the Protein Data Bank using PhaseCutToolbox (download at www.di.ens.fr/~aspremon). This example is entirely reproduced with comments in the script testPhaseCut.m.

2.6.1 Installation

Our toolbox works on all recent versions of MATLAB on Mac OS X, and on MATLAB versions anterior to 2008 on Linux (there might be conflicts with Arpack library for ulterior versions, when using BlockPhaseCut). Installation only requires to put the toolbox folder and subdirectories on the Matlab path. Use for instance the command:

```
>> addpath(genpath('MYPATH/PhaseCutToolbox'));
```

where MYPATH is the directory where you have copied the toolbox.

2.6.2 Generate the diffraction pattern of a molecule

Suppose we work with the caffeine molecule, on an image of resolution 128×128 pixels. We set the corresponding input variables.

```
>> nameMol='caffeine.pdb';
>> N = 128 ;
```

Now, we set the parameters of the masks. The number of masks (also called filters or illuminations) is set to 2. Moreover we set the filter resolution to 1. The filter resolution corresponds to the square root of the number of pixels in each block of the binary filter. The filter resolution must divide `N` (the square root of the number of pixels in the image).

```
>> filterRes = 1 ;
>> nb_filters=2;
```

Since the filters are generated randomly, we set the seed of the uniform random generator to 1 in order to get reproducible experiments. Note that the quality of the phase retrieval may depend on the shape of the generated masks, especially when using only 2 or 3 filters.

```
>> rand('seed',1);
```

Now we can generate an image, 2 masks and their corresponding diffraction patterns. We set the level of noise on the observations to zero here (i.e., no noise). α is the level of Poisson noise, and β is the level of Gaussian noise.

```
>> alpha=0;
>> beta=0;
```

We set the oversampling parameter for the Fourier transform to 2.

```
>> OSF = 2;
```

The total number of observations, i.e., the size of the vector b is

```
>> nbObs=N*N*OSF*OSF*nb_filters;
```

Suppose that we want to use only the first largest one thousand observations in PhaseCut, we set

```
>> nbObsKept=1000;
```

Note that the number of observations that is sufficient to get close to the optimal solution depends on the size of the data `N` and the sparsity of the vector `b`. From a more practical point of view, the larger `nbObsKept`, the more time intensive the optimization. Therefore, for a quick test we recommend setting `nbObsKept` to a few thousands, then increasing it if the results are not satisfying.

Finally we call the function `genData` which is going to generate both the image `x` we want to recover, `filters`, and observations `b`. `bs` corresponds to the thousand largest observations, `xs` is the image recovered with the true phase but using only `bs`. `idx_bs` is the logical indicator vector of `bs` (`bs=b(idx_bs)`). We put `displayFig` to 1 in order to display the filters, the images of the molecule `x` and `xs`, as well as the diffraction patterns (with and without noise).

```
>> displayFig=1;
>> [x,b,filters,bs,xs,idx_bs] = genData(nameMol, nb_filters, ...
 filterRes, N, alpha, beta, OSF, nbObsKept, displayFig);
```

2.6.3 Phase Retrieval using Fienup and/or PhaseCut

Using the data generated in the previous section, we retrieve the phase of the observations vector `b`. Suppose we want to use the SDP relaxation with greedy refinement, we set

```
>> method='SDPRefined';
```

The other choices for `method` are `'Fienup'`, `'Fienup HIO'` and `'SDP'` (no greedy refinement). We set the initial (full) phase vector u to the vector of ones, and the number of iterations for Fienup algorithm to 5000. The number of iterations for Fienup algorithm must be large enough so that the objective function converges to a stationary point. In most cases 5000 iterations seems to be enough.

```
>> param.uInit=ones(nbObs,1);
>> param.nbIterFienup=5000;
```

We also need to choose which algorithm we want to use in order to solve the SDP relaxation. For high resolution images, we recommend to always use the block coordinate descent algorithm with a low rank approximation of the lifted matrix (BCDLR), since interior points methods (when using SDPT3 or Mosek) and block coordinate descent without low rank approximation (BCD) become very slow when the number of observations used is over a few thousands.

```
>> param.SDPsolver='BCDLR';
```

If we had wanted to solve PhaseCutR or PhaseCut+ we would have set

```
>> param.SDPsolver='realSDPT3';
```

or

```
>> param.SDPsolver='ToepSDPT3';
```

We can now set up the parameters for the BCDLR solver.

```
>> param.nbCycles=20;
>> param.r=2;
```

One cycle corresponds to optimizing over all the columns of the lifted matrix. In most cases, it seems that using `nbCycles` between 20 and 40 is enough to get close to the optimum, at least when refining the solution with Fienup algorithm. `r` is the rank for the low rank approximation of the lifted matrix. Similarly it seems that `r` between 2 and 4 gives reasonable results. Note that you can check that the low rank approximation is valid by looking at the maximum ratio between the last and the first eigenvalues throughout all iterations of the BCDLR algorithm. This ratio is outputted as `relax.eigRatio` when calling the function `retrievePhase` (see below). We finally call the function `retrievePhase` in order to solve the SDP relaxation with greedy refinement.

```
>> data.b=b;
>> data.bs=bs;
>> data.idx_bs=idx_bs;
>> data.OSF=OSF;
>> data.filters=filters;
>> [retrievedPhase, objValues, finalObj,relax] = retrievePhase(data,method,param);
```

The function retrievePhase outputs the vector of retrieved phase as `retrievedPhase` and the values of the objective function at each iteration/cycle of the algorithm in `objValues` (add `.Fienup`, `.SDP` `.SDPREfined` to `retrievedPhase` and `objValues` to get the corresponding retrieved phase and objective value). If using the SDP relaxation, the vector `retrievedPhase` is the first eigenvector of the final lifted matrix in PhaseCut. Note that the objective value in Fienup and in the SDP relaxation do not correspond exactly since the lifted matrix may be of rank bigger than one during the iterations of the BCDLR. Therefore we also output `finalObj`, which is the objective value of the phase vector extracted from the lifted matrix (i.e., the vector `retrievedPhase`). The image can now be retrieved using the command

```
>> xRetreived=pseudo_inverse_A(retrievedPhase.SDPRefined.*b,filters,M);
```

Finally you can visualize the results using the following standard Matlab commands, plotting the objective values

```
>> figure(1)
>> subplot(2,1,1);
>> title(method)
>> plot(log10(abs(objValues))); axis tight
```

and displaying images

```
>> subplot(2,3,4)
>> imagesc(abs(x));axis off;
>> subplot(2,3,5)
>> imagesc(abs(xs));axis off;
>> subplot(2,3,6)
>> imagesc(abs(xRetreived)); axis off;
```

2.6.4 Reproducing the experiments of the Chapter

All the numerical experiments of this chapter can be reproduced using the Matlab scripts included in the toolbox directory `Experiments`.

- `phaseTransition_OSF1.m` (evolution of MSE with number of filters, with no oversampling of the Fourier transform, Figures 2.6, 2.7)
- `phaseTransition_OSF2.m` (evolution of MSE with number of filters, with oversampling of the Fourier transform Figures 2.8, 2.9)
- `filterResTransition.m` (evolution of MSE with filter resolution, figures 2.12, 2.13, 2.10, 2.11).
- `noiseTransition.m` (evolution of MSE with noise, Figures 2.14, 2.15)

- `testNoiseNbIllums.m` (test noise vs number of filters, Figures 2.3 and 2.4, and table 2.4)
- `testSeeds.m` (test different seeds to generate filters, Figure 2.5)

Chapter 3

Convex Relaxations for Permutation Problems

Chapter abstract: Seriation seeks to reconstruct a linear order between variables using unsorted, pairwise similarity information. It has direct applications in archeology and shotgun gene sequencing for example. We write seriation as an optimization problem by proving the equivalence between the seriation and combinatorial 2-SUM problems on similarity matrices (2-SUM is a quadratic minimization problem over permutations). The seriation problem can be solved exactly by a spectral algorithm in the noiseless case and we derive several convex relaxations for 2-SUM to improve the robustness of seriation solutions in noisy settings. These convex relaxations also allow us to impose structural constraints on the solution, hence solve semi-supervised seriation problems. We derive new approximation bounds for some of these relaxations and present numerical experiments on archeological data, Markov chains and DNA assembly from shotgun gene sequencing data.

The material of this part is based on the following publications:
F. Fogel, R. Jenatton, F. Bach, A. d'Aspremont, Convex relaxations for permutation problems. In *Advances in Neural Information Processing Systems*, pp. 1016-1024. 2013.
F. Fogel, R. Jenatton, F. Bach, A. d'Aspremont, Convex relaxations for permutation problems. To appear in *SIAM Journal on Matrix Analysis and Applications (SIMAX)*.

3.1 Introduction

We study optimization problems written over the set of permutations. While the relaxation techniques discussed in what follows are applicable to a much more general setting, most of this chapter is centered on the *seriation* problem: we are given a similarity matrix between a set

of n variables and assume that the variables can be ordered along a chain, where the similarity between variables decreases with their distance within this chain. The seriation problem seeks to reconstruct this linear ordering based on unsorted, possibly noisy, pairwise similarity information.

This problem has its roots in archeology (Robinson, 1951) and also has direct applications in e.g., envelope reduction algorithms for sparse linear algebra (Barnard et al., 1995), in identifying interval graphs for scheduling (Fulkerson and Gross, 1965), or in shotgun DNA sequencing where a single strand of genetic material is reconstructed from many cloned shorter reads (i.e., small, fully sequenced sections of DNA) (Garriga et al., 2011; Meidanis et al., 1998). With shotgun gene sequencing applications in mind, many references focused on the *consecutive ones problem* (C1P) which seeks to permute the rows of a binary matrix so that all the ones in each column are contiguous. In particular, Fulkerson and Gross (1965) studied further connections to interval graphs and Kendall (1971) crucially showed that a solution to C1P can be obtained by solving the seriation problem on the squared data matrix. We refer the reader to (Ding and He, 2004; Vuokko, 2010; Liiv, 2010) for a much more complete survey of applications.

On the algorithmic front, the seriation problem was shown to be NP-complete by George and Pothen (1997). Archeological examples are usually small scale and earlier references such as (Robinson, 1951) used greedy techniques to reorder matrices. Similar techniques were, and are still used to reorder genetic data sets. More general ordering problems were studied extensively in operations research, mostly in connection with the quadratic assignment problem (QAP), for which several convex relaxations were derived in e.g., (Lawler, 1963; Zhao et al., 1998). Since a matrix is a permutation matrix if and only if it is both orthogonal and doubly stochastic, much work also focused on producing semidefinite relaxations to orthogonality constraints (Nemirovski, 2007; So, 2011). These programs are convex and can be solved using conic programming solvers, but the relaxations are usually very large and scale poorly. More recently however, Atkins et al. (1998) produced a spectral algorithm that exactly solves the seriation problem in a noiseless setting. They show that for similarity matrices computed from serial variables (for which a total order exists), the ordering of the second eigenvector of the Laplacian (a.k.a. the Fiedler vector) matches that of the variables, in results that are closely connected to those obtained on the interlacing of eigenvectors for Sturm Liouville operators. A lot of work has focused on the minimum linear arrangement problem or 1-SUM, with (Even et al., 2000; Feige, 2000; Blum et al., 2000) and (Rao and Richa, 2005; Feige and Lee, 2007; Charikar et al., 2010) producing semidefinite relaxations with nearly dimension independent approximation ratios. While these relaxations form semidefinite programs that have an exponential number of constraints, they admit a polynomial-time separation oracle and can be solved using the ellipsoid method. The later algorithm being extremely slow, these programs have very little practical impact. Clémençon and Jakubowicz (2010) proposed relaxations on the set of doubly stochastic matrices for the problem of rank aggregation, which also seeks to find a global ordering. Finally,

seriation is also directly related to the manifold learning problem (Weinberger and Saul, 2006), which seeks to reconstruct a low dimensional manifold based on local metric information. Seriation can be seen as a particular instance of that problem, where the manifold is unidimensional but the similarity information is not metric.

Our contribution here is twofold. First, we explicitly write seriation as an optimization problem by proving the equivalence between the seriation and combinatorial 2-SUM problems on similarity matrices. 2-SUM, defined in e.g., (George and Pothen, 1997), is a quadratic minimization problem over permutations. Our result shows in particular that 2-SUM is polynomially solvable for matrices coming from serial data. This quadratic problem was mentioned in (Atkins et al., 1998), but no explicit connection was established between combinatorial problems like 2-SUM and seriation. While the contents of this chapter was under review, a recent working paper by (Laurent and Seminaroti, 2015) has extended the results in Propositions 3.11 and 3.12 here to show that the QAP problem $Q(A, B)$ is solved by the spectral seriation algorithm when A is a similarity matrix (satisfying the Robinson assumption detailed below) and B is a Toeplitz dissimilarity matrix (e.g., $B_{ij} = (i-j)^2$ in the 2-SUM problem discussed here).

Second, we derive several new convex relaxations for the seriation problem. Our simplest relaxation is written over the set of doubly stochastic matrices and appears to be more robust to noise than the spectral solution in a number of examples. Perhaps more importantly, it allows us to impose additional structural constraints to solve semi-supervised seriation problems. We also briefly outline a fast algorithm for projecting on the set of doubly stochastic matrices, which is of independent interest. In the Appendix section, we also produce a semidefinite relaxation for the seriation problem using the classical lifting argument in (Shor, 1987; Lovász and Schrijver, 1991) written on a non-convex quadratic program (QP) formulation of the combinatorial 2-SUM problem. Based on randomization arguments in (Nesterov, 1998; d'Aspremont and El Karoui, 2013) for the MaxCut and k-dense-subgraph problems, we show that this relaxation of the set of permutation *matrices* achieves an approximation ratio of $O(\sqrt{n})$. We also recall how several other relaxations of the minimum linear arrangement (MLA) problem, written on permutation *vectors*, can be adapted to get nearly dimension independent $O(\sqrt{\log n})$ approximation ratios by forming (exponentially large but tractable) semidefinite programs. While these results are of limited practical impact because of the computational cost of the semidefinite programs they form, they do show that certain QAP instances written on Laplacian matrices, such as the seriation problem considered here, are much simpler to approximate than generic QAP problems. They also partially explain the excellent empirical performance of our relaxations in the numerical experiments of Section 3.5.

This chapter is organized as follows. In Section 3.2, we show how to decompose similarity matrices formed in the C1P problem as conic combinations of CUT matrices, i.e., elementary block matrices. This allows us to connect the solutions of the seriation and 2-SUM minimization

problems on these matrices. In Section 3.3 we use these results to write convex relaxations of the seriation problem by relaxing the set of permutation matrices as doubly stochastic matrices in a QP formulation of the 2-SUM minimization problem. Section 3.4 briefly discusses first-order algorithms solving the doubly stochastic relaxation and details in particular a block coordinate descent algorithm for projecting on the set of doubly stochastic matrices. Finally, Section 3.5 describes applications and numerical experiments on archeological data, Markov chains and DNA assembly problems. In the Appendix, we describe larger semidefinite relaxations of the 2-SUM QP and obtain $O(\sqrt{n})$ approximation bounds using randomization arguments. We also detail several direct connections with the minimum linear arrangement problem.

Notation.

We use the notation $\mathcal{P}$ for both the set of permutations of $\{1,\dots,n\}$ and the set of permutation matrices. The notation π will refer to a permuted vector $(1,\dots,n)^T$ while the notation Π (in capital letter) will refer to the corresponding matrix permutation, which is a $\{0,1\}$ matrix such that $\Pi_{ij} = 1$ if and only if $\pi(i) = j$. Moreover y_π is the vector with coefficients $(y_{\pi(1)},\dots,y_{\pi(n)})$ hence $\Pi y = y_\pi$ and $\Pi^T y_\pi = y$. This also means that $A\Pi^T$ is the matrix with coefficients $A_{i\pi(j)}$, and $\Pi A\Pi^T$ is the matrix with coefficients $A_{\pi(i)\pi(j)}$. For a vector $y \in \mathbb{R}^n$, we write $\mathbf{var}(y)$ its variance, with $\mathbf{var}(y) = \sum_{i=1}^n y_i^2/n - (\sum_{i=1}^n y_i/n)^2$, we also write $y_{[u,v]} \in \mathbb{R}^{v-u+1}$ the vector $(y_u,\dots,y_v)^T$. Here, $e_i \in \mathbb{R}^n$ is i-the Euclidean basis vector and $\mathbf{1}$ is the vector of ones. Recall also that the matrix product can be written in terms of outer products, with $AB = \sum_i A_{(i)}B^{(i)}$, with $A_{(i)}$ (resp. $B^{(i)}$) the i-th column (resp. row) of A (resp. B). For a matrix $A \in \mathbb{R}^{m\times n}$, we write $\mathbf{vec}\, A \in \mathbb{R}^{mn}$ the vector formed by stacking up the columns of A. We write $\mathbf{I}$ the identity matrix and $\mathbf{S}_n$ the set of symmetric matrices of dimension n, $\|\cdot\|_F$ denotes the Frobenius norm, $\lambda_i(X)$ the i^{th} eigenvalue (in increasing order) of X and $\|X\|_\infty = \|\mathbf{vec}\, X\|_\infty$.

3.2 Seriation, 2-SUM & consecutive ones

Given a symmetric, binary matrix A, we will focus on variations of the following 2-SUM combinatorial minimization problem, studied in e.g., (George and Pothen, 1997), and written

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^n A_{ij}(\pi(i)-\pi(j))^2 \\ \text{subject to} & \pi \in \mathcal{P}, \end{array} \tag{3.1}$$

where $\mathcal{P}$ is the set of permutations of the vector $(1,\dots,n)^T$. This problem is used for example to reduce the envelope of sparse matrices and is shown in (George and Pothen, 1997, Th. 2.2) to be NP-complete. When A has a specific structure, Atkins et al. (1998) show that a related

matrix ordering problem used for seriation can be solved explicitly by a spectral algorithm. However, the results in Atkins et al. (1998) do not explicitly link spectral ordering and the optimum of (3.1). The main objective of this section is to show the equivalence between the 2-SUM and seriation problems for certain classes of matrices A. In particular, for some instances of A related to seriation and consecutive one problems, we will show below that the spectral ordering directly minimizes the objective of problem (3.1). We first focus on binary matrices, then extend our results to more general unimodal matrices.

Let $A \in \mathbf{S}_n$ and consider the following generalization of the 2-SUM minimization problem

$$\begin{array}{ll} \text{minimize} & f(y_\pi) \triangleq \sum_{i,j=1}^{n} A_{ij}(y_{\pi(i)} - y_{\pi(j)})^2 \\ \text{subject to} & \pi \in \mathcal{P}, \end{array} \tag{3.2}$$

in the permutation variable π, where $y \in \mathbb{R}^n$ is a given weight vector. The classical 2-SUM minimization problem (3.1) is a particular case of problem (3.2) with $y_i = i$. The main point of this section is to show that if A is the permutation of a similarity matrix formed from serial data, then minimizing (3.2) recovers the correct variable ordering. To do this, we simply need to show that when A is correctly ordered, a monotonic vector y solves (3.2), since reordering y is equivalent to reordering A. Our strategy is to first show that we can focus on matrices A that are sums of simple CUT matrices, i.e., symmetric block matrices with a single constant block (see Frieze and Kannan, 1999). We then show that all minimization problems (3.2) written on CUT matrices have a common optimal solution, where y_π is monotonic.

3.2.1 Similarity, C1P & unimodal matrices

We begin by introducing a few definitions on R-matrices (i.e., similarity matrices), C1P and unimodal matrices following (Atkins et al., 1998).

Definition 3.1. (R-matrices) We say that the matrix $A \in \mathbf{S}_n$ is an R-matrix (or Robinson matrix) if and only if it is symmetric and satisfies $A_{i,j} \leq A_{i,j+1}$ and $A_{i+1,j} \leq A_{i,j}$ in the lower triangle, where $1 \leq j < i \leq n$.

Another way to write the R-matrix conditions is to impose $A_{ij} \leq A_{kl}$ if $|i-j| \geq |k-l|$ off-diagonal, i.e., the coefficients of A decrease as we move away from the diagonal (cf. Figure 3.1). In that sense, R-matrices are similarity matrices between variables organized on a *chain*, i.e., where the similarity A_{ij} is monotonically decreasing with the distance between i and j on this chain. We also introduce a few definitions related to the consecutive ones problem (C1P) and its unimodal extension.

Definition 3.2. (P-matrices) We say that the $\{0,1\}$-matrix $A \in \mathbb{R}^{n \times m}$ is a P-matrix (or Petrie matrix) if and only if for each column of A, the ones form a consecutive sequence.

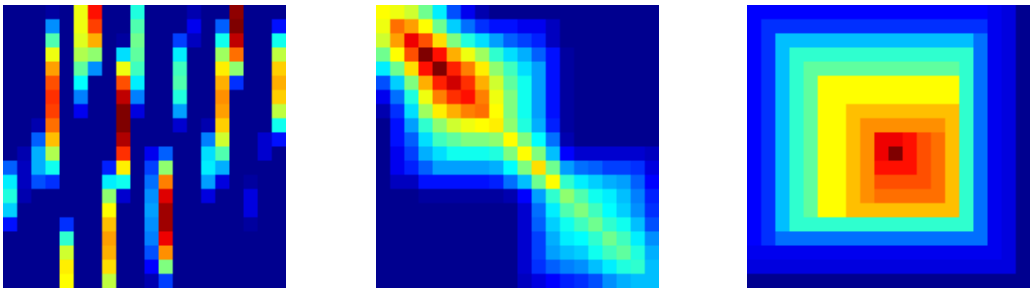


FIGURE 3.1: A sample Q-matrix (see Def. 3.3), which has unimodal columns *(left)*, its "circular square" $A \circ A^T$ (see Def. 3.4) which is an R-matrix *(center)*, and a matrix $a \circ a^T$ where a is a unimodal vector *(right)*.

As in (Atkins et al., 1998), we will say that A is *pre-R* (resp. *pre-P*) if and only if there is a permutation Π such that $\Pi A \Pi^T$ is an R-matrix (resp. ΠA is a P-matrix). Based on Kendall (1971), we also define a generalization of P-matrices called (appropriately enough) Q-matrices, i.e., matrices with unimodal columns.

Definition 3.3. (Q-matrices) We say that a matrix $A \in \mathbb{R}^{n \times m}$ is a Q-matrix if and only if each column of A is unimodal, i.e., the coefficients of each column increase to a maximum, then decrease.

Note that R-matrices are symmetric Q-matrices. We call a matrix A *pre-Q* if and only if there is a permutation Π such that ΠA is a Q-matrix. Next, again based on Kendall (1971), we define the *circular product* of two matrices.

Definition 3.4. Given $A, B^T \in \mathbb{R}^{n \times m}$, their circular product $A \circ B$ is defined as

$$(A \circ B)_{ij} = \sum_{k=1}^{m} \min\{A_{ik}, B_{kj}\} \quad i, j = 1, \ldots, n,$$

note that when A is a symmetric matrix, $A \circ A$ is also symmetric.

Remark that when A, B are $\{0,1\}$ matrices $\min\{A_{ik}, B_{kj}\} = A_{ik}B_{kj}$, so the circular product matches the regular matrix product AB. Similarly, a $\{0,1\}$ matrix with the consecutive one property (C1P) is also unimodal.

3.2.2 Seriation on CUT matrices

We now introduce CUT matrices (named after the CUT decomposition in (Frieze and Kannan, 1999) whose definition is slightly more flexible), and first study the seriation problem on these simple block matrices. The motivation for this definition is that if A is a P, Q or R matrix, then

$A \circ A^T$ can we decomposed as a sum of CUT matrices. This is illustrated in Figure 3.1 and means that we can start by studying problem (3.2) on CUT matrices.

Definition 3.5. For $u, v \in [1, n]$, we call $CUT(u, v)$ the matrix such that

$$CUT(u,v)_{ij} = \begin{cases} 1 & \text{if } u \leq i \leq v \text{ and } u \leq j \leq v \\ 0 & \text{otherwise,} \end{cases}$$

i.e., $CUT(u, v)$ is symmetric, block diagonal and has one square block equal to one.

We first show that the objective of (3.2) has a natural interpretation when A is a CUT matrix, as the variance of a subset of y under a uniform probability measure.

Lemma 3.6. *Suppose $A = CUT(u, v)$, then*

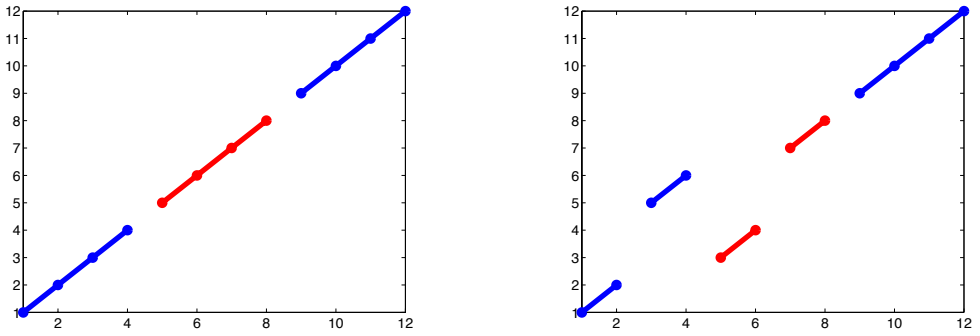
$$f(y) = \sum_{i,j=1}^{n} A_{ij}(y_i - y_j)^2 = (v - u + 1)^2 \, \mathbf{var}(y_{[u,v]}).$$

Proof. We can write $\sum_{ij} A_{ij}(y_i - y_j)^2 = y^T L_A y$ where $L_A = \mathbf{diag}(A\mathbf{1}) - A$ is the Laplacian of A, which is a block matrix with a single nonzero block equal to $(v - u + 1)\delta_{\{i=j\}} - 1$ for $u \leq i, j \leq v$. ■

This last lemma shows that solving the seriation problem (3.2) for CUT matrices amounts to finding a subset of y of size $(u - v + 1)$ with minimum variance. This is the key to all the results that follow. As illustrated in Figure 3.2, for CUT matrices and of course conic combinations of CUT matrices, monotonic sequences have lower variance than sequences where the ordering is broken and the results that follow make this explicit. We now show a simple technical lemma about the impact of switching two coefficients in y on the objective of problem (3.2), when A is a CUT matrix.

Lemma 3.7. *Let $A \in \mathbf{S}_n$, $y \in \mathbb{R}^n$ and $f(\cdot)$ be the objective of problem (3.2). Suppose we switch the values of y_j and y_{j+1} calling the new vector z, we have*

$$f(y) - f(z) = 4 \sum_{\substack{i=1 \\ i \neq j,\, i \neq j+1}}^{n} \left(\frac{y_j + y_{j+1}}{2} - y_i \right) (y_{j+1} - y_j)(A_{ij+1} - A_{ij}).$$



$\mathbf{var}(y_{[5,8]}) = \mathbf{1.6}$ $\mathbf{var}(y_{[5,8]}) = \mathbf{5.6}$

FIGURE 3.2: Objective values of 2-SUM problem (3.2) when $A = CUT(5, 8)$ and $y_i = i$, $i = 1, \ldots, 12$. We plot the permuted values $y_{\pi(i)}$ against i, linking consecutive values of y both inside and outside the interval $[5, 8]$. The solution on the left, where the values of $y_{[5,8]}$ are consecutive, has $\mathbf{var}(y_{[5,8]}) = 1.6$ while $\mathbf{var}(y_{[5,8]}) = 5.6$ on the right, where there is a gap between y_6 and y_7. Minimizing the 2-SUM objective for CUT matrices, i.e., the variance of a subset of the coefficients of y, tends to pull the coefficients in this subset together.

Proof. Because A is symmetric, we have

$$\begin{aligned}(f(y) - f(z))/2 &= \sum_{i\neq j,\, i\neq j+1} A_{ij}(y_i - y_j)^2 + \sum_{i\neq j,\, i\neq j+1} A_{ij+1}(y_i - y_{j+1})^2 \\ &\quad - \sum_{i\neq j,\, i\neq j+1} A_{ij}(y_i - y_{j+1})^2 - \sum_{i\neq j,\, i\neq j+1} A_{ij+1}(y_i - y_j)^2 \\ &= \sum_{i\neq j,\, i\neq j+1} 2A_{ij}(y_j - y_{j+1}) \left(\frac{y_j + y_{j+1}}{2} - y_i\right) \\ &\quad + \sum_{i\neq j,\, i\neq j+1} 2A_{ij+1}(y_{j+1} - y_j) \left(\frac{y_j + y_{j+1}}{2} - y_i\right),\end{aligned}$$

which yields the desired result. ■

The next lemma characterizes optimal solutions of problem (3.2) for CUT matrices and shows that they split the coefficients of y in disjoint intervals.

Lemma 3.8. *Suppose $A = CUT(u, v)$, and write $w = y_\pi$ the optimal solution to (3.2). If we call $I = [u, v]$ and I^c its complement, then*

$$w_j \notin [\min(w_I), \max(w_I)], \quad \textit{for all } j \in I^c,$$

in other words, the coefficients in w_I and w_{I^c} belong to disjoint intervals.

Proof. Without loss of generality, we can assume that the coefficients of w_I are sorted in increasing order. By contradiction, suppose that there is a w_j such that $j \in I^c$ and $w_j \notin [w_u, w_v]$. Suppose also that w is larger than the mean of coefficients inside I, i.e., $w_j \geq \sum_{i=u+1}^{v} w_i/(v - u)$.

This, combined with our assumption that $w_j \leq w_v$ and Lemma 3.7 means that switching the values of w_j and w_v will decrease the objective by

$$4\sum_{i=u}^{v-1}\left(\frac{w_j + w_v}{2} - y_i\right)(w_v - w_j)$$

which is positive by our assumptions on w_j and the mean which contradicts optimality. A symmetric result holds if w_j is smaller than the mean. ■

This last lemma shows that when A is a CUT matrix, then the monotonic vector $y_i = ai + b$, for $a, b \in \mathbb{R}$ and $i = 1, \dots, n$, is always an optimal solution to the 2-SUM problem (3.2), since all subvectors of y of a given size have the same variance. This means that, when y is a permutation of $y_i = ai + b$, all minimization problems (3.2) written on CUT matrices have *a common optimal solution*, where y_π is monotonic.

3.2.3 Ordering P, Q & R matrices

Having showed that all 2-SUM problems (3.2) written on CUT matrices share a common monotonic solution, this section now shows how to decompose the square of P, Q and R-matrices as a sum of CUT matrices, then links the reordering of a matrix with that of its square $A \circ A^T$. We will first show a technical lemma proving that if A is a Q-matrix, then $A \circ A^T$ is a conic combination of CUT matrices. The CUT decomposition of P and R-matrices will then naturally follow, since P-matrices are just $\{0, 1\}$ Q-matrices, and R-matrices are symmetric Q-matrices.

Lemma 3.9. *Suppose $A \in \mathbb{R}^{n\times m}$ is a Q-matrix, then $A \circ A^T$ is a conic combination of CUT matrices.*

Proof. Suppose, $a \in \mathbb{R}^n$ is a unimodal vector, let us show that the matrix $M = a \circ a^T$ with coefficients $M_{ij} = \min\{a_i, a_j\}$ is a conic combination of CUT matrices. Let $I = \{j : j = \operatorname{argmax}_i a_i\}$, then I is an index interval $[I_{\min}, I_{\max}]$ because a is unimodal. Call $\bar{a} = \max_i a_i$ and $b = \max_{i\in I^c} a_i$ (with $b = 0$ when $I^c = \emptyset$), the deflated matrix

$$M^- = M - (\bar{a} - b)\, CUT(I_{\min}, I_{\max})$$

can be written $M^- = a^- \circ (a^-)^T$ with

$$a^- = a - (\bar{a} - b)v$$

where $v_i = 1$ if and only if $i \in I$. By construction $|\operatorname{argmax} M^-| > |I|$, i.e., the size of $\operatorname{argmax} M$ increases by at least one, so this deflation procedure ends after at most n iterations.

This shows that $a \circ a^T$ is a conic combination of CUT matrices when a is unimodal. Now, we have $(A \circ A^T)_{ij} = \sum_{k=1}^{n} w_k \min\{A_{ik}, A_{jk}\}$, so $A \circ A^T$ is a sum of n matrices of the form $\min\{A_{ik}, A_{jk}\}$ where each column is unimodal, hence the desired result. ■

This last result also shows that, when A is a Q matrix, $A \circ A^T$ is a R-matrix as a sum of CUT matrices, which is illustrated in Figure 3.1. We now recall the central result in (Kendall, 1971, Th. 1) showing that for Q-matrices, reordering $A \circ A^T$ also reorders A.

Theorem 3.10. (Kendall, 1971, Th. 1) *Suppose $A \in \mathbb{R}^{n \times m}$ is pre-Q, then ΠA is a Q-matrix if and only if $\Pi(A \circ A^T)\Pi^T$ is a R-matrix.*

We use these last results to show that at least for some vectors y, if C is a Q-matrix then the 2-SUM problem (3.2) written on $A = C \circ C^T$ has a monotonic solution y_π.

Proposition 3.11. *Suppose $C \in \mathbb{R}^{n \times m}$ is a pre-Q matrix and $y_i = ai + b$ for $i = 1, \ldots, n$ and $a, b \in \mathbb{R}$ with $a \neq 0$. Let $A = C \circ C^T$, if Π is such that $\Pi A \Pi^T$ is an R-matrix, then the corresponding permutation π solves the combinatorial minimization problem* (3.2).

Proof. If $C \in \mathbb{R}^{n \times m}$ is pre-Q, then Lemma 3.9 and Theorem 3.10 show that there is a permutation Π such that $\Pi(C \circ C^T)\Pi^T$ is a sum of CUT matrices (hence a R-matrix). Now all monotonic subsets of y of a given length have the same variance, hence Lemmas 3.6 and 3.8 show that π solves problem (3.2). ■

We now show that when the R-constraints are strict, the converse is also true, i.e., for matrices that are the square of Q-matrices, if y_π solves the 2-SUM problem (3.2), then π makes A an R-matrix. In the next section, we will use this result to reorder *pre-R* matrices (with noise and additional structural constraints) by solving convex relaxations to the 2-SUM problem.

Proposition 3.12. *Suppose A is a* pre-R *matrix that can be written as $A = C \circ C^T$, where $C \in \mathbb{R}^{n \times m}$ is a pre-Q matrix, $y_i = ai + b$ for $i = 1, \ldots, n$ and $a, b \in \mathbb{R}$ with $a \neq 0$. Suppose moreover that A has strict R-constraints, i.e., the rows/columns of A are strictly unimodal after reordering. If the permutation π solves the 2-SUM problem* (3.2)*, then the corresponding permutation matrix Π is such that $\Pi A \Pi^T$ is an R-matrix.*

Proof. We can assume that A is a R-matrix without loss of generality. We will show that the identity is optimal for 2-SUM and that it is the unique such solution, hence solving 2-SUM solves seriation. Lemma 3.9 shows that A is a conic combination of CUT matrices. Moreover, by Proposition 3.11 the identity matrix solves problem (3.2). Following the proof of Proposition 3.11, the identity matrix is also optimal for each seriation subproblem on the CUT matrices of A.

Now remark that since the R-constraints are strict on the first column of A, there must be $n-2$ CUT matrices of the form $A_i = CUT(1,i)$ for $i = 2,\dots,n-1$ in the decomposition of A (otherwise, there would be some index $k > 1$ for which $A_{1k} = A_{1k+1}$ which would contradict our strict unimodal assumption). Following the previous remarks, the identity matrix is optimal for all the seriation subproblems in A_i, which means that the variance of all the corresponding subvectors of y_π, i.e., $(y_{\pi(1)}, y_{\pi(2)})$, $(y_{\pi(1)}, y_{\pi(2)}, y_{\pi(3)})$,..., $(y_{\pi(1)},\dots,y_{\pi(n-1)})$ must be minimized. Since these subvectors of y_π are monotonically embedded, up to a permutation of $y_{\pi(1)}$ and $y_{\pi(2)}$, Lemma 3.8 shows that this can only be achieved for contiguous $y_{\pi(i)}$, that is for π equal to the identity or the reverse permutation. Indeed, to minimize the variance of $(y_{\pi(1)},\dots,y_{\pi(n-1)})$, we have to choose $\pi(n) = n$ or $\pi(n) = 1$. Then to minimize the variance of $(y_{\pi(1)},\dots,y_{\pi(n-2)})$, we have to choose respectively $\pi(n-1) = n-1$ or $\pi(n-1) = 2$. Thus we get by induction respectively $\pi(i) = i$ or $\pi(i) = n-i+1$ for $i = 3,\dots,n$. Finally, there are only two permutations left for $y_{\pi(1)}$ and $y_{\pi(2)}$. Since $A_{31} < A_{32}$, we have to choose $(y_{\pi(3)} - y_{\pi(1)})^2 > (y_{\pi(3)} - y_{\pi(2)})^2$, and the remaining ambiguity on the order of $y_{\pi(1)}$ and $y_{\pi(2)}$ is removed. ■

These results shows that if A is *pre-R* and can be written $A = C \circ C^T$ with C pre-Q, then the permutation that makes A an R-matrix also solves the 2-SUM problem (3.2). Conversely, when A is *pre-R* (strictly), the permutation that solves (3.2) reorders A as a R-matrix. Since Atkins et al. (1998) show that sorting the Fiedler vector also orders A as an R-matrix, Proposition 3.11 gives a polynomial time solution to the 2-SUM problem (3.2) when A is *pre-R* with $A = C \circ C^T$ for some pre-Q matrix C. Note that the strict monotonicity constraints on the R-matrix can be somewhat relaxed (we only need one strictly monotonic column plus two more constraints), but requiring strict monotonicity everywhere simplifies the argument.

3.3 Convex relaxations

In the sections that follow, we will use the combinatorial results derived above to produce convex relaxations of optimization problems written over the set of permutation matrices. We mostly focus on the 2-SUM problem in (3.2), however many of the results below can be directly adapted to other objective functions. We detail several convex approximations, some new, some taken from the computer science literature, ranked by increasing numerical complexity. Without loss of generality, we always assume that the weight matrix A is nonnegative (if A has negative entries, it can be shifted to become nonnegative, with no impact on the permutation problem). The nonnegativity assumption is in any case natural since A represents a similarity matrix in the seriation problem.

3.3.1 Spectral ordering

We first recall classical definitions from spectral clustering and briefly survey the spectral ordering results in (Atkins et al., 1998) in the noiseless setting.

Definition 3.13. The Fiedler value of a symmetric, nonnegative matrix A is the smallest nonzero eigenvalue of its Laplacian $L_A = \mathbf{diag}(A\mathbf{1}) - A$. The corresponding eigenvector is called Fiedler vector and is the optimal solution to

$$\begin{array}{ll} \text{minimize} & y^T L_A y \\ \text{subject to} & y^T \mathbf{1} = 0, \|y\|_2 = 1. \end{array} \tag{3.3}$$

in the variable $y \in \mathbb{R}^n$.

We now recall the main result from (Atkins et al., 1998) which shows how to reorder *pre-R* matrices in a noise free setting.

Proposition 3.14. (Atkins et al., 1998, Th. 3.3) *Suppose $A \in \mathbf{S}_n$ is a* pre-R*-matrix, with a simple Fiedler value whose Fiedler vector v has no repeated values. Suppose that Π is a permutation matrix such that the permuted Fielder vector Πv is strictly monotonic, then $\Pi A \Pi^T$ is an R-matrix.*

We now extend the result of Proposition 3.11 to the case where the weights y are given by the Fiedler vector.

Proposition 3.15. *Suppose $A \in \mathbf{S}^{n\times n}$ is a R-matrix and y is its Fiedler vector. Then the identity permutation solves the 2-SUM problem* (3.2).

Proof. The combinatorial problem (3.2) can be rewritten

$$\begin{array}{ll} \text{minimize} & y^T \Pi^T L_A \Pi y \\ \text{subject to} & \Pi \in \mathcal{P}, \end{array}$$

which is also equivalent to

$$\begin{array}{ll} \text{minimize} & z^T L_A z \\ \text{subject to} & z^T \mathbf{1} = 0, \|z\|_2 = 1, z = \Pi y, \Pi \in \mathcal{P}, \end{array}$$

since y is the Fiedler vector of A. By dropping the constraints $z = \Pi y, \Pi \in \mathcal{P}$, we can relax the last problem into (3.3), whose solution is the Fiedler vector of A. Note that the optimal value of problem (3.2) is thus an upper bound on that of its relaxation (3.3), i.e., the Fiedler value of A. This upper bound is attained by the Fiedler vector, i.e., the optimum of (3.3), therefore the identity matrix is an optimal solution to (3.2). ■

Using the fact that the Fiedler vector of a R-matrix is monotonic (Atkins et al., 1998, Th. 3.2), the next corollary immediately follows.

Corollary 3.16. *If A is a* pre-R *matrix such that $\Pi^T A \Pi$ is a R-matrix, then π is an optimal solution to problem* (3.2) *when y is the Fiedler vector of A sorted in increasing order.*

The results in (Atkins et al., 1998) thus provide a polynomial time solution to the R-matrix ordering problem in a noise free setting (extremal eigenvalues of dense matrices can be computed by randomized polynomial time algorithms with complexity $O(n^2 \log n)$ (Kuczynski and Wozniakowski, 1992)). While Atkins et al. (1998) also show how to handle cases where the Fiedler vector is degenerate, these scenarios are highly unlikely to arise in settings where observations on A are noisy and we refer the reader to (Atkins et al., 1998, §4) for details.

3.3.2 QP relaxation

In most applications, A is typically noisy and the *pre-R* assumption no longer holds. The spectral solution is stable when the magnitude of the noise remains within the spectral gap (i.e., in a perturbative regime (Stewart and Sun, 1990)). Beyond that, while the Fiedler vector of A can still be used as a heuristic to find an approximate solution to (3.2), there is no guarantee that it will be optimal.

The results in Section 3.2 made the connection between the spectral ordering in (Atkins et al., 1998) and the 2-SUM problem (3.2). In what follows, we will use convex relaxations to (3.2) to solve matrix ordering problems in a noisy setting. We also show in §3.3.2.3 how to incorporate a priori knowledge on the true ordering in the formulation of the optimization problem to solve semi-supervised seriation problems. Numerical experiments in Section 3.5 show that semi-supervised seriation solutions are sometimes significantly more robust to noise than the spectral solutions ordered from the Fiedler vector.

3.3.2.1 Permutations and doubly stochastic matrices

We write $\mathcal{D}_n$ the set of doubly stochastic matrices, i.e., $\mathcal{D}_n = \{X \in \mathbb{R}^{n \times n} : X \geqslant 0, X\mathbf{1} = \mathbf{1}, X^T\mathbf{1} = \mathbf{1}\}$. Note that $\mathcal{D}_n$ is convex and polyhedral. Classical results show that the set of doubly stochastic matrices is the convex hull of the set of permutation matrices. We also have $\mathcal{P} = \mathcal{D} \cap \mathcal{O}$, i.e., a matrix is a permutation matrix if and only if it is both doubly stochastic and orthogonal. The fact that $L_A \succeq 0$ means that we can directly write a convex relaxation to the combinatorial problem (3.2) by replacing $\mathcal{P}$ with its convex hull $\mathcal{D}_n$, to get

$$\begin{array}{ll} \text{minimize} & g^T \Pi^T L_A \Pi g \\ \text{subject to} & \Pi \in \mathcal{D}_n, \end{array} \tag{3.4}$$

where $g = (1, \ldots, n)$, in the permutation matrix variable $\Pi \in \mathcal{P}$. By symmetry, if a vector Πy minimizes (3.4), then the reverse vector also minimizes (3.4). This often has a significant negative impact on the quality of the relaxation, and we add the linear constraint $e_1^T \Pi g + 1 \leq e_n^T \Pi g$ to break symmetries, which means that we always pick solutions where the first element comes before the last one. Because the Laplacian L_A is positive semidefinite, problem (3.4) is a convex quadratic program in the variable $\Pi \in \mathbb{R}^{n \times n}$ and can be solved efficiently. To produce approximate solutions to problem (3.2), we then generate permutations from the doubly stochastic optimal solution to the relaxation in (3.4) (we will describe an efficient procedure to do so in §3.3.2.4).

The results of Section 3.2 show that the optimal solution to (3.2) also solves the seriation problem in the noiseless setting when the matrix A is of the form $C \circ C^T$ with C a Q-matrix and y is an affine transform of the vector $(1, \ldots, n)$. These results also hold empirically for small perturbations of the vector y and to improve robustness to noisy observations of A, we average several values of the objective of (3.4) over these perturbations, solving

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(Y^T \Pi^T L_A \Pi Y)/p \\ \text{subject to} & e_1^T \Pi g + 1 \leq e_n^T \Pi g, \\ & \Pi \mathbf{1} = \mathbf{1},\ \Pi^T \mathbf{1} = \mathbf{1},\ \Pi \geq 0, \end{array} \tag{3.5}$$

in the variable $\Pi \in \mathbb{R}^{n \times n}$, where $Y \in \mathbb{R}^{n \times p}$ is a matrix whose columns are small perturbations of the vector $g = (1, \ldots, n)^T$. Solving (3.5) is roughly p times faster than individually solving p versions of (3.4).

3.3.2.2 Regularized QP relaxation

In the previous section, we have relaxed the combinatorial problem (3.2) by relaxing the set of permutation matrices into the set of doubly stochastic matrices. As the set of permutation matrices $\mathcal{P}$ is the intersection of the set of doubly stochastic matrices $\mathcal{D}$ and the set of orthogonal matrices $\mathcal{O}$, i.e., $\mathcal{P} = \mathcal{D} \cap \mathcal{O}$ we can add a penalty to the objective of the convex relaxed problem (3.5) to force the solution to get closer to the set of orthogonal matrices. Since a doubly stochastic matrix of Frobenius norm $\sqrt{n}$ is necessarily orthogonal, we would ideally like to solve

$$\begin{array}{ll} \text{minimize} & \frac{1}{p} \mathbf{Tr}(Y^T \Pi^T L_A \Pi Y) - \frac{\mu}{p} \|\Pi\|_F^2 \\ \text{subject to} & e_1^T \Pi g + 1 \leq e_n^T \Pi g, \\ & \Pi \mathbf{1} = \mathbf{1},\ \Pi^T \mathbf{1} = \mathbf{1},\ \Pi \geq 0, \end{array} \tag{3.6}$$

with μ large enough to guarantee that the global solution is indeed a permutation. However, this problem is not convex for any $\mu > 0$ since its Hessian is not positive semi-definite. Note

that the objective of (3.5) can be rewritten as $\mathrm{Vec}(\Pi)^T(YY^T \otimes L_A)\mathrm{Vec}(\Pi)/p$ so the Hessian here is $YY^T \otimes L_A - \mu I \otimes I$ and is never positive semidefinite when $\mu > 0$ since the first eigenvalue of L_A is always zero. Instead, we propose a slightly modified version of (3.6), which has the same objective function up to a constant, and is convex for some values of μ. Recall that the Laplacian matrix L_A is always positive semidefinite with at least one eigenvalue equal to zero corresponding to the eigenvector $\mathbf{1}/\sqrt{n}$ (strictly one if the graph is connected) and let $P = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$.

Proposition 3.17. *The optimization problem*

$$\begin{array}{ll} \text{minimize} & \frac{1}{p}\,\mathbf{Tr}(Y^T\Pi^T L_A \Pi Y) - \frac{\mu}{p}\|P\Pi\|_F^2 \\ \text{subject to} & e_1^T\Pi g + 1 \leq e_n^T \Pi g, \\ & \Pi\mathbf{1} = \mathbf{1},\ \Pi^T\mathbf{1} = \mathbf{1},\ \Pi \geq 0, \end{array} \tag{3.7}$$

is equivalent to problem (3.6), their objectives differ by a constant. Furthermore, when $\mu \leq \lambda_2(L_A)\lambda_1(YY^T)$, this problem is convex.

Proof. Let us first remark that

$$\begin{aligned} \|P\Pi\|_F^2 &= \mathbf{Tr}(\Pi^T P^T P \Pi) = \mathbf{Tr}(\Pi^T P \Pi) \\ &= \mathbf{Tr}(\Pi^T(I - \mathbf{1}\mathbf{1}^T/n)\Pi) = \mathbf{Tr}(\Pi^T\Pi - \mathbf{1}\mathbf{1}^T/n)) \\ &= \mathbf{Tr}(\Pi^T\Pi) - 1 \end{aligned}$$

where we used the fact that P is the (symmetric) projector matrix onto the orthogonal complement of $\mathbf{1}$ and Π is doubly stochastic (so $\Pi\mathbf{1} = \Pi^T\mathbf{1} = \mathbf{1}$). We deduce that problem (3.7) has the same objective function as (3.6) up to a constant. Moreover, it is convex when $\mu \leq \lambda_2(L_A)\lambda_1(YY^T)$ since the Hessian of the objective is given by

$$-\mathcal{A} = \frac{1}{p}\,YY^T \otimes L_A - \frac{\mu}{p}\,\mathbf{I} \otimes P \tag{3.8}$$

and the eigenvalues of $YY^T \otimes L_A$, which are equal to $\lambda_i(L_A)\lambda_j(YY^T)$ for all i, j in $\{1, \ldots, n\}$ are all superior or equal to the eigenvalues of $\mu\mathbf{I} \otimes P$ which are all smaller than μ. ■

To have μ strictly positive, we need YY^T to be definite, which can be achieved w.h.p. by setting p higher than n and sampling independent vectors y. The key motivation for including several monotonic vectors y in the objective of (3.7) is to increase the value of $\lambda_1(YY^T)$. The higher this eigenvalue, the stronger the effect of regularization term in (7), which in turn improves the quality of the solution (all of this being somewhat heuristic of course). The problem of generating good matrices Y with both monotonic columns and high values of $\lambda_1(YY^T)$ is not easy to solve however, hence we use randomization to generate Y.

3.3.2.3 Semi-supervised problems

The QP relaxation above allows us to add structural constraints to the problem. For instance, in archeological applications, one may specify that observation i must appear before observation j, i.e., $\pi(i) < \pi(j)$. In gene sequencing applications, one may constrain the distance between two elements (e.g., mate reads), which would be written $a \leq \pi(i) - \pi(j) \leq b$ and introduce an affine inequality on the variable Π in the QP relaxation of the form $a \leq e_i^T \Pi g - e_j^T \Pi g \leq b$. Linear constraints could also be extracted from a reference gene sequence. More generally, we can rewrite problem (3.7) with n_c additional linear constraints as follows

$$\begin{array}{ll} \text{minimize} & \frac{1}{p}\,\mathbf{Tr}(Y^T\Pi^T L_A \Pi Y) - \frac{\mu}{p}\|P\Pi\|_F^2 \\ \text{subject to} & D^T\Pi g + \delta \leq 0, \\ & \Pi\mathbf{1} = \mathbf{1},\ \Pi^T\mathbf{1} = \mathbf{1},\ \Pi \geq 0, \end{array} \tag{3.9}$$

where D is a matrix of size $n \times (n_c + 1)$ and δ is a vector of size n_c. The first column of D is equal to $e_1 - e_n$ and $\delta_1 = 1$ (to break symmetry).

3.3.2.4 Sampling permutations from doubly stochastic matrices

This procedure is based on the fact that a permutation can be defined from a doubly stochastic matrix S by the order induced on a monotonic vector. A similar argument was used in (Barvinok, 2006) to round orthogonal matrices into permutations. Suppose we generate a *monotonic* random vector v and compute Sv. To each v, we can associate a permutation Π such that ΠSv is monotonically increasing. If S is a permutation matrix, then the permutation Π generated by this procedure will be constant, if S is a doubly stochastic matrix but not a permutation, it might fluctuate. Starting from a solution S to problem (3.7), we can use this procedure to sample many permutation matrices Π and we pick the one with lowest cost $g^T\Pi^T L_A \Pi g$ in the combinatorial problem (3.2). We could also project S on permutations using the Hungarian algorithm, but this proved more costly and less effective in our experiments.

3.4 Algorithms

The convex relaxation in (3.9) is a quadratic program in the variable $\Pi \in \mathbb{R}^{n\times n}$, which has dimension n^2. For reasonable values of n (around a few hundreds), interior point solvers such as MOSEK (Andersen and Andersen, 2000) solve this problem very efficiently (the experiments in this chapter were performed using this library). Furthermore, most *pre-R* matrices formed by squaring pre-Q matrices are very sparse, which considerably speeds up linear algebra. However, first-order methods remain the only alternative for solving (3.9) beyond a certain scale. We

quickly discuss below the implementation of two classes of methods: the conditional gradient (a.k.a. Frank-Wolfe) algorithm, and accelerated gradient methods. Alternatively, (Goemans, 2014) produced an extended formulation of the permutahedron using only $O(n \log n)$ variables and constraints, which can be used to write QP relaxations of 2-SUM with only $O(n \log n)$ variables. While the constant in these representations is high, more practical formulations are available with $O(n \log^2 n)$ variables. This formulation was tested by (Lim and Wright, 2014) while the contents of this chapter was under review, and combined with an efficient interior point solver (GUROBI) provides significant speed-up.

3.4.1 Conditional gradient

Solving (3.9) using the conditional gradient algorithm in e.g., (Frank and Wolfe, 1956) requires minimizing an affine function over the set of doubly stochastic matrices at each iteration. This amounts to solving a classical transportation (or matching) problem for which very efficient solvers exist (Portugal et al., 1996).

3.4.2 Accelerated smooth optimization

On the other hand, solving (3.9) using accelerated gradient algorithms requires solving a projection step on doubly stochastic matrices at each iteration (Nesterov, 2003). Here too, exploiting structure significantly improves the complexity of these steps. Given some matrix Π_0, the Euclidean projection problem is written

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|\Pi - \Pi_0\|_F^2 \\ \text{subject to} & D^T \Pi g + \delta \leq 0, \\ & \Pi \mathbf{1} = \mathbf{1},\ \Pi^T \mathbf{1} = \mathbf{1},\ \Pi \geq 0 \end{array} \tag{3.10}$$

in the variable $\Pi \in \mathbb{R}^{n \times n}$, with parameter $g \in \mathbb{R}^n$. The dual is written

$$\begin{array}{ll} \text{maximize} & -\frac{1}{2}\|x\mathbf{1}^T + \mathbf{1}y^T + Dzg^T - Z\|_F^2 - \mathbf{Tr}(Z^T \Pi_0) \\ & +x^T(\Pi_0 \mathbf{1} - \mathbf{1}) + y^T(\Pi_0^T \mathbf{1} - \mathbf{1}) + z(D^T \Pi_0 g + \delta) \\ \text{subject to} & z \geq 0,\ Z \geq 0 \end{array} \tag{3.11}$$

in the variables $Z \in \mathbb{R}^{n \times n}$, $x, y \in \mathbb{R}^n$ and $z \in \mathbb{R}^{n_c}$. The dual optimizes over decoupled linear constraints in $(z,\ Z)$, while x and y are unconstrained.

Each subproblem is equivalent to computing a conjugate norm and can be solved in closed form. This means that, with independent constraints (D full rank), at each iteration, explicit formulas are available to update variables block by block in the dual Euclidean projection problem (3.11)

over doubly stochastic matrices (*cf.* Algorithm 9). Problem (3.11) can thus be solved very efficiently by block-coordinate ascent, whose convergence is guaranteed in this setting (Bertsekas, 1998), and a solution to (3.10) can be reconstructed from the optimum in (3.11).

The detailed procedure for block coordinate ascent in the dual Euclidean projection problem (3.11) is described in Algorithm 9. We perform block coordinate ascent until the duality gap between the primal and the dual objective is below the required precision. Warm-starting the projection step in both primal and dual provided a very significant speed-up in our experiments.

Algorithm 9 Projection on doubly stochastic matrices.

Input: A matrix $Z \in \mathbb{R}_+^{n \times n}$, a vector $z \in \mathbb{R}_+^{n_c}$, two vectors $x, y \in \mathbb{R}^n$, a target precision ϵ, a maximum number of iterations N.

1: Set $k = 0$.
2: **while** duality gap $> \epsilon$ & $k \leq N$ **do**
3: Update dual variables

$$\begin{cases} Z = \max\{\mathbf{0},\ x\mathbf{1}^T + \mathbf{1}y^T + Dzg^T - \Pi_0\} \\ x = \frac{1}{n}(\Pi_0\mathbf{1} - (y^T\mathbf{1} + 1)\mathbf{1} - Dzg^T\mathbf{1} + Z\mathbf{1}) \\ y = \frac{1}{n}(\Pi_0^T\mathbf{1} - (x^T\mathbf{1} + 1)\mathbf{1} - gz^TD\mathbf{1} + Z^T\mathbf{1}) \\ z = \frac{1}{\|g\|_2^2}\max\{0,\ (D^TD)^{-1}(D^T(Z + \Pi_0)g + \delta - D^Txg^T\mathbf{1} - D^T\mathbf{1}g^Ty)\} \end{cases}$$

4: Set $k = k + 1$.
5: **end while**

Output: A doubly stochastic matrix Π.

3.5 Applications & numerical experiments

We now study the performance of the relaxations detailed above in some classical applications of seriation. Other applications not discussed here include: social networks, sociology, cartography, ecology, operations research, psychology (Liiv, 2010).

In most of the examples below, we will compare the performance of the spectral solution, that of the QP relaxation in (3.7) and the semi-supervised seriation QP in (3.9). In the semi-supervised experiments, we randomly sample pairwise orderings either from the true order information (if known), or from noisy ordering information. We use a simple symmetric Erdös-Rényi model for collecting these samples, so that a pair of indices (i, j) is included with probability p, with orderings sampled independently. Erdös and Rényi (1960) show that there is a sharp phase transition in the connectivity of the sampled graphs, with the graphs being almost surely disconnected when $p < \frac{(1-\epsilon)\log n}{n}$ and almost surely connected when $p > \frac{(1+\epsilon)\log n}{n}$ for $\epsilon > 0$ and n large enough. Above that threshold, i.e., when $O(n \log n)$ pairwise orders are specified, the graph is fully connected so the full variable ordering is specified *if the ordering information is noiseless*.

Of course, when the samples include errors, some of the sampled pairwise orderings could be inconsistent, so the total order is not fully specified.

3.5.1 Archeology

We reorder the rows of the Hodson's Munsingen dataset (as provided by Hodson (1968) and manually ordered by Kendall (1971)), to date 59 graves from 70 recovered artifact types (under the assumption that graves from similar periods contain similar artifacts). The results are reported in Table 3.1. We use a fraction of the pairwise orders in Kendall (1971) to solve the semi-supervised version. Note that the original data contains errors, so Kendall's ordering cannot be fully consistent. In fact, we will see that the semi-supervised relaxation actually improves on Kendall's manual ordering.

In Figure 3.3 the first plot on the left shows the row ordering on 59×70 grave by artifacts matrix given by Kendall, the middle plot is the Fiedler solution, the plot on the right is the best QP solution from 100 experiments with different Y (based on the combinatorial objective in (3.2)). The quality of these solutions is detailed in Table 3.1.

	Kendall (1971)	**Spectral**	**QP Reg**	**QP Reg + 0.1%**	**QP Reg + 47.5%**
Kendall τ	1.00	0.75	0.73±0.22	0.76±0.16	0.97±0.01
Spearman ρ	1.00	0.90	0.88±0.19	0.91±0.16	1.00±0.00
Comb. Obj.	38520	38903	41810±13960	43457±23004	37602±775
# R-constr.	1556	1802	2021±484	2050±747	1545±43

TABLE 3.1: Performance metrics (median and stdev over 100 runs of the QP relaxation, for Kendall's τ, Spearman's ρ ranking correlations (large values are good), the objective value in (3.2), and the number of R-matrix monotonicity constraint violations (small values are good), comparing Kendall's original solution with that of the Fiedler vector, the seriation QP in (3.7) and the semi-supervised seriation QP in (3.9) with 0.1% and 47.5% pairwise ordering constraints specified. Note that the semi-supervised solution actually improves on both Kendall's manual solution and on the spectral ordering.

FIGURE 3.3: The Hodson's Munsingen dataset: row ordering given by Kendall *(left)*, Fiedler solution *(center)*, best unsupervised QP solution from 100 experiments with different Y, based on combinatorial objective *(right)*.

3.5.2 Markov chains

Here, we observe many *disordered* samples from a Markov chain. The mutual information matrix of these variables must be decreasing with $|i - j|$ when ordered according to the true generating Markov chain (this is the "data processing inequality" in (Cover and Thomas, 2012, Th. 2.8.1)), hence the mutual information matrix of these variables is a *pre-R*-matrix. We can thus recover the order of the Markov chain by solving the seriation problem on this matrix. In the following example, we try to recover the order of a Gaussian Markov chain written $X_{i+1} = b_i X_i + \epsilon_i$ with $\epsilon_i \sim N(0, \sigma_i^2)$. The results are presented in Table 3.2 on 30 variables. We test performance in a noise free setting where we observe the randomly ordered model covariance, in a noisy setting with enough samples (6000) to ensure that the spectral solution stays in a perturbative regime, and finally using much fewer samples (60) so the spectral perturbation condition fails. In Figure 3.4, the first plot on the left shows the true Markov chain order, the middle plot is the Fiedler solution, the plot on the right is the best QP solution from 100 experiments with different Y (based on combinatorial objective).

	No noise	**Noise within spectral gap**	**Large noise**
True	1.00±0.00	1.00±0.00	1.00±0.00
Spectral	1.00±0.00	0.86±0.14	0.41±0.25
QP Reg	0.50±0.34	0.58±0.31	0.45±0.27
QP + 0.2%	0.65±0.29	0.40±0.26	0.60±0.27
QP + 4.6%	0.71±0.08	0.70±0.07	0.68±0.08
QP + 54.3%	0.98±0.01	0.97±0.01	0.97±0.02

TABLE 3.2: Kendall's τ between the true Markov chain ordering, the Fiedler vector, the seriation QP in (3.7) and the semi-supervised seriation QP in (3.9) with varying numbers of pairwise orders specified. We observe the (randomly ordered) model covariance matrix (no noise), the sample covariance matrix with enough samples so the error is smaller than half of the spectral gap, then a sample covariance computed using much fewer samples so the spectral perturbation condition fails.

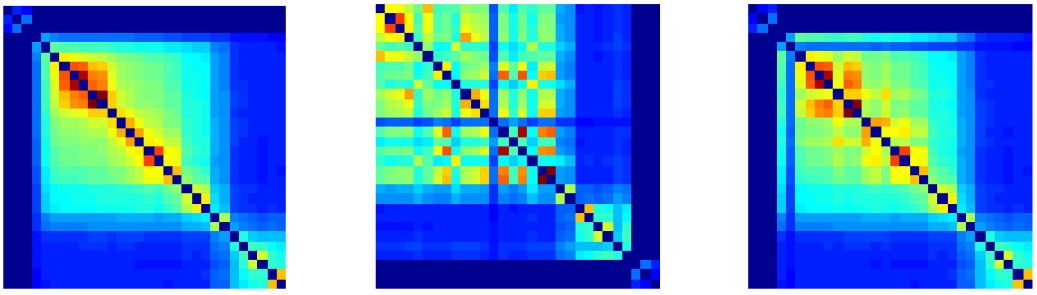


FIGURE 3.4: Markov chain experiments: true Markov chain order *(left)*, Fiedler solution *(center)*, best unsupervised QP solution from 100 experiments with different Y, based on combinatorial objective *(right)*.

3.5.3 Gene sequencing

In next generation shotgun genome sequencing experiments, DNA strands are cloned about ten to a hundred times before being decomposed into very small subsequences called “reads", each of them fifty to a few hundreds base pairs long. Current machines can only accurately sequence these small reads, which must then be reordered by “assembly" algorithms, using the overlaps between reads. These short reads are often produced in pairs, starting from both ends of a longer sequence of known length, hence a rough estimate of the distance between these “mate pairs” of reads is known, giving additional structural information on the semi-supervised assembly problem.

Here, we generate artificial sequencing data by (uniformly) sampling reads from chromosome 22 of the human genome from NCBI, then store k-mer hit versus read in a binary matrix C (a k-mer is a fixed sequence of k base pairs). If the reads are ordered correctly and have identical length, this matrix is C1P, hence we solve the C1P problem on the $\{0,1\}$-matrix whose rows correspond to k-mers hits for each read, i.e., the element (i,j) of the matrix is equal to one if k-mer j is present in read i. The corresponding *pre-R* matrix obtained CC^T, which measures overlap between reads, is extremely sparse, as it is approximately band-diagonal with roughly constant bandwidth b when reordered correctly, and computing the Fiedler vector can be done with complexity $O(bn\log n)$ w.h.p. using the Lanczos method (Kuczynski and Wozniakowski, 1992), as it amounts to computing the second largest eigenvector of $\lambda_n(L)\mathbf{I}-L$, where L is the Laplacian of the matrix. In our experiments, computing the Fiedler vector from 250000 reads takes a few seconds using MATLAB’s `eigs` on a standard desktop machine.

In practice, besides sequencing errors (handled relatively well by the high coverage of the reads), there are often repeats in long genomes. If the repeats are longer than the k-mers, the C1P assumption is violated and the order given by the Fiedler vector is not reliable anymore. On the other hand, handling the repeats is possible using the information given by mate pairs, i.e., reads that are known to be separated by a given number of base pairs in the original genome. This structural knowledge can be incorporated into the relaxation (3.9). While our algorithm for solving (3.9) only scales up to a few thousands base pairs on a regular desktop, it can be used to solve the sequencing problem hierarchically, i.e., to refine the spectral solution.

In Figure 3.5, we show the result of spectral ordering on simulated reads from human chromosome 22. The full R matrix formed by squaring the reads $\times$ kmers matrix is too large to be plotted in MATLAB and we zoom in on two diagonal block submatrices. In the first submatrix, the reordering is good and the matrix has very low bandwidth, the corresponding gene segment (called contig) is well reconstructed. In the second the reordering is less reliable, and the bandwidth is larger, so the reconstructed gene segment contains errors.

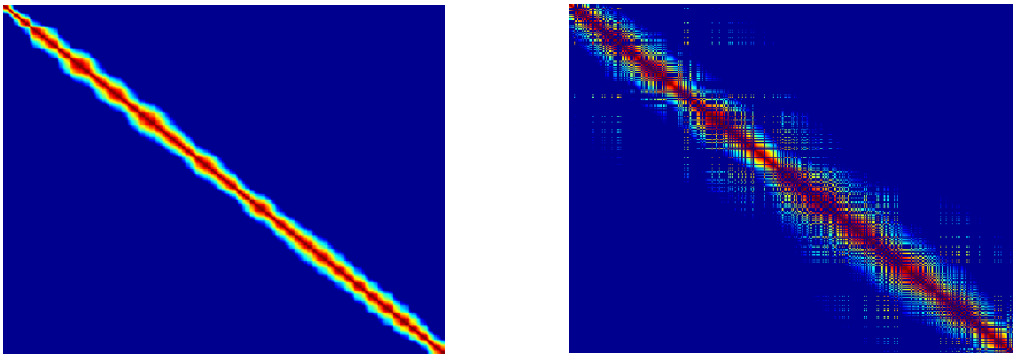


FIGURE 3.5: We plot the $reads \times reads$ matrix measuring the number of common k-mers between read pairs, reordered according to the spectral ordering on two submatrices.

In Figure 3.6, we show recovered read position versus true read position for the Fiedler vector and the Fiedler vector followed by semi-supervised seriation, where the QP relaxation is applied to groups of reads (contigs) assembled by the spectral solution, on the 250 000 reads generated in our experiments. The spectral solution orders most of these reads correctly, which means that the relaxation is solved on a matrix of dimension about 100. We see that the number of misplaced reads significantly decreases in the semi-supervised seriation solution. Looking at the correlation between the true positions and the retrieved positions of the reads, both Kendall τ and Spearman ρ are equal to one for Fiedler+QP ordering while they are equal to respectively 0.87 and 0.96 for Fiedler ordering alone. A more complete description of the assembly algorithm is given in the appendix.

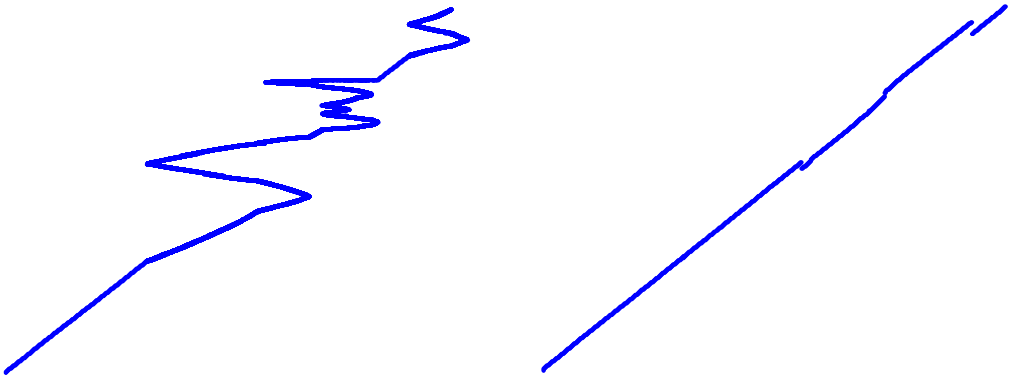


FIGURE 3.6: We plot the Fiedler and Fiedler+QP read orderings versus true ordering. The semi-supervised solution contains much fewer misplaced reads.

3.5.4 Generating Y

We conclude by testing the impact of Y on the performance of the QP relaxation in (3.6) on a simple ranking example. In Figure 3.7, we generate several matrices $Y \in \mathbb{R}^{n \times p}$ as in §3.3.2.2

and compare the quality of the solutions (permutations issued from the procedure described in 3.3.2.4) obtained for various values of the number of columns p. On the left, we plot the histogram of the values of $g^T \Pi^T L_A \Pi g$ obtained for 100 solutions with random matrices Y where $p = 1$ (i.e., rank one). On the right, we compare these results with the average value of $g^T \Pi^T L_A \Pi g$ for solutions obtained with random matrices Y with p varying from 1 to $5n$ (sample of 50 random matrices Y for each value of p). The red horizontal line, represents the best solution obtained for $p = 1$ over all experiments. By raising the value of $\lambda_1(YY^T)$, larger values of p allow for higher values of μ in Proposition 3.17, which seems to have a positive effect on performance until a point where p is much larger than n and the improvement becomes insignificant. We do not have an intuitive explanation for this behavior at this point.

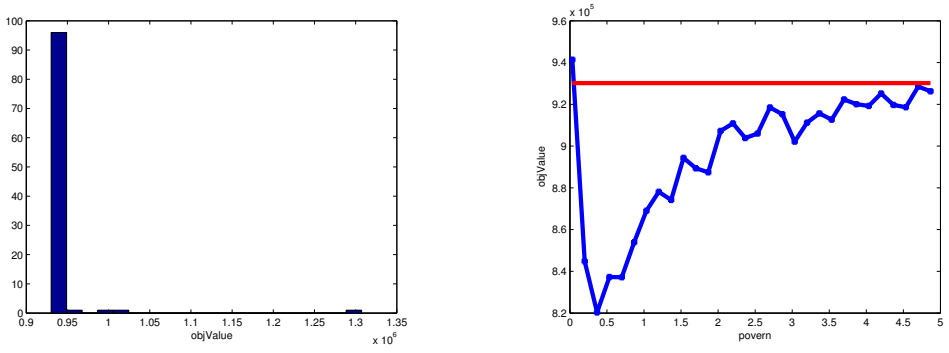


FIGURE 3.7: *Left:* we plot the histogram of the values of $g^T \Pi^T L_A \Pi g$ obtained for 50 solutions with random matrices Y where $p = 1$ (i.e., rank one). *Right:* we compare these results with the average value of $g^T \Pi^T L_A \Pi g$ for solutions obtained with random matrices Y with p varying from 1 to $5n$. The red horizontal line, represents the best solution obtained for $p = 1$ over all experiments.

3.6 Discussion

In this chapter, we have introduced new convex relaxations for the seriation problem. Besides being more robust to noise than the classical spectral relaxation of (Atkins et al., 1998), these convex relaxations also allow us to impose structural constraints on the solution, hence solve semi-supervised seriation problems. Numerical experiments on DNA de novo assembly gave promising results. Ongoing work with Antoine Recanati, Alexandre d'Aspremont (ENS Paris) and Thomas Bruls (Génoscope) is focused on how to improve the design of the similarity matrix in order to be more robust to repetitions in the DNA and high sequencing noise in reads. On the algorithmic side Lim and Wright (2014) have extended our convex relaxation of the seriation problem by using sorting networks representations of the permutohedron that are cheaper than representations of permutation matrices (Goemans, 2014). Furthermore, the use of phase retrieval algorithms to solve seriation problems is also being investigated (*cf.* Section 1.5).

3.7 Appendix

In this appendix, we first briefly detail two semidefinite programming based relaxations for the 2-SUM problem, one derived from results in (Nesterov, 1998; d'Aspremont and El Karoui, 2013), the other adapted from work on the Minimum Linear Arrangement (MLA) problem in (Even et al., 2000; Feige, 2000; Blum et al., 2000) among many others. While their complexity is effectively too high to make them practical seriation algorithms, these relaxations come with explicit approximation bounds which aren't yet available for the QP relaxation in Section 3.3.2. These SDP relaxations illustrate a clear tradeoff between approximation ratios and computational complexity: low complexity but unknown approx. ratio for the QP relaxation in (3.4), high complexity and $\sqrt{n}$ approximation ratio for the first semidefinite relaxation, very high complexity but excellent $\sqrt{\log n}$ approximation ratio for the second SDP relaxation. The question of how to derive convex relaxations with nearly dimension independent approximation ratios (e.g., $O(\sqrt{\log n})$) and good computational complexity remains open at this point.

We then describe in detail the data sets and procedures used in the DNA sequencing experiments of Section 3.5.

3.7.1 SDP relaxations & doubly stochastic matrices

Using randomization techniques derived from (Nesterov, 1998; d'Aspremont and El Karoui, 2013), we can produce approximation bounds for a relaxation of the non-convex QP representation of (3.2) derived in (3.7), namely

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}(Y^T \Pi^T L_A \Pi Y) - \mu \|P\Pi\|_F^2 \\ \text{subject to} & \Pi \mathbf{1} = \mathbf{1},\ \Pi^T \mathbf{1} = \mathbf{1},\ \Pi \geq 0, \end{array}$$

which is a (possibly non convex) quadratic program in the matrix variable $\Pi \in \mathbb{R}^{n \times n}$, where $P = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$. We now set the penalty $\mu > 0$ sufficiently high to ensure that the objective is concave and the constraint $\|\Pi\| = \sqrt{n}$ is saturated. From Proposition (3.17) above, this means $\mu > \|L_A\|_2 \|Y\|_2^2$. The solution of this concave minimization problem over the convex set of doubly stochastic matrices will then be at an extreme point, i.e., a permutation matrix. We first rewrite the above QP as a more classical maximization problem over vectors

$$\begin{array}{ll} \text{maximize} & \left(\mathbf{vec}\, \Pi^T \mathcal{A}\, \mathbf{vec}\, \Pi\right)^{1/2} \\ \text{subject to} & (\mathbf{1}^T \otimes \mathbf{I})\, \mathbf{vec}\, \Pi = \mathbf{1},\ (\mathbf{I} \otimes \mathbf{1}^T)\, \mathbf{vec}\, \Pi = \mathbf{1},\ \Pi \geq 0. \end{array}$$

We use a square root in the objective here to maintain the same homogeneity properties as in the linear arrangement problems that follow. Because the objective is constructed from a Laplacian matrix, we have $\mathbf{1}^T \mathcal{A} \mathbf{1} = 0$ so the objective is invariant by a shift in the variables. We now show

that the equality constraints can be relaxed without loss of generality. We first recall a simple scaling algorithm due to (Sinkhorn, 1964) which shows how to normalize to one the row and column sums of a strictly positive matrix. Other algorithms based on geometric programming with explicit complexity bounds can be found in e.g., (Nemirovski and Rothblum, 1999).

Algorithm 10 Matrix scaling (Sinkhorn).

Input: A matrix $\Pi \in \mathbb{R}^{m\times n}$

1: **for** $k = 1$ to $N-1$ **do**
2: Scale row sums to one: $\Pi_{k+1/2} = \mathbf{diag}(\Pi_k \mathbf{1})^{-1}\Pi_k$
3: Scale column sums to one: $\Pi_{k+1} = \Pi_{k+1/2}\,\mathbf{diag}(\mathbf{1}^T\Pi_{k+1/2})^{-1}$
4: **end for**

Output: A scaled matrix Π_N.

The next lemma shows that the only matrices satisfying both $\|\Pi\|_F = \sqrt{n}$ and $\Pi\mathbf{1} \leq \mathbf{1}$, $\Pi^T\mathbf{1} \leq \mathbf{1}$, with $\Pi \geq 0$ are doubly stochastic.

Lemma 3.18. *Let* $\Pi \in \mathbb{R}^{n\times n}$*, if* $\|\Pi\|_F = \sqrt{n}$ *and* $\Pi\mathbf{1} \leq \mathbf{1}$*,* $\Pi^T\mathbf{1} \leq \mathbf{1}$*, with* $\Pi \geq 0$*, then* Π *is doubly stochastic.*

Proof. Suppose $\Pi\mathbf{1} \leq \mathbf{1}$, $\Pi^T\mathbf{1} \leq \mathbf{1}$, $\Pi > 0$, each iteration of Algorithm 10 multiplies $\mathbf{vec}\,\Pi$ by a diagonal matrix D with diagonal coefficients greater than one, with at least one coefficient strictly greater than one if Π is not doubly stochastic, hence $\|\Pi\|_F$ is strictly increasing if Π is not doubly stochastic. This means that the only maximizers of $\|\Pi\|_F$ over the feasible set of (3.7.1) are doubly stochastic matrices. ■

We let $z = \mathbf{vec}\,\Pi$, the above lemma means that problem (3.7.1) is equivalent to the following QP

$$\begin{array}{ll} \text{maximize} & \|\mathcal{A}^{1/2}z\|_2 \\ \text{subject to} & (\mathbf{1}^T \otimes \mathbf{I})z \leq \mathbf{1},\ (\mathbf{I}\otimes \mathbf{1}^T)z \leq \mathbf{1}, \\ & z \geq 0, \end{array} \tag{QP}$$

in the variable $z \in \mathbb{R}^{n^2}$. Furthermore, since permutation matrices are binary matrices, we can impose the redundant constraints that $z_i \in \{0,1\}$ or equivalently $z_i^2 = z_i$ at the optimum. Lifting the quadratic objective and constraints as in (Shor, 1987; Lovász and Schrijver, 1991) yields the following relaxation

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(\mathcal{A}Z) \\ \text{subject to} & (\mathbf{1}^T \otimes \mathbf{I})z \leq \mathbf{1},\ (\mathbf{I}\otimes \mathbf{1}^T)z \leq \mathbf{1}, \\ & Z_{ii} = z_i,\ Z_{ij} \geq 0, \quad i,j = 1,\ldots,n, \\ & \begin{pmatrix} Z & z \\ z^T & 1 \end{pmatrix} \succeq 0, \end{array} \tag{SDP1}$$

which is a semidefinite program in the matrix variable $Z \in \mathbf{S}_{n^2}$ and the vector $z \in \mathbb{R}^{n^2}$. By adapting a randomization argument used in the MaxCut relaxation bound in (Nesterov, 1998) and adapted to the k-dense-subgraph problem in (d'Aspremont and El Karoui, 2013), we can show the following $O(\sqrt{n})$ approximation bound on the quality of this relaxation.

Proposition 3.19. *Let OPT be the optimal value of problem* (QP) *and SDP1 be that of* (SDP1)*, then*

$$0 \leq \frac{\mathbf{Tr}(\mathcal{A}G)}{4n} + \frac{SDP1}{2\pi n} \leq OPT^2 \leq SDP1.$$

with $G_{ij} = \sqrt{Z_{ii}Z_{jj}}$, $i = 1, \ldots, n$ *and* $\mathbf{Tr}(\mathcal{A}G) \leq 0$.

Proof. The fact that $\mathcal{A} \succeq 0$ by construction shows $0 \leq OPT^2 \leq SDP1$. Let $\xi \sim \mathcal{N}(0, Z)$, and define

$$y_i = \begin{cases} \sqrt{z_i} & \text{if } \xi_i \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

We write $C = \mathbf{diag}(Z)^{-1/2} Z \, \mathbf{diag}(Z)^{-1/2}$ the correlation matrix associated with Z (under the convention that $C_{ij} = 0$ whenever $Z_{ii}Z_{jj} = 0$). A classical result from (Sheppard, 1900) (see also (Johnson and Kotz, 1972, p.95)) shows

$$\mathbf{E}[y_i y_j] = \sqrt{z_i z_j}\left(\tfrac{1}{4} + \tfrac{1}{2\pi}\arcsin(C)\right), \quad i = 1, \ldots, n,$$

and $\mathcal{A} \succeq 0$ together with $\arcsin(C) \succeq C$ (with the $\arcsin(\cdot)$ taken elementwise) and $z_i = Z_{ii}$ means that, writing $G_{ij} = \sqrt{z_i z_j} = \sqrt{Z_{ii}Z_{jj}}$, we get

$$\begin{aligned} \mathbf{E}[y^T \mathcal{A} y] &= \mathbf{E}[\mathbf{Tr}(\mathcal{A} y y^T)] \\ &= \mathbf{Tr}\left(\mathcal{A}\left(G \circ \left(\frac{1}{4}\mathbf{1}\mathbf{1}^T + \frac{1}{2\pi}\arcsin(C)\right)\right)\right) \\ &\leq \mathbf{Tr}\left(\mathcal{A}\left(\frac{1}{4}G + \frac{1}{2\pi}Z\right)\right) \\ &= \frac{1}{4}\mathbf{Tr}(\mathcal{A}G) + \frac{1}{2\pi}SDP1, \end{aligned}$$

because Schur's theorem shows that $A \circ B \succeq 0$ when $A, B \succeq 0$. It remains to notice that, because $(\mathbf{1}^T \otimes \mathbf{I})z \leq \mathbf{1}$, and $(\mathbf{I} \otimes \mathbf{1}^T)z \leq \mathbf{1}$, with $z \geq 0$, then

$$(\mathbf{1}^T \otimes \mathbf{I})\sqrt{z} \leq \sqrt{n}\mathbf{1}, \quad \text{and} \quad (\mathbf{I} \otimes \mathbf{1}^T)\sqrt{z} \leq \sqrt{n}\mathbf{1},$$

so all the points y generated using this procedure are feasible for (QP) if we scale them by a factor $\sqrt{n}$. ■

While the $O(\sqrt{n})$ bound grows relatively fast with problem dimension, remember that the problem has n^2 variables because it is written on permutation *matrices*. In what follows, we will

see that better theoretical approximation bounds can be found if we write the seriation problem directly over permutation *vectors*, which is of course a much more restrictive formulation.

3.7.2 SDP relaxations & minimum linear arrangement

Several other semidefinite relaxations have been derived for the 2-SUM problem and the directly related 1-SUM, or *minimum linear arrangement* (MLA) problem. While these relaxations have very high computational complexity, to the point of being impractical, they come with excellent approximation bounds. We briefly recall these results in what follows. The 2-SUM minimization problem (3.1) is written (after taking square roots)

$$\begin{array}{ll} \text{minimize} & \left(\sum_{i,j=1}^n A_{ij}(\pi(i)-\pi(j))^2\right)^{\frac{1}{2}} \\ \text{subject to} & \pi \in \mathcal{P}. \end{array} \tag{3.12}$$

in the variable $\pi \in \mathcal{P}$ which is a permutation of the vector $(1,\ldots,n)^T$. Even et al. (2000); Feige (2000); Blum et al. (2000) form the following semidefinite relaxation

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^n A_{ij}X_{ij} \\ \text{subject to} & \frac{1}{|S|}\sum_{j\in S}(X_{ii}-2X_{ij}+X_{jj}) \geq \frac{1}{6}(|S|/2+1)(|S|+1), \quad \text{for all } S\subset[1,n],\ i=1,\ldots,n \\ & \frac{1}{|S|}\sum_{k\in S}\Delta^2(i,j,k) \geq \epsilon(X_{ii}-2X_{ij}+X_{jj})|S|^2, \quad \text{for all } S\subset[1,n],\ i,j=1,\ldots,n \\ & X \succeq 0,\ X_{ij}\geq 0, \quad i,j=1,\ldots,n \end{array} \tag{SDP2}$$

in the variable $X \in \mathbf{S}_n$, where $\epsilon > 0$ and $\Delta(i,j,k)$ is given by the determinant

$$\Delta(i,j,k) = \begin{vmatrix} X_{jj}-2X_{ij}+X_{ii} & X_{jk}-X_{ij}-X_{jk}+X_{ii} \\ X_{jk}-X_{ij}-X_{jk}+X_{ii} & X_{kk}-2X_{ik}+X_{ii} \end{vmatrix}.$$

(Blum et al., 2000, Th. 2) shows that if OPT is the optimal value of the 2-SUM problem (3.12) and SDP2 the optimal value of the relaxation in (SDP2), then

$$SDP2\ (\log n)^{-1/2} \leq OPT \leq SDP2\ (\log n)^{3/2}.$$

While problem (SDP2) has an exponential number of constraints, efficient linear separation oracles can be constructed for the last two spreading constraints, hence the problem can be solved in polynomial time (Grötschel et al., 1988).

Tighter bounds can be obtained by exploiting approximation results on the minimum linear arrangement problem, noting that, after taking the square root of the objective, the 2-SUM

problem is equivalent to

$$\min_{\pi\in\mathcal{P}} \max_{\{\|V\|_F\leq 1,\, V\geq 0\}} \sum_{i,j=1}^{n} V_{ij}A_{ij}^{1/2}|\pi(i)-\pi(j)| \tag{3.13}$$

in the variables $\pi \in \mathcal{P}$ and $V \in \mathbb{R}^{n\times n}$ (note that this is true for the support function of any set contained in the nonnegative orthant). Using results in (Rao and Richa, 2005; Feige and Lee, 2007; Charikar et al., 2010), the minimum linear arrangement problem, written

$$\min_{\pi\in\mathcal{P}} \sum_{i,j=1}^{n} W_{ij}|\pi(i)-\pi(j)| \tag{MLA}$$

over the variable $\pi \in \mathcal{P}$, with nonnegative weights $W \in \mathbb{R}^{n\times n}$, can be relaxed as

$$\begin{array}{ll}
\text{minimize} & \sum_{i,j=1}^{n} W_{ij}(X_{ii}-2X_{ij}+X_{jj}) \\
\text{subject to} & \frac{1}{|S|}\sum_{j\in S}(X_{ii}-2X_{ij}+X_{jj}) \geq \frac{|S|^2}{5}, \quad \text{for all } S\subset[1,n],\ i=1,\ldots,n \\
& (X_{ii}-2X_{ij}+X_{jj}) \leq (X_{ii}-2X_{ik}+X_{kk})+(X_{kk}-2X_{kj}+X_{jj}), \quad i,j,k=1,\ldots,n \\
& (X_{ii}-2X_{ij}+X_{jj}) \geq 1, \quad i,j=1,\ldots,n \\
& X \succeq 0,
\end{array} \tag{SDP3}$$

in the variable $X \in \mathbf{S}_n$. The constraints above ensure that $d_{ij} = (X_{ii}-2X_{ij}+X_{jj})$ is a squared Euclidean metric (hence a metric of negative type). If MLA is the optimal value of the minimum linear arrangement problem (MLA) and SDP3 the optimum of the relaxation in (SDP3), (Feige and Lee, 2007, Th. 2.1) and (Charikar et al., 2010) show that

$$SDP3 \leq MLA \leq SDP3\, O(\sqrt{\log n}\,\log\log n),$$

which immediately yields a convex relaxation with $O(\sqrt{\log n}\,\log\log n)$ approximation ratio for the minmax formulation of the 2-SUM problem in (3.13).

3.7.3 Procedure for gene sequencing

We first order all the reads using the spectral algorithm. Then, in order to handle repeats in the DNA sequence, we adopt a divide and conquer approach and reorder smaller groups of reads partitioned using the spectral order. Finally we use the information given by mate pairs to reorder the resulting clusters of reads, using the QP relaxation. Outside of spectral computations which take less than a minute in our experiments, most computations can be naively parallelized. The details of the procedure are given below.

- Extract uniformly reads of length a few hundreds bp (base pairs) from DNA sequence. In our experiments, we artificially extract reads of length 200 bp from the true sequence of

a million bp of the human chromosome 22. We perform a high coverage (each bp is contained in approx. 50 reads) uniform sampling. To replicate the setting of real sequencing data, we extract pairs of reads, with a distance of 5000 bp between each "mate" pairs.

- Extract all possible k-mers from reads, i.e., for each read, record all subsequence of size k. We use k=100 in our experiments. The size of k-mers may be tuned to deal with noise in sequencing data (use small k) or repeats (use large k).

- Solve the C1P problem on the $\{0, 1\}$-matrix whose rows correspond to k-mers hits for each read, i.e., the element (i, j) of the matrix is equal to one if k-mer j is included in read i. Note that solving this C1P problem corresponds to reordering the similarity matrix between reads whose element (r, s) is the number of shared k-mers between reads r and s. In the presence of noise in sequencing data, this similarity matrix can be made more robust by recomputing for instance an edit distance between reads sharing k-mers. Moreover, if there are no repeated k-mers in the original sequence, i.e., a k-mer appears in two reads only if they overlap in the original sequence, then the C1P problem is solved exactly by the spectral relaxation and the original DNA sequence is retrieved by concatenating the overlapping reordered reads. Unfortunately, for large sequences, repeats are frequent and the spectral solution "mixes" together different areas of the original sequence. We deal with repeats in what follows.

- We extract contigs from the reordered reads: extract with high coverage (e.g., 10) sequences of a few thousands reads from the reordered sequence of reads (250 000 reads in our experiments). Although there were repeats in the whole sequence, a good proportion of the contigs do not contain reads with repeats. By reordering each contig (using the spectral relaxation) and looking at the corresponding similarity (R-like) matrix, we can discriminate between "good" contigs (with no repeats and therefore a perfectly reordered similarity matrix which is an R-matrix) and "bad" contigs (with repeats and a badly reordered similarity matrix).

- Reorder the "good" contigs from the previous step using the spectral relaxation and agglomerate overlapping contigs. The aggregation can be done using again the spectral algorithm on the sub matrix of the original similarity matrix corresponding to the two clusters of reads. Now there should be only a few (long) contigs left (usually less than a few hundreds in our experiments).

- Use the mate pairs to refine the order of the contigs with the QP relaxation to solve the semi-supervised seriation problem. Gaps are filled by incorporating the reads from the "bad" contigs (contigs with repeats).

Overall, the spectral preprocessing usually shrinks the ordering problem down to dimension $n \sim 100$, which is then solvable using the convex relaxations detailed in Section 3.3.

Chapter 4

Spectral Ranking Using Seriation

Chapter abstract: We describe a seriation algorithm for ranking a set of items given pairwise comparisons between these items. Intuitively, the algorithm assigns similar rankings to items that compare similarly with all others. It does so by constructing a similarity matrix from pairwise comparisons, using seriation methods to reorder this matrix and construct a ranking. We first show that this spectral seriation algorithm recovers the true ranking when all pairwise comparisons are observed and consistent with a total order. We then show that ranking reconstruction is still exact when some pairwise comparisons are corrupted or missing, and that seriation based spectral ranking is more robust to noise than classical scoring methods. Finally, we bound the ranking error when only a random subset of the comparions are observed. An additional benefit of the seriation formulation is that it allows us to solve semi-supervised ranking problems. Experiments on both synthetic and real datasets demonstrate that seriation based spectral ranking achieves competitive and in some cases superior performance compared to classical ranking methods.

The material of this part is based on the following publications:
F. Fogel, A. d'Aspremont, M. Vojnovic: Serialrank: spectral ranking using seriation. In *Advances in Neural Information Processing Systems*, pp. 900-908. 2014.
F. Fogel, A. d'Aspremont, M. Vojnovic: Spectral ranking using seriation. In submission.

4.1 Introduction

We study in this chapter the problem of ranking a set of n items given pairwise comparisons between these items[1]. The problem of aggregating binary relations has been formulated more than two centuries ago, in the context of emerging social sciences and voting theories (de Borda,

[1] A subset of these results appeared at NIPS 2014.

1781; de Condorcet, 1785). The setting we study here goes back at least to (Zermelo, 1929; Kendall and Smith, 1940) and seeks to reconstruct a ranking of items from pairwise comparisons reflecting a total ordering. In this case, the directed graph of all pairwise comparisons, where every pair of vertices is connected by exactly one of two possible directed edges, is usually called a *tournament* graph in the theoretical computer science literature or a "round robin" in sports, where every player plays every other player once and each preference marks victory or defeat. The motivation for this formulation often stems from the fact that in many applications, e.g. music, images, and movies, preferences are easier to express in relative terms (e.g. a is better than b) rather than absolute ones (e.g. a should be ranked fourth, and b seventh). In practice, the information about pairwise comparisons is usually *incomplete*, especially in the case of a large set of items, and the data may also be *noisy*, that is some pairwise comparisons could be incorrectly measured and inconsistent with a total order.

Ranking is a classical problem but its formulations vary widely. In particular, assumptions about how the pairwise preference information is obtained vary a lot from one reference to another. A subset of preferences is measured adaptively in (Ailon, 2011; Jamieson and Nowak, 2011), while (Freund et al., 2003; Negahban et al., 2012) extract them at random. In other settings, the full preference matrix is observed, but is perturbed by noise: in e.g. (Bradley and Terry, 1952; Luce, 1959; Herbrich et al., 2006), a parametric model is assumed over the set of permutations, which reformulates ranking as a maximum likelihood problem.

Loss functions, performance metrics and algorithmic approaches vary as well. Kenyon-Mathieu and Schudy (2007), for example, derive a PTAS for the minimum feedback arc set problem on tournaments, i.e. the problem of finding a ranking that minimizes the number of upsets (a pair of players where the player ranked lower on the ranking beats the player ranked higher). In practice, the complexity of this method is relatively high, and other authors (see e.g. Keener, 1993; Negahban et al., 2012) have been using spectral methods to produce more efficient algorithms (each pairwise comparison is understood as a link pointing to the preferred item). In other cases, such as the classical Analytic Hierarchy Process (AHP) (Saaty, 1980; Barbeau, 1986) preference information is encoded in a "reciprocal" matrix whose Perron-Frobenius eigenvector provides the global ranking. Simple scoring methods such as the point difference rule (Huber, 1963; Wauthier et al., 2013) produce efficient estimates at very low computational cost. Website ranking methods such as PageRank (Page et al., 1998) and HITS (Kleinberg, 1999) seek to rank web pages based on the hyperlink structure of the web, where links do not necessarily express consistent preference relationships (e.g. a can link to b and b can link c, and c can link to a). (Negahban et al., 2012) adapt the PageRank argument to the ranking from pairwise comparisons and Vigna (2009) provides a review of ranking algorithms given pairwise comparisons, in particular those involving the estimation of the stationary distribution of a Markov chain. Ranking has also been approached as a prediction problem, i.e. learning to rank (Schapire et al., 1998;

Rajkumar and Agarwal, 2014), with (Joachims, 2002) for example using support vector machines to learn a score function. Finally, in the Bradley-Terry-Luce framework, where multiple observations on pairwise preferences are observed and assumed to be generated by a generalized linear model, the maximum likelihood problem is usually solved using fixed point algorithms or EM-like majorization-minimization techniques (Hunter, 2004). Jiang et al. (2011) describes the HodgeRank algorithm, which formulates ranking given pairwise comparisons as a least-square problem. This formulation is based on Hodge theory and provides tools to measure the consistency of a set of pairwise comparisons with the existence of a global ranking. Duchi et al. (2010, 2013) analyze the consistency of various ranking algorithms given pairwise comparisons and a query. Preferences are aggregated through standard procedures, e.g., computing the mean of comparisons from different users, then ranking are derived using classical algorithms, e.g., Borda Count, Bradley-Terry-Model maximum likelihood estimation, least squares, odd-ratios (Saaty, 2003).

Here, we show that the ranking problem is directly related to another classical ordering problem, namely *seriation*. Given a similarity matrix between a set of n items and assuming that the items can be ordered along a chain (path) such that the similarity between items decreases with their distance within this chain (i.e. a total order exists), the seriation problem seeks to reconstruct the underlying linear ordering based on unsorted, possibly noisy, pairwise similarity information. Atkins et al. (1998) produced a spectral algorithm that exactly solves the seriation problem in the noiseless case, by showing that for similarity matrices computed from serial variables, the ordering of the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix (a.k.a. the Fiedler vector) matches that of the variables. In practice, this means that performing spectral ordering on the similarity matrix exactly reconstructs the correct ordering provided items are organized in a chain.

We adapt these results to ranking to produce a very efficient *spectral ranking algorithm with provable recovery and robustness guarantees*. Furthermore, the seriation formulation allows us to handle semi-supervised ranking problems. In Chapter 3 we have shown that seriation is equivalent to the 2-SUM problem and studied convex relaxations to seriation in a semi-supervised setting, where additional structural constraints are imposed on the solution. Several authors (Blum et al., 2000; Feige and Lee, 2007) have also focused on the directly related Minimum Linear Arrangement (MLA) problem, for which excellent approximation guarantees exist in the noisy case, albeit with very high polynomial complexity.

The main contributions of this paper can be summarized as follows. We link seriation and ranking by showing how to construct a consistent similarity matrix based on consistent pairwise comparisons. We then recover the true ranking by applying the spectral seriation algorithm in (Atkins et al., 1998) to this similarity matrix (we call this method SerialRank in what follows). In the noisy case, we then show that spectral seriation can perfectly recover the true ranking

even when some of the pairwise comparisons are either corrupted or missing, provided that the pattern of errors is somewhat unstructured. We show in particular that, in a regime where a high proportion of comparisons are observed, some incorrectly, the spectral solution is more robust to noise than classical scoring based methods. On the other hand, when only few comparisons are observed, we show that for Erdös-Rényi graphs, i.e., when pairwise comparisons are observed independently with a given probability, $\Omega(n \log^4 n)$ comparisons suffice for ℓ_2 consistency of the Fiedler vector and hence ℓ_2 consistency of the retreived ranking w.h.p. On the other hand we need $\Omega(n^{3/2} \log^4 n)$ comparisons to retrieve a ranking whose local perturbations are bounded in ℓ_∞ norm. Since for Erdös-Rényi graphs the induced graph of comparisons is connected with high probability only when the total number of pairs sampled scales as $\Omega(n \log n)$ (aka the coupon collector effect), we need at least that many comparisons in order to retrieve a ranking, therefore the ℓ_2 consistency result can be seen as optimal up to a polylogarithmic factor. Finally, we use the seriation results in Chapter 3 to produce semi-supervised ranking solutions.

The paper is organized as follows. In Section 4.2 we recall definitions related to seriation, and link ranking and seriation by showing how to construct well ordered similarity matrices from well ranked items. In Section 4.3 we apply the spectral algorithm of (Atkins et al., 1998) to reorder these similarity matrices and reconstruct the true ranking in the noiseless case. In Section 4.4 we then show that this spectral solution remains exact in a noisy regime where a random subset of comparisons is corrupted. In Section 4.5 we analyze ranking perturbation results when only few comparisons are given following an Erdös-Rényi graph. Finally, in Section 4.6 we illustrate our results on both synthetic and real datasets, and compare ranking performance with classical MLE, spectral and scoring based approaches.

4.2 Seriation, Similarities & Ranking

In this section we first introduce the seriation problem, i.e. reordering items based on pairwise similarities. We then show how to write the problem of ranking given pairwise comparisons as a seriation problem.

4.2.1 The Seriation Problem

The seriation problem seeks to reorder n items given a similarity matrix between these items, such that the more similar two items are, the closer they should be. This is equivalent to supposing that items can be placed on a chain where the similarity between two items decreases with the distance between these items in the chain. We formalize this below, following (Atkins et al., 1998).

Definition 4.1. We say that a matrix $A \in \mathbf{S}_n$ is an R-matrix (or Robinson matrix) if and only if it is symmetric and $A_{i,j} \leq A_{i,j+1}$ and $A_{i+1,j} \leq A_{i,j}$ in the lower triangle, where $1 \leq j < i \leq n$.

Another way to formulate R-matrix conditions is to impose $A_{ij} \geq A_{kl}$ if $|i-j| \leq |k-l|$ off-diagonal, i.e. the coefficients of A decrease as we move away from the diagonal. We also introduce a definition for strict R-matrices A, whose rows and columns cannot be permuted without breaking the R-matrix monotonicity conditions. We call *reverse identity* permutation the permutation that puts rows and columns $1, 2, \ldots, n$ of a matrix A in reverse order $n, n-1, \ldots, 1$.

Definition 4.2. An R-matrix $A \in \mathbf{S}_n$ is called strict-R if and only if the identity and reverse identity permutations of A are the only permutations reordering A as an R-matrix.

Note that this definition is less restrictive than the definition from Chapter 3 (any R-matrix with only strict R-constraints is a strict R-matrix). Following (Atkins et al., 1998), we will say that A is *pre-R* if there is a permutation matrix Π such that $\Pi A \Pi^T$ is an R-matrix. Given a pre-R matrix A, the seriation problem consists in finding a permutation Π such that $\Pi A \Pi^T$ is an R-matrix. Note that there might be several solutions to this problem. In particular, if a permutation Π is a solution, then the reverse permutation is also a solution. When only two permutations of A produce R-matrices, A will be called *pre-strict-R*.

4.2.2 Constructing Similarity Matrices from Pairwise Comparisons

Given an ordered input pairwise comparison matrix, we now show how to construct a similarity matrix which is *strict-R* when all comparisons are given and consistent with the identity ranking (i.e., items are ranked in increasing order of indices). This means that the similarity between two items decreases with the distance between their ranks. We will then be able to use the spectral seriation algorithm by (Atkins et al., 1998) described in Section 4.3 to reconstruct the true ranking from a disordered similarity matrix.

We first show how to compute a pairwise similarity from pairwise comparisons between items by counting the number of matching comparisons. Another formulation allows us to handle the generalized linear model. These two examples are only two particular instances of a broader class of ranking algorithms derived here. Any method which produces R-matrices from pairwise preferences yields a valid ranking algorithm.

4.2.2.1 Similarities from Pairwise Comparisons

Suppose we are given a matrix of pairwise comparisons $C \in \{-1, 0, 1\}^{n \times n}$ such that $C_{i,j} = -C_{j,i}$ for every $i \neq j$ and

$$C_{i,j} = \begin{cases} 1 & \text{if } i \text{ is ranked higher than } j \\ 0 & \text{if } i \text{ and } j \text{ are not compared or in a draw} \\ -1 & \text{if } j \text{ is ranked higher than } i \end{cases} \tag{4.1}$$

setting $C_{i,i} = 1$ for all $i \in \{1, \ldots, n\}$. We define the pairwise similarity matrix S^{match} as

$$S^{\text{match}}_{i,j} = \sum_{k=1}^{n} \left(\frac{1 + C_{i,k} C_{j,k}}{2} \right). \tag{4.2}$$

Since $C_{i,k}C_{j,k} = 1$, if $C_{i,k}$ and $C_{j,k}$ have matching signs, and $C_{i,k}C_{j,k} = -1$ if they have opposite signs, $S^{\text{match}}_{i,j}$ counts the number of matching comparisons between i and j with other reference items k. If i or j is not compared with k, then $C_{i,k}C_{j,k} = 0$ and the term $(1 + C_{i,k}C_{j,k})/2$ has an neutral effect on the similarity of $1/2$. Note that we also have

$$S^{\text{match}} = \frac{1}{2} \left(n \mathbf{1}\mathbf{1}^T + CC^T \right). \tag{4.3}$$

The intuition behind the similarity S^{match} is easy to understand in a tournament setting: players that beat the same players and are beaten by the same players should have a similar ranking.

The next result shows that when all comparisons are given and consistent with the identity ranking, then the similarity matrix S^{match} is a strict R-matrix. Without loss of generality, we assume that items are ranked in increasing order of their indices. In the general case, we can simply replace the *strict-R* property by the *pre-strict-R* property.

Proposition 4.3. *Given all pairwise comparisons between items ranked according to the identity permutation (with no ties), the similarity matrix S^{match} constructed in (4.2) is a strict R-matrix and*

$$S^{\text{match}}_{i,j} = n - |i - j| \tag{4.4}$$

for all $i, j = 1, \ldots, n$.

Proof. Since items are ranked as $1, 2, \ldots, n$ with no ties and all comparisons given, $C_{i,j} = -1$ if $i < j$ and $C_{i,j} = 1$ otherwise. Therefore we obtain from definition (4.2)

$$\begin{aligned} S^{\text{match}}_{i,j} &= \sum_{k=1}^{\min(i,j)-1} \left(\frac{1+1}{2}\right) + \sum_{k=\min(i,j)}^{\max(i,j)-1} \left(\frac{1-1}{2}\right) + \sum_{k=\max(i,j)}^{n} \left(\frac{1+1}{2}\right) \\ &= n - (\max(i,j) - \min(i,j)) \\ &= n - |i-j| \end{aligned}$$

This means in particular that S^{match} is strictly positive and its coefficients are strictly decreasing when moving away from the diagonal, hence S^{match} is a strict R-matrix. ∎

4.2.2.2 Similarities in the Generalized Linear Model

Suppose that paired comparisons are generated according to a generalized linear model (GLM), i.e., we assume that the outcomes of paired comparisons are independent and for any pair of distinct items, item i is observed ranked higher than item j with probability

$$P_{i,j} = H(\nu_i - \nu_j) \tag{4.5}$$

where $\nu \in \mathbb{R}^n$ is a vector of skill parameters and $H : \mathbb{R} \to [0,1]$ is a function that is increasing on $\mathbb{R}$ and such that $H(-x) = 1 - H(x)$ for all $x \in \mathbb{R}$, and $\lim_{x\to-\infty} H(x) = 0$ and $\lim_{x\to\infty} H(x) = 1$. A well known special instance of the generalized linear model is the Bradley-Terry-Luce model for which $H(x) = 1/(1+e^{-x})$, for $x \in \mathbb{R}$.

Let $m_{i,j}$ be the number of times items i and j were compared, $C^s_{i,j} \in \{-1, 1\}$ be the outcome of comparison s and Q be the matrix of corresponding sample probabilities, i.e. if $m_{i,j} > 0$ we have

$$Q_{i,j} = \frac{1}{m_{i,j}} \sum_{s=1}^{m_{i,j}} \frac{C^s_{i,j} + 1}{2}$$

and $Q_{i,j} = 1/2$ in case $m_{i,j} = 0$. We define the similarity matrix S^{glm} from the observations Q as

$$S^{\text{glm}}_{i,j} = \sum_{k=1}^{n} \mathbf{1}_{\{m_{i,k}m_{j,k}>0\}} \left(1 - |Q_{i,k} - Q_{j,k}|\right) + \frac{\mathbf{1}_{\{m_{i,k}m_{j,k}=0\}}}{2}. \tag{4.6}$$

Since the comparison observations are independent we have that $Q_{i,j}$ converges to $P_{i,j}$ as $m_{i,j}$ goes to infinity and the central limit theorem implies that $S^{\text{glm}}_{i,j}$ converges to a Gaussian variable with mean

$$\sum_{k=1}^{n} \left(1 - |P_{i,k} - P_{j,k}|\right).$$

The result below shows that this limit similarity matrix is a strict R-matrix when the variables are properly ordered.

Proposition 4.4. *If the items are ordered according to the order in decreasing values of the skill parameters, the similarity matrix S^{glm} is a strict R matrix with high probability as the number of observations goes to infinity.*

Proof. Without loss of generality, we suppose the true order is $1, 2, \ldots, n$, with $\nu(1) > \ldots > \nu(n)$. For any i, j, k such that $i > j$, using the GLM assumption (i) we get

$$P_{i,k} = H(\nu(i) - \nu(k)) > H(\nu(j) - \nu(k)) = P_{j,k}.$$

Since empirical probabilities $Q_{i,j}$ converge to $P_{i,j}$, when the number of observations is large enough, we also have $Q_{i,k} > Q_{j,k}$ for any i, j, k such that $i > j$ (we focus w.l.o.g. on the lower triangle), and we can therefore remove the absolute value in the expression of $S^{\mathrm{glm}}_{i,j}$ for $i > j$. Hence for any $i > j$ we have

$$\begin{aligned} S^{\mathrm{glm}}_{i+1,j} - S^{\mathrm{glm}}_{i,j} &= -\sum_{k=1}^{n} |Q_{i+1,k} - Q_{j,k}| + \sum_{k=1}^{n} |Q_{i,k} - Q_{j,k}| \\ &= \sum_{k=1}^{n} -(Q_{i+1,k} - Q_{j,k}) + (Q_{i,k} - Q_{j,k}) \\ &= \sum_{k=1}^{n} Q_{i,k} - Q_{i+1,k} < 0. \end{aligned}$$

Similarly for any $i > j$, $S^{\mathrm{glm}}_{i,j-1} - S^{\mathrm{glm}}_{i,j} < 0$, so S^{glm} is a strict R-matrix. ∎

Notice that we recover the original definition of S^{match} in the case of binary comparisons, though it does not fit in the Generalized Linear Model. Note also that these definitions can be directly extended to the setting where multiple comparisons are available for each pair and aggregated in comparisons that take fractional values (e.g., a tournament setting where participants play several times against each other).

4.3 Spectral Algorithms

We first recall how spectral ordering can be used to recover the true ordering in seriation problems. We then apply this method to the ranking problem.

4.3.1 Spectral Seriation Algorithm

We use the spectral computation method originally introduced in (Atkins et al., 1998) to solve the seriation problem based on the similarity matrices defined in the previous section. We

first recall the definition of the Fiedler vector (which is shown to be unique in our setting in Lemma 4.7).

Definition 4.5. The Fiedler value of a symmetric, nonnegative and irreducible matrix A is the smallest non-zero eigenvalue of its Laplacian matrix $L_A = \mathbf{diag}(A\mathbf{1}) - A$. The corresponding eigenvector is called Fiedler vector and is the optimal solution to $\min\{y^T L_A y : y \in \mathbb{R}^n, y^T \mathbf{1} = 0, \|y\|_2 = 1\}$.

The main result from (Atkins et al., 1998), detailed below, shows how to reorder pre-R matrices in a noise free case.

Proposition 4.6. *(Atkins et al., 1998, Th. 3.3) Let $A \in \mathbf{S}_n$ be an irreducible pre-R-matrix with a simple Fiedler value and a Fiedler vector v with no repeated values. Let $\Pi_1 \in \mathcal{P}$ (respectively, Π_2) be the permutation such that the permuted Fiedler vector $\Pi_1 v$ is strictly increasing (decreasing). Then $\Pi_1 A \Pi_1^T$ and $\Pi_2 A \Pi_2^T$ are R-matrices, and no other permutations of A produce R-matrices.*

The next technical lemmas extend the results in Atkins et al. (1998) to strict R-matrices and will be used to prove Theorem 4.10 in next section. The first one shows that without loss of generality, the Fiedler value is simple.

Lemma 4.7. *If A is an irreducible R-matrix, up to a uniform shift of its coefficients, A has a simple Fiedler value and a monotonic Fiedler vector.*

Proof. We use (Atkins et al., 1998, Th. 4.6) which states that if A is an irreducible R-matrix with $A_{n,1} = 0$, then the Fiedler value of A is a simple eigenvalue. Since A is an R-matrix, $A_{n,1}$ is among its minimal elements. Subtracting it from A does not affect the nonnegativity of A and we can apply (Atkins et al., 1998, Th. 4.6). Monotonicity of the Fiedler vector then follows from (Atkins et al., 1998, Th. 3.2). ■

The next lemma shows that the Fiedler vector is strictly monotonic if A is a strict R-matrix.

Lemma 4.8. *Let $A \in \mathbf{S}_n$ be an irreducible R-matrix. Suppose there are no distinct indices $r < s$ such that for any $k \notin [r, s]$, $A_{r,k} = A_{r+1,k} = \ldots = A_{s,k}$, then, up to a uniform shift, the Fiedler value of A is simple and its Fiedler vector is strictly monotonic.*

Proof. By Lemma 4.7, the Fiedler value of A is simple (up to a uniform shift of A). Let x be the corresponding Fiedler vector of A, x is monotonic by Lemma 4.7. Suppose $[r, s]$ is a nontrivial maximal interval such that $x_r = x_{r+1} = \ldots = x_s$, then by (Atkins et al., 1998, lemma 4.3), for any $k \notin [r, s]$, $A_{r,k} = A_{r+1,k} = \ldots = A_{s,k}$, which contradicts the initial assumption. Therefore x is strictly monotonic. ■

In fact, we only need a small portion of the R-constraints to be strict for the previous lemma to hold. We now show that the main assumption on A in Lemma 4.8 is equivalent to A being strict-R.

Lemma 4.9. *An irreducible R-matrix $A \in \mathbf{S}_n$ is strictly R if and only if there are no distinct indices $r < s$ such that for any $k \notin [r, s]$, $A_{r,k} = A_{r+1,k} = \ldots = A_{s,k}$.*

Proof. Let $A \in \mathbf{S}_n$ an R-matrix. Let us first suppose there are no distinct indices $r < s$ such that for any $k \notin [r, s]$, $A_{r,k} = A_{r+1,k} = \ldots = A_{s,k}$. By Lemma 4.8 the Fiedler value of A is simple and its Fiedler vector is strictly monotonic. Hence by Proposition 4.6, only the identity and reverse identity permutations of A produce R-matrices. Now suppose there exist two distinct indices $r < s$ such that for any $k \notin [r, s]$, $A_{r,k} = A_{r+1,k} = \ldots = A_{s,k}$. In addition to the identity and reverse identity permutations, we can locally reverse the order of rows and columns from r to s, since the sub matrix $A_{r:s,r:s}$ is an R-matrix and for any $k \notin [r, s]$, $A_{r,k} = A_{r+1,k} = \ldots = A_{s,k}$. Therefore at least four different permutations of A produce R-matrices, which means that A is not strictly R. ■

4.3.2 SerialRank: a Spectral Ranking Algorithm

In Section 4.2, we showed that similarities S^{match} and S^{glm} are *pre-strict-R* when all comparisons are available and consistent with an underlying ranking of items. We now use the spectral seriation method in (Atkins et al., 1998) to reorder these matrices and produce a ranking. Spectral ordering requires computing an extremal eigenvector, at a cost of $O(n^2 \log n)$ flops (Kuczynski and Wozniakowski, 1992). We call this algorithm SerialRank and prove the following result.

Theorem 4.10. *Given all pairwise comparisons for a set of totally ordered items and assuming there are no ties between items, algorithm SerialRank, i.e., sorting the Fiedler vector of the matrix S^{match} defined in (4.3), recovers the true ranking of items.*

Proof. From Proposition 4.3, under assumptions of the proposition S^{match} is a pre-strict R-matrix. Now combining the definition of strict-R matrices in Lemma 4.9 with Lemma 4.8, we deduce that Fiedler value of S^{match} is simple and its Fiedler vector has no repeated values. Hence by Proposition 4.6, only the two permutations that sort the Fiedler vector in increasing and decreasing order produce strict R-matrices and are candidate rankings (by Proposition 4.3 S^{match} is a strict R-matrix when ordered according to the true ranking). Finally we can choose between the two candidate rankings (increasing and decreasing) by picking the one with the least upsets. ■

Algorithm 11 (SerialRank)

Input: A set of pairwise comparisons $C_{i,j} \in \{-1, 0, 1\}$ or $[-1, 1]$.

1: Compute a similarity matrix S as in §4.2.2

2: Compute the Laplacian matrix

$$L_S = \mathbf{diag}(S\mathbf{1}) - S \qquad \text{(SerialRank)}$$

3: Compute the Fiedler vector of S.

Output: A ranking induced by sorting the Fiedler vector of S (choose either increasing or decreasing order to minimize the number of upsets).

Similar results apply for S^{glm} given enough comparisons in the Generalized Linear Model. This last result guarantees recovery of the true ranking of items in the noiseless case. In the next section, we will study the impact of corrupted or missing comparisons on the inferred ranking of items.

4.4 Exact Recovery with Corrupted and Missing Comparisons

In this section we study the robustness of SerialRank using S^{match} with respect to noisy and missing pairwise comparisons. We will see that noisy comparisons cause ranking ambiguities for the point score method and that such ambiguities are to be lifted by the spectral ranking algorithm. We show in particular that the SerialRank algorithm recovers the exact ranking when the pattern of errors is random and errors are not too numerous. We first study the impact of one corrupted comparison on SerialRank, then extend the result to multiple corrupted comparisons. A similar analysis is provided for missing comparisons as Corollary 4.27. in the Appendix. Finally, Proposition 4.14 provides an estimate of the number of randomly corrupted entries that

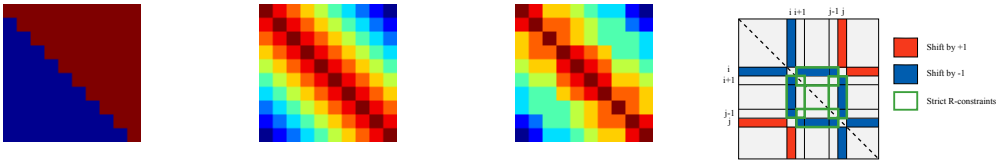


FIGURE 4.1: The matrix of pairwise comparisons C *(far left)* when the rows are ordered according to the true ranking. The corresponding similarity matrix S^{match} is a strict R-matrix *(center left)*. The same S^{match} similarity matrix with comparison (3,8) corrupted *(center right)*. With one corrupted comparison, S^{match} keeps enough strict R-constraints to recover the right permutation. In the noiseless case, the difference between all coefficients is at least one and after introducing an error, the coefficients inside the green rectangles still enforce strict R-constraints *(far right)*.

can be tolerated for perfect recovery of the true ranking. We begin by recalling the definition of the *point score* of an item.

Definition 4.11. The *point score* w_i of an item i, also known as point-difference, or *row-sum* is defined as $w_i = \sum_{k=1}^{n} C_{k,i}$, which corresponds to the number of wins minus the number of losses in a tournament setting.

In the following we will denote by w the point score vector.

Proposition 4.12. *Given all pairwise comparisons $C_{s,t} \in \{-1, 1\}$ between items ranked according to their indices, suppose the sign of one comparison $C_{i,j}$ (and its counterpart $C_{j,i}$) is switched, with $i < j$. If $j - i > 2$ then S^{match} defined in (4.3) remains* strict-R*, whereas the point score vector w has ties between items i and $i+1$ and items j and $j-1$.*

Proof. We give some intuition for the result in Figure 4.1. We write the true score and comparison matrix w and C, while the observations are written $\hat{w}$ and $\hat{C}$ respectively. This means in particular that $\hat{C}_{i,j} = -C_{i,j} = 1$ and $\hat{C}_{j,i} = -C_{j,i} = -1$. To simplify notations we denote by S the similarity matrix S^{match} (respectively $\hat{S}$ when the similarity is computed from observations). We first study the impact of a corrupted comparison $C_{i,j}$ for $i < j$ on the point score vector $\hat{w}$. We have

$$\hat{w}_i = \sum_{k=1}^{n} \hat{C}_{k,i} = \sum_{k=1}^{n} C_{k,i} + \hat{C}_{j,i} - C_{j,i} = w_i - 2 = w_{i+1},$$

similarly $\hat{w}_j = w_{j-1}$, whereas $\hat{w}_k = w_k$ for $k \neq i, j$. Hence, the incorrect comparison induces two ties in the point score vector w. Now we show that the similarity matrix defined in (4.3) breaks these ties, by showing that it is a strict R-matrix. Writing $\hat{S}$ in terms of S, we get for any $t \neq i, j$

$$[\hat{C}\hat{C}^T]_{i,t} = \sum_{k \neq j} \left(\hat{C}_{i,k}\hat{C}_{t,k}\right) + \hat{C}_{i,j}\hat{C}_{t,j} = \sum_{k \neq j} (C_{i,k}C_{t,k}) + \hat{C}_{i,j}C_{t,j} = \begin{cases} [CC^T]_{i,t} - 2 & \text{if } t < j \\ \left[CC^T\right]_{i,t} + 2 & \text{if } t > j. \end{cases}$$

Thus we obtain

$$\hat{S}_{i,t} = \begin{cases} S_{i,t} - 1 & \text{if } t < j \\ S_{i,t} + 1 & \text{if } t > j, \end{cases}$$

(remember there is a factor $1/2$ in the definition of S). Similarly we get for any $t \neq i, j$

$$\hat{S}_{j,t} = \begin{cases} S_{j,t} + 1 & \text{if } t < i \\ S_{j,t} - 1 & \text{if } t > i. \end{cases}$$

Finally, for the single corrupted index pair (i, j), we get

$$\hat{S}_{i,j} = \frac{1}{2}\left(n + \sum_{k \neq i,j} \left(\hat{C}_{i,k}\hat{C}_{j,k}\right) + \hat{C}_{i,i}\hat{C}_{j,i} + \hat{C}_{i,j}\hat{C}_{j,j}\right) = S_{i,j} - 1 + 1 = S_{i,j}.$$

The diagonal of S is not impacted since $[\hat{C}\hat{C}^T]_{i,i} = \sum_{k=1}^{n} \left(\hat{C}_{i,k}\hat{C}_{i,k}\right) = n$. For all other coefficients (s,t) such that $s,t \neq i,j$, we also have $\hat{S}_{s,t} = S_{s,t}$, which means that all rows or columns outside of i,j are left unchanged. We first observe that these last equations, together with our assumption that $j - i > 2$ and the fact that the elements of the exact S in (4.4) differ by at least one, imply that

$$\hat{S}_{s,t} \leq \hat{S}_{s+1,t} \quad \text{and} \quad \hat{S}_{s,t+1} \leq \hat{S}_{s,t}, \quad \text{for } s < t$$

so $\hat{S}$ remains an R-matrix. Note that this result remains true even when $j - i = 2$, but we need some strict inequalities to show uniqueness of the retrieved order. Indeed, because $j - i > 2$ all these R constraints are strict except between elements of rows i and $i+1$, and rows $j-1$ and j (and similarly for columns). These ties can be broken using the fact that

$$\hat{S}_{i,j-1} = S_{i,j-1} - 1 < S_{i+1,j-1} - 1 = \hat{S}_{i+1,j-1} - 1 < \hat{S}_{i+1,j-1}$$

which means that $\hat{S}$ is still a strict R-matrix (see Figure 4.1) since $j - 1 > i + 1$ by assumption. ■

We now extend this result to multiple errors.

Proposition 4.13. *Given all pairwise comparisons $C_{s,t} \in \{-1,1\}$ between items ranked according to their indices, suppose the signs of m comparisons indexed $(i_1,j_1),\ldots,(i_m,j_m)$ are switched. If the following condition* (4.7) *holds true,*

$$|s-t| > 2, \text{ for all } s,t \in \{i_1,\ldots,i_m,j_1,\ldots,j_m\} \text{ with } s \neq t, \tag{4.7}$$

then S^{match} defined in (4.3) *remains* strict-R*, whereas the point score vector w has $2m$ ties.*

Proof. We write the true score and comparison matrix w and C, while the observations are written $\hat{w}$ and $\hat{C}$ respectively, and without loss of generality we suppose $i_l < j_l$. This implies that $\hat{C}_{i_l,j_l} = -C_{i_l,j_l} = 1$ and $\hat{C}_{j_l,i_l} = -C_{j_l,i_l} = -1$ for all l in $\{1,\ldots,m\}$. To simplify notations, we denote by S the similarity matrix S^{match} (respectively $\hat{S}$ when the similarity is computed from observations).

As in the proof of Proposition 4.12, corrupted comparisons indexed (i_l,j_l) induce shifts of ± 1 on columns and rows i_l and j_l of the similarity matrix S^{match}, while $S^{\text{match}}_{i_l,j_l}$ values remain the same. Since there are several corrupted comparisons, we also need to check the values of $\hat{S}$ at the intersections of rows and columns with indices of corrupted comparisons. Formally, for any

$(i,j) \in \{(i_1,j_1),\dots (i_m,j_m)\}$ and $t \notin \{i_1,\dots,i_m,j_1,\dots,j_m\}$

$$\hat{S}_{i,t} = \begin{cases} S_{i,t}+1 & \text{if } t<j \\ S_{i,t}-1 & \text{if } t>j, \end{cases}$$

Similarly for $t \notin \{i_1,\dots,i_m,j_1,\dots,j_m\}$

$$\hat{S}_{j,t} = \begin{cases} S_{j,t}-1 & \text{if } t<i \\ S_{j,t}+1 & \text{if } t>i. \end{cases}$$

Let (s,s') and $(t,t') \in \{(i_1,j_1),\dots (i_m,j_m)\}$, we have

$$\begin{aligned} \hat{S}_{s,t} &= \tfrac{1}{2}\left(n + \textstyle\sum_{k\neq s',t'}\left(\hat{C}_{s,k}\hat{C}_{t,k}\right) + \hat{C}_{s,s'}\hat{C}_{t,s'} + \hat{C}_{s,t'}\hat{C}_{t,t'}\right) \\ &= \tfrac{1}{2}\left(n + \textstyle\sum_{k\neq s',t'}\left(C_{s,k}C_{t,k}\right) - C_{s,s'}C_{t,s'} - C_{s,t'}C_{t,t'}\right) \end{aligned}$$

Without loss of generality we suppose $s<t$, and since $s<s'$ and $t<t'$, we obtain

$$\hat{S}_{s,t} = \begin{cases} S_{s,t} & \text{if } t>s' \\ S_{s,t}+2 & \text{if } t<s'. \end{cases}$$

Similar results apply for other intersections of rows and columns with indices of corrupted comparisons (i.e., shifts of 0, $+2$, or -2). For all other coefficients (s,t) such that $s,t \notin \{i_1,\dots,i_m,j_1,\dots,j_m\}$, we have $\hat{S}_{s,t} = S_{s,t}$. We first observe that these last equations, together with our assumption that $j_l - i_l > 2$, mean that

$$\hat{S}_{s,t} \geq \hat{S}_{s+1,t} \quad \text{and} \quad \hat{S}_{s,t+1} \geq \hat{S}_{s,t}, \quad \text{for any } s<t$$

so $\hat{S}$ remains an R-matrix. Moreover, since $j_l - i_l > 2$ all these R constraints are strict except between elements of rows i_l and i_l+1, and rows j_l-1 and j_l (similar for columns). These ties can be broken using the fact that for $k = j_l - 1$

$$\hat{S}_{i_l,k} = S_{i_l,k} - 1 < S_{i_l+1,k} - 1 = \hat{S}_{i_l+1,k} - 1 < \hat{S}_{i_l+1,k}$$

which means that $\hat{S}$ is still a strict R-matrix since $k = j_l - 1 > i_l + 1$. Moreover, using the same argument as in the proof of Proposition 4.12, corrupted comparisons induces $2m$ ties in the point score vector w. ■

For the case of one corrupted comparison, note that the separation condition on the pair of items (i,j) is necessary. When the comparison $C_{i,j}$ between two adjacent items is corrupted, no ranking method can break the resulting tie. For the case of arbitrary number of corrupted comparisons, condition (4.7) is a sufficient condition only. We study exact ranking recovery

conditions with missing comparisons in the Appendix, using similar arguments. We now estimate the number of randomly corrupted entries that can be tolerated while maintaining exact recovery of the true ranking.

Proposition 4.14. *Given a comparison matrix for a set of n items with m corrupted comparisons selected uniformly at random from the set of all possible item pairs. Algorithm SerialRank guarantees that the probability of recovery $p(n,m)$ satisfies $p(n,m) \geq 1-\delta$, provided that $m = O(\sqrt{\delta n})$. In particular, this implies that $p(n,m) = 1 - o(1)$ provided that $m = o(\sqrt{n})$.*

Proof. Let $\mathcal{P}$ be the set of all distinct pairs of items from the set $\{1, 2, \ldots, n\}$. Let $\mathcal{X}$ be the set of all admissible sets of pairs of items, i.e. containing each $X \subseteq \mathcal{P}$ such that X satisfies condition (4.7). We consider the case of $m \geq 1$ distinct pairs of items sampled from the set $\mathcal{P}$ uniformly at random without replacement. Let X_i denote the set of sampled pairs given that i pairs are sampled. We seek to bound $p(n,m) = \mathbf{Prob}(X_m \in \mathcal{X})$. Given a set of pairs $X \in \mathcal{X}$, let $T(X)$ be the set of non-admissible pairs, i.e. containing $(i,j) \in \mathcal{P} \setminus X$ such that $X \cup (i,j) \notin \mathcal{X}$. We have

$$\mathbf{Prob}(X_m \in \mathcal{X}) = \sum_{x \in \mathcal{X} : |x| = m-1} \left(1 - \frac{|T(x)|}{|\mathcal{P}| - (m-1)}\right) \mathbf{Prob}(X_{m-1} = x). \tag{4.8}$$

Note that every selected pair from $\mathcal{P}$ contributes at most $15n$ non-admissible pairs. Indeed, given a selected pair (i,j), a non-admissible pair (s,t) should respect one of the following conditions $|s-i| \leq 2$, $|s-j| \leq 2$, $|t-i| \leq 2$, $|t-j| \leq 2$ or $|s-t| \leq 2$. Given any item s, there are 15 possible choice of t to output a non-admissible pair (s,t), resulting in at most $15n$ non-admissible pairs for the selected pair (i,j).

Hence, for every $x \in \mathcal{X}$ we have

$$|T(x)| \leq 15n|x|.$$

Combining this with (4.8) and the fact that $|\mathcal{P}| = \binom{n}{2}$, we have

$$\mathbf{Prob}(X_m \in \mathcal{X}) \geq \left(1 - \frac{15n}{\binom{n}{2} - (m-1)}(m-1)\right) \mathbf{Prob}(X_{m-1} \in \mathcal{X}).$$

From this it follows

$$\begin{aligned} p(n,m) &\geq \prod_{i=1}^{m-1} \left(1 - \frac{15n}{\binom{n}{2} - (i-1)} i\right) \\ &\geq \prod_{i=1}^{m-1} \left(1 - \frac{i}{a(n,m)}\right) \end{aligned}$$

where

$$a(n,m) = \frac{\binom{n}{2} - (m-1)}{15n}.$$

Notice that when $m = o(n)$ we have $\left(1 - \frac{i}{a(n,m)}\right) \sim \exp(-30i/n)$ and

$$\prod_{i=1}^{m-1} \left(1 - \frac{i}{a(n,m)}\right) \sim \prod_{i=1}^{m-1} \exp(-30i/n) \sim \exp\left(-\frac{15m^2}{n}\right) \text{ for large } n.$$

Hence, given $\delta > 0$, $p(n,m) \geq 1 - \delta$ provided that $m = O(\sqrt{n\delta})$. If $\delta = o(1)$, the condition is $m = o(\sqrt{n})$. ■

4.5 Spectral Perturbation Analysis

In this section we analyze how SerialRank performs when only a small fraction of pairwise comparisons are given. We show that for Erdös-Rényi graphs, i.e., when pairwise comparisons are observed independently with a given probability, $\Omega(n \log^4 n)$ comparisons suffice for ℓ_2 consistency of the Fiedler vector and hence ℓ_2 consistency of the retreived ranking w.h.p. On the other hand we need $\Omega(n^{3/2} \log^4 n)$ comparisons to retrieve a ranking whose local perturbations are bounded in ℓ_∞ norm. Since Erdös-Rényi graphs are connected with high probability only when the total number of pairs sampled scales as $\Omega(n \log n)$, we need at least that many comparisons in order to retrieve a ranking, therefore the ℓ_2 consistency result can be seen as optimal up to a polylogarithmic factor.

Our bounds are mostly related to the work of (Wauthier et al., 2013). In its simplified version (Theorem 4.2 Wauthier et al., 2013) shows that when ranking items according to their point score, for any precision parameter $\mu \in (0, 1)$, sampling independently with fixed probability $\Omega\left(\frac{n \log n}{\mu^2}\right)$ comparisons guarantees that the maximum displacement between the retrieved ranking and the true ranking, i.e., the ℓ_∞ distance to the true ranking, is bounded by μn with high probability for n large enough.

Sample complexity bounds have also been studied for the Rank Centrality algorithm (Dwork et al., 2001a; Negahban et al., 2012). In their analysis, (Negahban et al., 2012) suppose that some pairs are sampled independently with fixed probability, and then k comparisons are generated for each sampled pair, under a Bradley-Terry-Luce model (BTL). When ranking items according to the stationary distribution of a transition matrix estimated from comparisons, sampling $\Omega(n \cdot \text{polylog}(n))$ pairs are enough to bound the relative ℓ_2 norm perturbation of the stationary distribution. However, as pointed out by (Wauthier et al., 2013), repeated measurements are not practical, e.g., if comparisons are derived from the outcomes of sports games or the purchasing behavior of a customer (a customer typically wants to purchase a product only once). Moreover, (Negahban et al., 2012) do not provide bounds on the relative ℓ_∞ norm perturbation of the ranking.

We also refer the reader to the recent work of Rajkumar and Agarwal (2014), who provide a survey of sample complexity bounds for Rank Centrality, maximum likelihood estimation, least-square ranking and an SVM based ranking, under a more flexible sampling model. However, those bounds only give the sampling complexity for exact recovery of ranking, which is usually prohibitive when n is large, and are more difficult to interpret.

Finally, we refer the interested reader to (Huang et al., 2008; Shamir and Tishby, 2011) for sampling complexity bounds in the context of spectral clustering.

Limitations. We emphasize that sampling models based on Erdös-Rényi graphs are not the most realistic, though they have been studied widely in the literature (see for instance Feige et al., 1994; Braverman and Mossel, 2008; Wauthier et al., 2013). Indeed, pairs are not likely to be sampled independently. For instance, when ranking movies, popular movies in the top ranks are more likely to be compared. Corrupted comparisons are also more likely between items that have close rankings. We hope to extend our perturbation analysis to more general models in future work.

A second limitation of our perturbation analysis comes from the setting of ordinal comparisons, i.e., binary comparisons, since in many applications, several comparisons are provided for each sampled pair. Nevertheless, the setting of ordinal comparisons is interesting for the analysis of SerialRank, since numerical experiments suggest that it is the setting for which SerialRank provides the best results compared to other methods. Note that in practice, we can easily get rid of this limitation (see Section 4.2.2.2 and 4.6). We refer the reader to numerical experiments in Section 4.6, as well as a recent paper by Cucuringu (2015), which introduces another ranking algorithm called SyncRank, and provides extensive numerical experiments on state-of-the-art ranking algorithms, including SerialRank.

Choice of Laplacian: normalized vs. unnormalized. In the spectral clustering literature, several constructions for the Laplacian operators are suggested, namely the unnormalized Laplacian (used in SerialRank), the symmetric normalized Laplacian, and the non-symmetric normalized Laplacian. Von Luxburg et al. (2008) show stronger consistency results for spectral clustering by using the non-symmetric normalized Laplacian. Here, we show that the Fiedler vector of the normalized Laplacian is an affine function of the ranking, hence sorting the Fiedler vector still guarantees exact recovery of the ranking, when all comparisons are observed and consistent with a global ranking. In contrast, we only get an asymptotic expression for the unnormalized Laplacian (*cf.* section 4.8). This motivated us to provide an analysis of SerialRank robustness based on the normalized Laplacian, though in practice the use of the unnormalized Laplacian is valid and seems to give better results (*cf.* Figures 4.2 and 4.5).

Notations. Throughout this section, we only focus on the similarity S^{match} in (4.3) and write it S to simplify notations. W.l.o.g. we assume in the following that the true ranking is the identity,

hence S is an R-matrix. We write $\|\cdot\|_2$ the operator norm of a matrix, which corresponds to the maximal absolute eigenvalue for symmetric matrices. $\|\cdot\|_F$ denotes the Frobenius norm. We refer to the eigenvalues of the Laplacian as λ_i, with $\lambda_1 = 0 \leq \lambda_2 \leq \ldots \leq \lambda_n$. For any quantity x, we denote by $\tilde{x}$ its perturbed analogue. We define the residual matrix $R = \tilde{S} - S$ and write f the normalized Fiedler vector of the Laplacian matrix L_S. We define the degree matrix $D_S = \mathbf{diag}(D\mathbf{1})$ the diagonal matrix whose elements are the row-sums of matrix S. Whenever we use the abreviation w.h.p., this means that the inequality is true with probability greater than $1 - 2/n$. Finally, we will use $c > 0$ for absolute constants, whose values are allowed to vary from one equation to another.

We assume that our information on preferences is both incomplete and corrupted. Specifically, pairwise comparisons are independently sampled with probability q and these sampled comparisons are consistent with the underlying total ranking with probability p. Let us define $\tilde{C} = B \circ C$ the matrix of observed comparisons, where C is the true comparison matrix defined in (4.1), $\circ$ is the Hadamard product and B is a symmetric matrix with entries

$$B_{i,j} = \begin{cases} 0 & \text{with probability } 1-q \\ 1 & \text{with probability } qp \\ -1 & \text{with probability } q(1-p). \end{cases}$$

In order to obtain an unbiased estimator of the comparison matrix defined in (4.1), we normalize $\tilde{C}$ by its mean value $q(2p-1)$ and redefine $\tilde{S}$ as

$$\tilde{S} = \frac{1}{q^2(2p-1)^2}\tilde{C}\tilde{C}^T + n\mathbf{1}\mathbf{1}^T.$$

For ease of notations we have dropped the factor $1/2$ in (4.3) w.l.o.g. (positive multiplicative factors of the Laplacian do not affect its eigenvectors).

4.5.1 Results

We now state our main results. The first one bounds ℓ_2 perturbations of the Fiedler vector f with both missing and corrupted comparisons. Note that f and $\tilde{f}$ are normalized.

Theorem 4.21. *For every $\mu \in (0,1)$ and n large enough, if $q > \frac{\log^4 n}{\mu^2(2p-1)^4 n}$, then*

$$\|\tilde{f} - f\|_2 \leq c\frac{\mu}{\sqrt{\log n}}$$

with probability at least $1 - 2/n$, where $c > 0$ is an absolute constant.

As n goes to infinity the perturbation of the Fiedler vector goes to zero, and we can retrieve the "true" ranking by reordering the Fiedler vector. Hence this bounds provides ℓ_2 consistency of the ranking, with an optimal sampling complexity (up to a polylogarithmic factor).

The second result bounds local perturbations of the ranking with π referring to the "true" ranking and $\tilde{\pi}$ to the ranking retrieved by SerialRank.

Theorem 4.24. *For every $\mu \in (0,1)$ and n large enough, if $q > \frac{\log^4 n}{\mu^2(2p-1)^4\sqrt{n}}$, then*

$$\|\tilde{\pi} - \pi\|_\infty \leq c\mu n$$

with probability at least $1 - 2/n$, where $c > 0$ is an absolute constant.

This bound quantifies the maximum displacement of any item's ranking. μ can be seen a "precision" parameter. For instance, if we set $\mu = 0.1$, Theorem 4.24 means that we can expect the maximum displacement of any item's ranking to be less than $0.1 \cdot n$ when observing $c^2 \cdot 100 \cdot n\sqrt{n} \cdot \log^4 n$ comparisons (with $p = 1$).

We conjecture Theorem 4.24 still holds true if the condition $q > \log^4 n/\mu^2(2p-1)^4\sqrt{n}$ is replaced by the weaker condition $q > \log^4 n/\mu^2(2p-1)^4 n$.

4.5.2 Sketch of the proof.

The proof of these results relies on classical perturbation arguments and is structured as follows.

- **Step 1:** Bound $\|\tilde{D}_S - D_S\|_2$, $\|\tilde{S} - S\|_2$ with high probability using concentration inequalities on quadratic forms of Bernoulli variables and results from (Achlioptas and McSherry, 2007).
- **Step 2.** Show that the normalized Laplacian $L = \mathbf{I} - D^{-1}S$ has a linear Fiedler vector and bound the eigengap between the Fiedler value and other eigenvalues.
- **Step 3.** Bound $\|\tilde{f} - f\|_2$ using Davis-Kahan theorem and bounds of steps 1 and 2.
- **Step 4.** Use the linearity of the Fiedler vector to translate this result into a bound on the maximum displacement of the retrieved ranking $\|\tilde{\pi} - \pi\|_\infty$.

We now turn to the proof itself.

4.5.3 Step 1: Bounding $\|\tilde{D}_S - D_S\|_2$ and $\|\tilde{S} - S\|_2$

Here, we seek to bound $\|\tilde{D}_S - D_S\|_2$ and $\|\tilde{S} - S\|_2$ with high probability using concentration inequalities.

4.5.3.1 Bounding the norm of the degree matrix

We first bound perturbations of the degree matrix with both missing and corrupted comparisons.

Lemma 4.15. *For every $\mu \in (0,1)$ and $n \geq 100$, if $q \geq \frac{\log^4 n}{\mu^2(2p-1)^4 n}$ then*

$$\|\tilde{D}_S - D_S\|_2 \leq \frac{3\mu n^2}{\sqrt{\log n}}$$

with probability at least $1 - 1/n$.

Proof. Let $R = \tilde{S} - S$ and $\delta = \mathbf{diag}\, D_R = \mathbf{diag}((\tilde{S} - S)\mathbf{1})$. Since D_S and $\tilde{D}_S$ are diagonal matrices, $\|\tilde{D}_S - D_S\|_2 = \max|\delta_i|$. We first seek a concentration inequality for each δ_i and then derive a bound on $\|\tilde{D}_S - D_S\|_2$.

By definition of the similarity matrix S and its perturbed analogue $\tilde{S}$ we have

$$R_{ij} = \sum_{k=1}^{n} C_{ik}C_{jk}\left(\frac{B_{ik}B_{jk}}{q^2(2p-1)^2} - 1\right).$$

Hence

$$\delta_i = \sum_{j=1}^{n} R_{ij} = \sum_{j=1}^{n}\sum_{k=1}^{n} C_{ik}C_{jk}\left(\frac{B_{ik}B_{jk}}{q^2(2p-1)^2} - 1\right).$$

Notice that we can arbitrarily fix the diagonal values of R to zeros. Indeed, the similarity between an element and itself should be a constant by convention, which leads to $R_{ii} = \tilde{S}_{ii} - S_{ii} = 0$ for all items i. Hence we could take $j \neq i$ in the definition of δ_i, and we can consider B_{ik} independent of B_{jk} in the associated summation.

We first seek a concentration inequality for each δ_i. Notice that

$$\begin{aligned}
\delta_i &= \sum_{j=1}^{n}\sum_{k=1}^{n} C_{ik}C_{jk}\left(\frac{B_{ik}B_{jk}}{q^2(2p-1)^2} - 1\right) \\
&= \underbrace{\sum_{k=1}^{n}\left(\frac{C_{ik}B_{ik}}{q(2p-1)}\sum_{j=1}^{n} C_{jk}\left(\frac{B_{jk}}{q(2p-1)} - 1\right)\right)}_{\text{Quad}} + \underbrace{\sum_{k=1}^{n}\sum_{j=1}^{n} C_{ik}C_{jk}\left(\frac{B_{ik}}{q(2p-1)} - 1\right)}_{\text{Lin}}.
\end{aligned}$$

The first term (denoted Quad in the following) is quadratic with respect to the B_{jk} while the second term (denoted Lin in the following) is linear. Both terms have mean zero since the B_{ik} are independent of the B_{jk}. We begin by bounding the quadratic term Quad. Let $X_{jk} = C_{jk}\left(\frac{1}{q(2p-1)}B_{jk} - 1\right)$. We have

$$\mathbf{E}(X_{jk}) = C_{jk}\left(\tfrac{qp-q(1-p)}{q(2p-1)} - 1\right) = 0,$$

$$\mathbf{var}(X_{jk}) = \frac{\mathbf{var}(B_{jk})}{q^2(2p-1)^2} = \frac{1}{q^2(2p-1)^2}(q - q^2(2p-1)^2) = \frac{1}{q(2p-1)^2} - 1 \leq \frac{1}{q(2p-1)^2},$$

and

$$|X_{jk}| = \left|\frac{B_{jk}}{q(2p-1)} - 1\right| \leq 1 + \frac{1}{q(2p-1)} \leq \frac{2}{q(2p-1)} \leq \frac{2}{q(2p-1)^2}.$$

By applying Bernstein's inequality for any $t > 0$

$$\mathbf{Prob}\left(\left|\sum_{j=1}^{n} X_{jk}\right| > t\right) \leq 2\exp\left(\frac{-q(2p-1)^2t^2}{2(n+2t/3)}\right) \leq 2\exp\left(\frac{-q(2p-1)^2t^2}{2(n+t)}\right). \quad (4.9)$$

Now notice that

$$\begin{aligned}\mathbf{Prob}(|\text{Quad}| > t) &= \mathbf{Prob}\left(\left|\sum_{k=1}^{n}\left(C_{ik}\frac{B_{ik}}{q(2p-1)}\sum_{j=1}^{n} X_{jk}\right)\right| > t\right) \\ &\leq \mathbf{Prob}\left(\sum_{k=1}^{n}\left(\frac{|B_{ik}|}{q(2p-1)}\right)\max_l\left|\sum_{j=1}^{n} X_{jl}\right| > t\right).\end{aligned}$$

By applying a union bound to the first Bernstein inequality (4.9), for any $t > 0$

$$\mathbf{Prob}\left(\max_l\left|\sum_{j=1}^{n} X_{jl}\right| > \sqrt{t}\right) \leq 2n\exp\left(\frac{-tq(2p-1)^2}{2(n+\sqrt{t})}\right).$$

Moreover, since $\mathbf{E}\,|B_{ik}| = q$ we also get from Bernstein's inequality that for any $t > 0$

$$\mathbf{Prob}\left(\sum_{k=1}^{n}\frac{|B_{ik}|}{q(2p-1)} > \frac{n}{2p-1} + \sqrt{t}\right) \leq \exp\left(\frac{-tq(2p-1)^2}{2(n+\sqrt{t})}\right).$$

We deduce from these last three inequalities that for any $t > 0$

$$\mathbf{Prob}(|\text{Quad}| > t) \leq (2n+1)\exp\left(\frac{-tq(2p-1)^2}{2(n+\sqrt{t})}\right).$$

Taking $t = \mu^2(2p-1)^2n^2/\log n$ and $q \geq \frac{\log^4 n}{\mu^2(2p-1)^4 n}$, with $\mu \leq 1$, we have $\sqrt{t} \leq n$ and we deduce that

$$\mathbf{Prob}\left(|\text{Quad}| > \frac{2\mu n^2}{\sqrt{\log n}}\right) \leq (2n+1)\exp\left(-\frac{\log^3 n}{4}\right). \quad (4.10)$$

We now bound the linear term Lin.

$$\begin{aligned}
\mathbf{Prob}(|\text{Lin}| > t) &= \mathbf{Prob}\left(\left|\sum_{j=1}^{n}\sum_{k=1}^{n} C_{ik}C_{jk}\left(\frac{B_{ik}}{q(2p-1)} - 1\right)\right| > t\right) \\
&\leq \mathbf{Prob}\left(\sum_{k=1}^{n} |C_{ik}| \max_{l} |\sum_{j=1}^{n} X_{jl}| > t\right) \\
&\leq \mathbf{Prob}\left(\max_{k} |\sum_{j=1}^{n} X_{jk}| > t/n\right),
\end{aligned}$$

hence

$$\mathbf{Prob}(|\text{Lin}| > t) \leq 2n \exp\left(\frac{-t^2 q(2p-1)^2}{2n^2(n + t/n)}\right).$$

Taking $t = \mu n^2/(\log n)^{1/2}$ and $q \geq \frac{\log^4 n}{\mu^2(2p-1)^4 n}$, with $\mu \leq 1$, we have $t \leq n^2$ and we deduce that

$$\mathbf{Prob}(|\text{Lin}| > t) \leq 2n \exp\left(-\frac{\log^3 n}{4}\right). \tag{4.11}$$

Finally, combining equations (4.10) and (4.11), we obtain for $q \geq \frac{\log^4 n}{\mu^2(2p-1)^4 n}$, with $\mu \leq 1$

$$\mathbf{Prob}\left(|\delta_i| > \frac{3\mu n^2}{\sqrt{\log n}}\right) \leq (4n+1) \exp\left(-\frac{\log^3 n}{4}\right).$$

Now, using a union bound, this shows that for $q \geq \frac{\log^4 n}{\mu^2(2p-1)^4 n}$,

$$\mathbf{Prob}\left(\max |\delta_i| > \frac{3\mu n^2}{\sqrt{\log n}}\right) \leq n(4n+1) \exp\left(-\frac{\log^3 n}{4}\right),$$

which is less than $1/n$ for $n \geq 100$. ■

4.5.3.2 Bounding perturbations of the comparison matrix C

Here, we adapt results in (Achlioptas and McSherry, 2007) to bound perturbations of the comparison matrix. We will then use bounds on the perturbations of C to bound $\|\tilde{S} - S\|_2$.

Lemma 4.16. *For $n \geq 104$ and $q \geq \frac{\log^3 n}{n}$,*

$$\|C - \tilde{C}\|_2 \leq \frac{c}{2p-1}\sqrt{\frac{n}{q}}, \tag{4.12}$$

with probability at least $1 - 2/n$, where c is an absolute constant.

Proof. The main argument of the proof is to use the independence of the C_{ij} for $i < j$ in order to bound $\|\tilde{C} - C\|_2$ by a constant times $\sigma\sqrt{n}$, where σ is the standard deviation of C_{ij}. To

isolate independent entries in the perturbation matrix, we first need to break the anti-symmetry of $\tilde{C}-C$ by decomposing $X=\tilde{C}-C$ into its upper triangular part and its lower triangular part, i.e., $\tilde{C}-C=X_{\text{up}}+X_{\text{low}}$, with $X_{\text{up}}=-X_{\text{low}}^T$ (diagonal entries of $\tilde{C}-C$ can be arbitrarily set to 0). Entries of X_{up} are all independent, with variance less than the variance of $\tilde{C}_{ij}$. Indeed, lower entries of X_{up} are equal to 0 and hence have variance 0. Notice that

$$\|\tilde{C}-C\|_2=\|X_{\text{up}}+X_{\text{low}}\|_2\leq\|X_{\text{up}}\|_2+\|X_{\text{low}}\|_2\leq 2\|X_{\text{up}}\|_2,$$

so bounding $\|X_{\text{up}}\|_2$ will give us a bound on $\|X\|_2$. In the rest of the proof we write X_{up} instead of X to simplify notations. We can now apply (Achlioptas and McSherry, 2007, Th. 3.1) to X. Since

$$X_{ij}=\tilde{C}_{ij}-C_{ij}=C_{ij}\left(\frac{B_{ij}}{q(2p-1)}-1\right),$$

we have (*cf.* proof of Lemma 4.15) $\mathbf{E}(X_{ij})=0$, $\mathbf{var}(X_{ij})\leq\frac{1}{q(2p-1)^2}$, and $|X_{ij}|\leq\frac{2}{q(2p-1)}$. Hence for a given $\epsilon>0$ such that

$$\frac{4}{q(2p-1)}\leq\left(\frac{\log(1+\epsilon)}{2\log(2n)}\right)^2\frac{\sqrt{2n}}{\sqrt{q}(2p-1)}, \tag{4.13}$$

for any $\theta>0$ and $n\geq 76$,

$$\mathbf{Prob}\left(\|X\|_2\geq 2(1+\epsilon+\theta)\frac{1}{\sqrt{q}(2p-1)}\sqrt{2n}\right)<2\exp\left(-16\frac{\theta^2}{\epsilon^4}\log^3 n\right). \tag{4.14}$$

For $q\geq\frac{(\log 2n)^3}{n}$ and taking $\epsilon\geq\exp(\sqrt(16/\sqrt(2)))-1$ (so $\log(1+\epsilon)^2\geq 16/\sqrt{2}$) means inequality (4.13) holds. Taking (4.14) with $\epsilon=30$ and $\theta=30$ we get

$$\mathbf{Prob}\left(\|X\|_2\geq\frac{2\sqrt{2}(1+30+30)}{2p-1}\sqrt{\frac{n}{q}}\right)<2\exp\left(-10^{-2}\log^3 n\right). \tag{4.15}$$

Hence for $n\geq 104$, we have $\log^3 n>100$ and

$$\mathbf{Prob}\left(\|X\|_2\geq\frac{112}{2p-1}\sqrt{\frac{n}{q}}\right)<2/n.$$

Noting that $\log 2n\leq 1.15\log n$ for $n\geq 104$, we obtain the desired result by choosing $c=2\times 112\times\sqrt{1.15}\leq 241$. ■

4.5.3.3 Bounding the perturbation of the similarity matrix $\|S\|$.

We now seek to bound $\|\tilde{S}-S\|$ with high probability.

Lemma 4.17. *For every $\mu \in (0,1)$, $n \geq 104$, if $q > \frac{\log^4 n}{\mu^2(2p-1)^2 n}$, then*

$$\|\tilde{S} - S\|_2 \leq c\frac{\mu n^2}{\sqrt{\log n}},$$

with probability at least $1 - 2/n$, where c is an absolute constant.

Proof. Let $X = \tilde{C} - C$. We have

$$\tilde{C}\tilde{C}^T = (C+X)(C+X)^T = CC^T + XX^T + XC^T + CX^T,$$

hence

$$\tilde{S} - S = XX^T + XC^T + CX^T,$$

and

$$\|\tilde{S} - S\|_2 \leq \|XX^T\|_2 + \|XC^T\|_2 + \|CX^T\|_2 \leq \|X\|_2^2 + 2\|X\|_2\|C\|_2.$$

From Lemma 4.16 we deduce that for $n \geq 104$ and $q \geq \frac{\log^4 n}{n}$, with probability at least $1 - 2/n$

$$\|\tilde{S} - S\|_2 \leq \frac{c^2 n}{q(2p-1)^2} + \frac{2c}{2p-1}\sqrt{\frac{n}{q}}\|C\|_2. \tag{4.16}$$

Notice that $\|C\|_2^2 \leq \mathbf{Tr}(CC^T) = n^2$, hence $\|C\|_2 \leq n$ and

$$\|\tilde{S} - S\|_2 \leq \frac{c^2 n}{q(2p-1)^2} + \frac{2cn}{2p-1}\sqrt{\frac{n}{q}}. \tag{4.17}$$

By taking $q > \frac{\log^4 n}{\mu^2(2p-1)^2 n}$, we get for $n \geq 104$ with probability at least $1 - 2/n$

$$\|\tilde{S} - S\|_2 \leq \frac{c^2\mu^2 n^2}{\log^4 n} + \frac{2c\mu n^2}{\log^2 n}.$$

Hence setting a new constant c with $c = \max(c^2(\log 104)^{-7/2}, 2c(\log 104)^{-3/2}) \leq 270$,

$$\|\tilde{S} - S\|_2 \leq c\frac{\mu n^2}{\sqrt{\log n}}$$

with probability at least $1 - 2/n$, which is the desired result. ∎

4.5.4 Step 2: Controlling the eigengap

In the following proposition we show that the normalized Laplacian of the similarity matrix S has a constant Fiedler value and a linear Fiedler vector. We then deduce bounds on the eigengap between the first, second and third smallest eigenvalues of the Laplacian.

Proposition 4.18. *Let $L^{\text{norm}} = \mathbf{I} - D^{-1}S$ be the non-symmetric normalized Laplacian of S. L^{norm} has a linear Fiedler vector, and its Fiedler value is equal to $2/3$.*

Proof. Let $x_i = i - \frac{n+1}{2}$ (x is linear with mean zero). We want to show that $L^{\text{norm}}x = \lambda_2 x$ or equivalently $Sx = (1-\lambda_2)Dx$. We develop both sides of the last equation, and use the following facts

$$S_{i,j} = n - |j-i|, \quad \sum_{k=1}^{n} k = \frac{n(n+1)}{2}, \quad \sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}.$$

We first get an expression for the degree of S, defined by $d = S\mathbf{1} = \sum_{i=1}^{n} S_{i,k}$, with

$$\begin{aligned} d_i &= \sum_{k=1}^{i-1} S_{i,k} + \sum_{k=i}^{n} S_{i,k} \\ &= \sum_{k=1}^{i-1}(n-i+k) + \sum_{k=i}^{n}(n-k+i) \\ &= \frac{n(n-1)}{2} + i(n-i+1). \end{aligned}$$

Similarly we have

$$\begin{aligned} \sum_{k=1}^{n} kS_{i,k} &= \sum_{k=1}^{i-1} k(n-i+k) + \sum_{k=i}^{n} k(n-k+i) \\ &= \frac{n^2(n+1)}{2} + \frac{i(i-1)(2i-1)}{3} - \frac{n(n+1)(2n+1)}{6} - i^2(i-1) + i\frac{n(n+1)}{2}. \end{aligned}$$

Finally, setting $\lambda_2 = 2/3$, notice that

$$\begin{aligned} [Sx]_i &= \sum_{k=1}^{n} S_{i,k}\left(k - \frac{n+1}{2}\right) \\ &= \sum_{k=1}^{n} kS_{i,k} - \frac{n+1}{2}d_i \\ &= \frac{1}{3}\left(\frac{n(n-1)}{2} + i(n-i+1)\right)\left(i - \frac{n+1}{2}\right) \\ &= (1-\lambda_2)d_i x_i, \end{aligned}$$

which shows that $Sx = (1-\lambda_2)Dx$. ∎

The next corollary will be useful in following proofs.

Corollary 4.19. *The Fiedler vector f of the unperturbed Laplacian satisfies $\|f\|_\infty \leq 2/\sqrt{n}$.*

Proof. We use the fact that f is collinear to the vector x defined by $x_i = i - \frac{n+1}{2}$ and verifies $\|f\|_2 = 1$. Let us consider the case of n odd. The Fiedler vector verifies $f_i = \frac{i-(n+1)/2}{a_n}$, with

$$a_n^2 = 2\sum_{k=0}^{(n-1)/2} k^2 = \frac{2}{6}\frac{n-1}{2}\left(\frac{n-1}{2}+1\right)((n-1)+1) = \frac{n^3-n}{12}.$$

Hence

$$\|f\|_\infty = f_n = \frac{n-1}{2a_n} \leq \sqrt{\frac{3}{n-1}} \leq \frac{2}{\sqrt{n}} \text{ for } n \geq 5.$$

A similar reasoning applies for n even. ■

Lemma 4.20. *The minimum eigengap between the Fiedler value and other eigenvalues is bounded below by a constant for n sufficiently large.*

Proof. The first eigenvalue of the Laplacian is always 0, so we have for any n, $\lambda_2 - \lambda_1 = \lambda_2 = 2/3$. Moreover, using results from (Von Luxburg et al., 2008), we know that eigenvalues of the normalized Laplacian that are different from one converge to an asymptotic spectrum, and that the limit eigenvalues are "isolated". Hence there exists $n_0 > 0$ and $c > 0$ such that for any $n \geq n_0$ we have $\lambda_3 - \lambda_2 > c$. ■

Numerical experiments show that λ_3 converges to $0.93\ldots$ very fast when n grows towards infinity.

4.5.5 Step 3: Bounding the perturbation of the Fiedler vector $\|\tilde{f} - f\|_2$

We can now compile results from previous sections to get a first perturbation bound and show ℓ_2 consistency of the Fiedler vector when comparisons are both missing and corrupted.

Theorem 4.21. *For every $\mu \in (0,1)$ and n large enough, if $q > \frac{\log^4 n}{\mu^2(2p-1)^4 n}$, then*

$$\|\tilde{f} - f\|_2 \leq c\frac{\mu}{\sqrt{\log n}},$$

with probability at least $1 - 2/n$.

Proof. In order to use Davis-Kahan theorem, we need to relate perturbations of the normalized Laplacian matrix to perturbations of the similarity and degree matrices. To simplify notations, we write $L = \mathbf{I} - D^{-1}S$ and $\tilde{L} = \mathbf{I} - \tilde{D}^{-1}\tilde{S}$.

Since the normalized Laplacian is not symmetric, we will actually apply Davis-Kahan theorem to the symmetric normalized Laplacian $L_{sym} = \mathbf{I} - D^{-1/2}SD^{-1/2}$. It is easy to see that L_{sym} and L have the same Fiedler value, and that the Fiedler vector f_{sym} of L_{sym} is equal to $D^{1/2}f$

(up to normalization). Indeed, if v is the eigenvector associated to the i^{th} eigenvalue of L (denoted by λ_i), then

$$L_{sym}D^{1/2}v = D^{-1/2}(D-S)D^{-1/2}D^{1/2}v = D^{-1/2}(D-S)v = D^{1/2}(\mathbf{I}-D^{-1}S)v = \lambda_i D^{1/2}v.$$

Hence perturbations of the Fiedler vector of L_{sym} are directly related to perturbations of the Fiedler vector of L.

The proof relies mainly on Lemma 4.15, which states that for $n \geq 100$, denoting by d the vector of diagonal elements of D_S,

$$\|D_R\|_2 = \max |\tilde{d}_i - d_i| \leq \frac{3\mu n^2}{\sqrt{\log n}}$$

with probability at least $1 - \frac{2}{n}$. Combined with the fact that $d_i = \frac{n(n-1)}{2} + i(n-i+1)$ (*cf.* proof of Proposition 4.18), this guarantees that d_i and $\tilde{d}_i$ are strictly positive. Hence $D^{-1/2}$ and $\tilde{D}^{-1/2}$ are well defined. We now decompose the perturbation of the Laplacian matrix. Let $\Delta = D^{-1/2}$, we have

$$\begin{aligned}
\|\tilde{L}_{sym} - L_{sym}\|_2 &= \|\tilde{\Delta}\tilde{S}\tilde{\Delta} - \Delta S \Delta\|_2 \\
&= \|\tilde{\Delta}\tilde{S}\tilde{\Delta} - \tilde{\Delta}S\tilde{\Delta} + \tilde{\Delta}S\tilde{\Delta} - \Delta S \Delta\|_2 \\
&= \|\tilde{\Delta}(\tilde{S}-S)\tilde{\Delta} + \tilde{\Delta}S\tilde{\Delta} - \Delta S\tilde{\Delta} + \Delta S\tilde{\Delta} - \Delta S \Delta\|_2 \\
&= \|\tilde{\Delta}(\tilde{S}-S)\tilde{\Delta} + (\tilde{\Delta}-\Delta)S\tilde{\Delta} + \Delta S(\tilde{\Delta}-\Delta)\|_2 \\
&\leq \|\tilde{\Delta}\|_2^2\|\tilde{S}-S\|_2 + \|S\|_2(\|\tilde{\Delta}\|_2 + \|\Delta\|_2)\|\tilde{\Delta}-\Delta\|_2.
\end{aligned}$$

We first bound $\|\tilde{\Delta} - \Delta\|_2$. Notice that

$$\|\tilde{\Delta} - \Delta\|_2 = \max_i |\tilde{d}_i^{-1/2} - d_i^{-1/2}|,$$

where d_i (respectively $\tilde{d}_i$) is the sum of elements of the i^{th} row of S (respectively $\tilde{S}$). Hence

$$\|\tilde{\Delta} - \Delta\|_2 = \max_i \frac{\left|\sqrt{\tilde{d}_i} - \sqrt{d_i}\right|}{\sqrt{\tilde{d}_i d_i}} = \max_i \frac{\left|\tilde{d}_i - d_i\right|}{\sqrt{\tilde{d}_i d_i}(\sqrt{\tilde{d}_i} + \sqrt{d_i})}.$$

Using Lemma 4.15 we obtain

$$\|\tilde{\Delta} - \Delta\|_2 \leq \max_i \frac{\frac{3\mu n^2}{\sqrt{\log n}}}{\sqrt{d_i}(d_i - \frac{3\mu n^2}{\sqrt{\log n}}) + d_i\sqrt{d_i - \frac{3\mu n^2}{\sqrt{\log n}}}}, \quad i = 1, \ldots, n, \text{ w.h.p.}$$

Since $d_i = \frac{n(n-1)}{2} + i(n-i+1)$ (*cf.* proof of Proposition 4.18), for $\mu < 1$ there exists a constant c such that $d_i > d_i - \frac{3\mu n^2}{\sqrt{\log n}} > cn^2$. We deduce that there exists an absolute constant c such that

$$\|\tilde{\Delta} - \Delta\|_2 \leq \frac{c\mu}{n\sqrt{\log n}} \text{ w.h.p.} \tag{4.18}$$

Similarly we obtain that

$$\|\Delta\|_2 \leq \frac{c}{n} \text{ w.h.p,} \tag{4.19}$$

and

$$\|\tilde{\Delta}\|_2 \leq \frac{c}{n} \text{ w.h.p.} \tag{4.20}$$

Moreover, we have

$$\|S\|_2 = \|CC^T + n\mathbf{1}\mathbf{1}^T\|_2 \leq \|C\|_2^2 + n\|\mathbf{1}\mathbf{1}^T\|_2 \leq 2n^2.$$

Hence,

$$\|S\|_2(\|\tilde{\Delta}\|_2 + \|\Delta\|_2)\|\tilde{\Delta} - \Delta\|_2 \leq \frac{c\mu}{\sqrt{\log n}} \text{ w.h.p,}$$

where $c := 4c^2$. Using Lemma 4.17, we can similarly bound $\|\tilde{\Delta}\|_2^2\|\tilde{S} - S\|_2$ and obtain

$$\|\tilde{L}_{sym} - L_{sym}\|_2 \leq \frac{c\mu}{\sqrt{\log n}} \text{ w.h.p,} \tag{4.21}$$

where c is an absolute constant. Finally, for small μ, Weyl's inequality, equation (4.21) together with Lemma 4.20 ensure that for n large enough with high probability $|\tilde{\lambda}_3 - \lambda_2| > |\lambda_3 - \lambda_2|/2$ and $|\tilde{\lambda}_1 - \lambda_2| > |\lambda_1 - \lambda_2|/2$. Hence we can apply Davis-Kahan theorem. Compiling all constants into c we obtain

$$\|\tilde{f}_{sym} - f_{sym}\|_2 \leq \frac{c\mu}{\sqrt{\log n}} \text{ w.h.p.} \tag{4.22}$$

Finally we relate the perturbations of f_{sym} to the perturbations of f. Since $f_{sym} = \frac{D^{1/2}f}{\|D^{1/2}f\|_2}$, letting $\alpha_n = \|D^{1/2}f\|$, we deduce that

$$\begin{aligned}
\|\tilde{f} - f\|_2 &= \|\tilde{\alpha}_n\tilde{\Delta}\tilde{f}_{sym} - \alpha_n\Delta f_{sym}\|_2 \\
&= \|\Delta(\tilde{\alpha}_n\tilde{f}_{sym} - \alpha_n f_{sym}) + \tilde{\alpha}_n(\tilde{\Delta} - \Delta)\tilde{f}_{sym}\|_2 \\
&\leq \|\Delta\|_2\|\tilde{\alpha}_n\tilde{f}_{sym} - \alpha_n f_{sym}\|_2 + \|\tilde{\alpha}_n\|_2\|\tilde{\Delta} - \Delta\|_2.
\end{aligned}$$

Similarly as for inequality (4.18), we can show that $\|\tilde{D}^{1/2}\|$ and $\|D^{1/2}\|$ are of the same order $O(n)$. Since $\|f\|_2 = \|\tilde{f}\|_2 = 1$, this is also true for $\|\alpha_n\|_2$ and $\|\tilde{\alpha}_n\|_2$. We conclude the proof using inequalities (4.18), (4.19) and (4.22). ■

4.5.6 Bounding ranking perturbations $\|\tilde{\pi} - \pi\|_\infty$

SerialRank's ranking is derived by sorting the Fiedler vector. While the consistency result in Theorem 4.21 shows the ℓ_2 estimation error going to zero as n goes to infinity, this is not sufficient to quantify the maximum displacement of the ranking. To quantify the maximum displacement of the ranking, as in (Wauthier et al., 2013), we need to bound $\|\tilde{\pi} - \pi\|_\infty$ instead.

We bound the maximum displacement of the ranking here with an extra factor $\sqrt{n}$ compared to the sampling rate in (Wauthier et al., 2013). We would only need a better component-wise bound on $\tilde{S} - S$ to get rid of this extra factor $\sqrt{n}$, and we hope to achieve it in future work.

The proof is in two parts: we first bound the ℓ_∞ norm of the perturbation of the Fiedler vector, then translate this perturbation of the Fiedler vector into a perturbation of the ranking.

4.5.6.1 Bounding the ℓ_∞ norm of the Fiedler vector perturbation

We start by a technical lemma bounding $\|(\tilde{S} - S)f\|_\infty$.

Lemma 4.22. *Let $r > 0$, for every $\mu \in (0, 1)$ and n large enough, if $q > \frac{\log^4 n}{\mu^2(2p-1)^4 n}$, then*

$$\|(\tilde{S} - S)f\|_\infty \leq \frac{3\mu n^{3/2}}{\sqrt{\log n}}$$

with probability at least $1 - 2/n$.

Proof. The proof is very much similar to the proof of Lemma 4.15 and can be found the Appendix (section 4.8.2). ■

We now prove the main result of this section, bounding $\|\tilde{f} - f\|_\infty$ with high probability when roughly $O(n^{3/2})$ comparisons are sampled.

Lemma 4.23. *For every $\mu \in (0, 1)$ and n large enough, if $q > \frac{\log^4 n}{\mu^2(2p-1)^4\sqrt{n}}$, then*

$$\|\tilde{f} - f\|_\infty \leq c\frac{\mu}{\sqrt{n \log n}}$$

with probability at least $1 - 2/n$, where c is an absolute constant.

Proof. Notice that by definition $\tilde{L}\tilde{f} = \tilde{\lambda}_2\tilde{f}$ and $Lf = \lambda_2 f$. Hence for $\tilde{\lambda}_2 > 0$

$$\begin{aligned}\tilde{f} - f &= \frac{\tilde{L}\tilde{f}}{\tilde{\lambda}_2} - f \\ &= \frac{\tilde{L}\tilde{f} - Lf}{\tilde{\lambda}_2} + \frac{(\lambda_2 - \tilde{\lambda}_2)f}{\tilde{\lambda}_2}.\end{aligned}$$

Moreover

$$\begin{aligned}\tilde{L}\tilde{f} - Lf &= (\mathbf{I} - \tilde{D}^{-1}\tilde{S})\tilde{f} - (\mathbf{I} - D^{-1}S)f \\ &= (\tilde{f} - f) + D^{-1}Sf - \tilde{D}^{-1}\tilde{S}\tilde{f} \\ &= (\tilde{f} - f) + D^{-1}Sf - \tilde{D}^{-1}\tilde{S}f + \tilde{D}^{-1}\tilde{S}f - \tilde{D}^{-1}\tilde{S}\tilde{f} \\ &= (\tilde{f} - f) + (D^{-1}S - \tilde{D}^{-1}\tilde{S})f + \tilde{D}^{-1}\tilde{S}(f - \tilde{f})\end{aligned}$$

Hence

$$(\mathbf{I}(\tilde{\lambda}_2 - 1) + \tilde{D}^{-1}\tilde{S})(\tilde{f} - f) = (D^{-1}S - \tilde{D}^{-1}\tilde{S} + (\lambda_2 - \tilde{\lambda}_2)\mathbf{I})f. \tag{4.23}$$

Writing S_i the i^{th} row of S and d_i the degree of row i, using the triangle inequality, we deduce that

$$|\tilde{f}_i - f_i| \leq \frac{1}{|\tilde{\lambda}_2 - 1|}\left(|(d_i^{-1}S_i - \tilde{d}_i^{-1}\tilde{S}_i)f| + |\lambda_2 - \tilde{\lambda}_2||f_i| + |\tilde{d}_i^{-1}\tilde{S}_i(\tilde{f} - f)|\right). \tag{4.24}$$

It remains to bound each term separately, using Weyl's inequality for the denominator and previous lemmas for numerator terms, which is detailed in the Appendix (section 4.8.2). ■

4.5.6.2 Bounding the ℓ_∞ norm of the ranking perturbation

First note that the ℓ_∞-norm of the ranking perturbation is equal to the number of pairwise disagreements between the true ranking and the retrieved one, i.e., for any i

$$|\tilde{\pi}_i - \pi_i| = \sum_{j<i} \mathbf{1}_{\tilde{f}_j > \tilde{f}_i} + \sum_{j>i} \mathbf{1}_{\tilde{f}_j < \tilde{f}_i}.$$

Now we will argue that when i and j are far apart, with high probability

$$\tilde{f}_j - \tilde{f}_i = (\tilde{f}_j - f_j) + (f_j - f_i) + (f_i - \tilde{f}_i)$$

will have the same sign as $j-i$. Indeed $|\tilde{f}_j - f_j|$ and $|\tilde{f}_i - f_i|$ can be bounded with high probability by a quantity less than $|f_j - f_i|/2$ for i and j sufficiently "far apart". Hence, $|\tilde{\pi}_i - \pi_i|$ is bounded by the number of pairs that are not sufficiently "far apart". We quantify the term "far apart" in the following proposition.

Theorem 4.24. *For every $\mu \in (0,1)$ and n large enough, if $q > \frac{\log^4 n}{\mu^2(2p-1)^2\sqrt{n}}$, then*

$$\|\tilde{\pi} - \pi\|_\infty \leq c\mu n,$$

with probability at least $1 - 2/n$, where c is an absolute constant.

Proof. We assume w.l.o.g. in the following that the true ranking is the identity, hence the unperturbed Fiedler vector f is strictly increasing. We first notice that for any $j > i$

$$\tilde{f}_j - \tilde{f}_i = (\tilde{f}_j - f_j) + (f_j - f_i) + (f_i - \tilde{f}_i).$$

Hence for any $j > i$

$$\|\tilde{f} - f\|_\infty \leq \frac{|f_j - f_i|}{2} \implies \tilde{f}_j \geq \tilde{f}_i.$$

Consequently, fixing an index i_0,

$$\sum_{j>i_0} \mathbf{1}_{\tilde{f}_j < \tilde{f}_{i_0}} \leq \sum_{j>i_0} \mathbf{1}_{\|\tilde{f}-f\|_\infty > \frac{|f_j - f_{i_0}|}{2}}.$$

Now recall that by Lemma 4.23, for $q > \frac{\log^4 n}{\mu^2 (2p-1)^2 \sqrt{n}}$

$$\|\tilde{f} - f\|_\infty \leq c \frac{\mu}{\sqrt{n \log n}}$$

with probability at least $1 - 2/n$. Hence

$$\sum_{j>i_0} \mathbf{1}_{\tilde{f}_j < \tilde{f}_{i_0}} \leq \sum_{j>i_0} \mathbf{1}_{\|\tilde{f}-f\|_\infty > \frac{|f_j - f_{i_0}|}{2}} \leq \sum_{j>i_0} \mathbf{1}_{\frac{c\mu}{\sqrt{n \log n}} > \frac{|f_j - f_{i_0}|}{2}} \quad \text{w.h.p.}$$

We now consider the case of n odd (a similar reasoning applies for n even). We have $f_j = \frac{j - (n+1)/2}{a_n}$ for all j, with

$$a_n^2 = 2 \sum_{k=0}^{(n-1)/2} k^2 = \frac{2}{6} \frac{n-1}{2} \left(\frac{n-1}{2} + 1 \right) ((n-1) + 1) = \frac{n^3 - n}{12}.$$

Therefore

$$\frac{c\mu}{\sqrt{n \log n}} > \frac{|f_j - f_{i_0}|}{2} \iff \frac{c\mu}{\sqrt{n \log n}} > \frac{|j - i_0| \sqrt{3}}{n^{3/2}} \iff \frac{c\mu n}{\sqrt{3 \log n}} > |j - i_0|.$$

Dividing c by $\sqrt{3}$, we deduce that

$$\sum_{j>i_0} \mathbf{1}_{\tilde{f}_j < \tilde{f}_{i_0}} \leq \sum_{j>i_0} \mathbf{1}_{\frac{c\mu n}{\sqrt{\log n}} > |j - i_0|} = \left\lfloor \frac{c\mu n}{\sqrt{\log n}} \right\rfloor \leq \frac{c\mu n}{\sqrt{\log n}} \quad \text{w.h.p.}$$

Similarly

$$\sum_{j<i_0} \mathbf{1}_{\tilde{f}_j > \tilde{f}_{i_0}} \leq \frac{c\mu n}{\sqrt{\log n}} \quad \text{w.h.p.}$$

Finally, we obtain

$$|\tilde{\pi}_{i_0} - \pi_{i_0}| = \sum_{j<i_0} \mathbf{1}_{\tilde{f}_j > \tilde{f}_{i_0}} + \sum_{j>i_0} \mathbf{1}_{\tilde{f}_j < \tilde{f}_{i_0}} \leq \frac{c\mu n}{\sqrt{\log n}} \quad \text{w.h.p.},$$

where c is an absolute constant. Since the last inequality relies on $\|\tilde{f} - f\|_\infty \leq \frac{c\mu}{\sqrt{n \log n}}$, it is true for all i_0 with probabilty $1 - 2/n$, which concludes the proof. ■

4.6 Numerical Experiments

We now describe numerical experiments using both synthetic and real datasets to compare the performance of SerialRank with several classical ranking methods.

4.6.1 Synthetic Datasets

The first synthetic dataset consists of a matrix of pairwise comparisons derived from a given ranking of n items with uniform, randomly distributed corrupted or missing entries. A second synthetic dataset consists of a full matrix of pairwise comparisons derived from a given ranking of n items, with added "local" noise on the similarity between nearby items. Specifically, given a positive integer m, we let $C_{i,j} = 1$ if $i < j - m$, $C_{i,j} \sim \text{Unif}[-1, 1]$ if $|i - j| \leq m$, and $C_{i,j} = -1$ if $i > j + m$. In Figure 4.2, we measure the Kendall τ correlation coefficient between the true ranking and the retrieved ranking, when varying either the percentage of corrupted comparisons or the percentage of missing comparisons. Kendall's τ counts the number of agreeing pairs minus the number of disagreeing pairs between two rankings, scaled by the total number of pairs, so that it takes values between -1 and 1. Experiments were performed with $n = 100$ and reported Kendall τ values were averaged over 50 experiments, with standard deviation less than 0.02 for points of interest (i.e., with Kendall $\tau > 0.8$).

Results suggest that SerialRank (SR, full red line) produces more accurate rankings than point score (PS, (Wauthier et al., 2013) dashed blue line), Rank Centrality (RC (Negahban et al., 2012) dashed green line), and maximum likelihood (BTL (Bradley and Terry, 1952), dashed magenta line) in regimes with limited amount of corrupted and missing comparisons. In particular SerialRank seems more robust to corrupted comparisons. On the other hand, the performance deteriorates more rapidly in regimes with very high number of corrupted/missing comparisons. For a more exhaustive comparison of SerialRank to state-of-the art ranking algorithms, we refer the interested reader to a recent paper by Cucuringu (2015), which introduces another ranking algorithm called SyncRank, and provides extensive numerical experiments.

4.6.2 Real Datasets

The first real dataset consists of pairwise comparisons derived from outcomes in the TopCoder algorithm competitions. We collected data from 103 competitions among 2742 coders over a

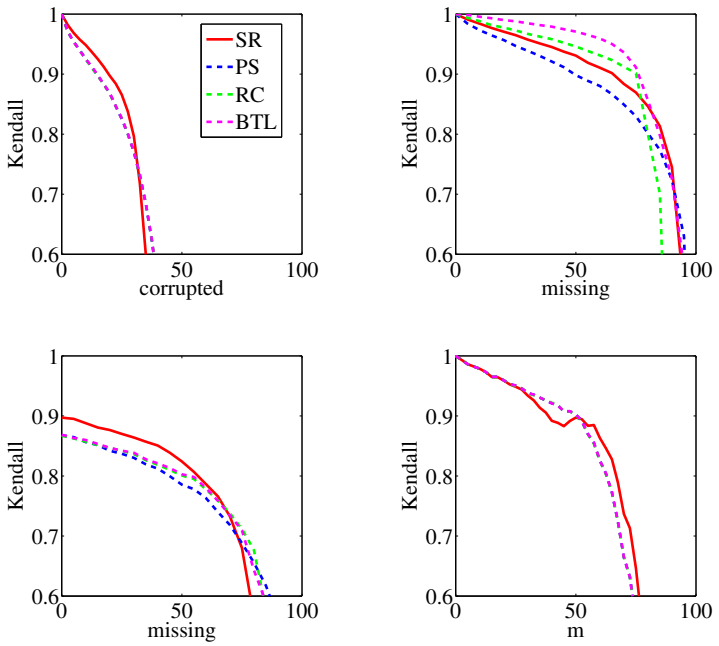


FIGURE 4.2: Kendall τ (higher is better) for SerialRank (SR, full red line), point score (PS, (Wauthier et al., 2013) dashed blue line), Rank Centrality (RC (Negahban et al., 2012) dashed green line), and maximum likelihood (BTL (Bradley and Terry, 1952), dashed magenta line). In the first synthetic dataset, we vary the proportion of corrupted comparisons *(top left)*, the proportion of observed comparisons *(top right)* and the proportion of observed comparisons, with 20% of comparisons being corrupted *(bottom left)*. We also vary the parameter m in the second synthetic dataset *(bottom right)*.

period of about one year. Pairwise comparisons are extracted from the ranking of each competition and then averaged for each pair. TopCoder maintains ratings for each participant, updated in an online scheme after each competition, which were also included in the benchmarks. To measure performance in Figure 4.3, we compute the percentage of upsets (i.e. comparisons disagreeing with the computed ranking), which is closely related to the Kendall τ (by an affine transformation if comparisons were coming from a consistent ranking). We refine this metric by considering only the participants appearing in the top k, for various values of k, i.e. computing

$$l_k = \frac{1}{|\mathcal{C}_k|} \sum_{i,j \in \mathcal{C}_k} \mathbf{1}_{r(i)>r(j)} \mathbf{1}_{C_{i,j}<0}, \tag{4.25}$$

where $\mathcal{C}$ are the pairs (i, j) that are compared and such that i, j are both ranked in the top k, and $r(i)$ is the rank of i. Up to scaling, this is the loss considered in (Kenyon-Mathieu and Schudy, 2007).

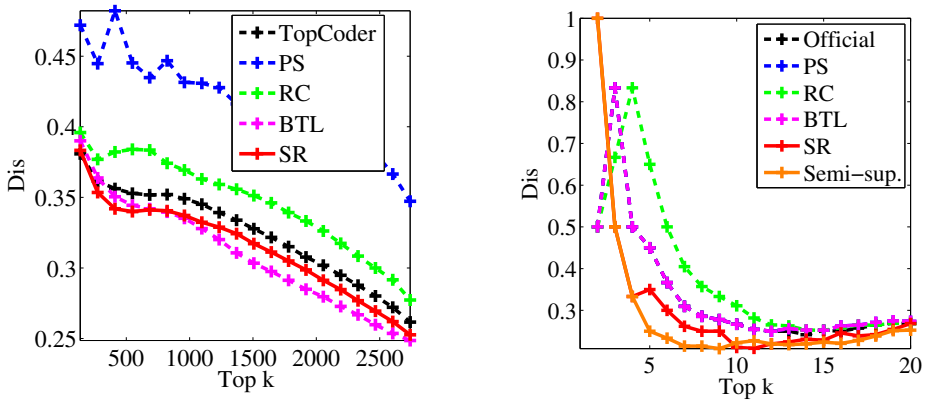


FIGURE 4.3: Percentage of upsets (i.e. disagreeing comparisons, lower is better) defined in (4.25), for various values of k and ranking methods, on TopCoder (*left*) and football data (*right*).

This experiment shows that SerialRank gives competitive results with other ranking algorithms. Notice that rankings could probably be refined by designing a similarity matrix taking into account the specific nature of the data.

TABLE 4.1: Ranking of teams in the England premier league season 2013-2014.

Official	**Row-sum**	**RC**	**BTL**	**SerialRank**	**Semi-Supervised**
Man City (86)	Man City	Liverpool	Man City	Man City	Man City
Liverpool (84)	Liverpool	Arsenal	Liverpool	Chelsea	Chelsea
Chelsea (82)	Chelsea	Man City	Chelsea	Liverpool	Liverpool
Arsenal (79)	Arsenal	Chelsea	Arsenal	Arsenal	Everton
Everton (72)	Everton	Everton	Everton	Everton	Arsenal
Tottenham (69)	Tottenham	Tottenham	Tottenham	Tottenham	Tottenham
Man United (64)	Man United	Man United	Man United	Southampton	Man United
Southampton (56)	Southampton	Southampton	Southampton	Man United	Southampton
Stoke (50)	Stoke	Stoke	Stoke	Stoke	Newcastle
Newcastle (49)	Newcastle	Newcastle	Newcastle	Swansea	Stoke
Crystal Palace (45)	Crystal Palace	Swansea	Crystal Palace	Newcastle	West Brom
Swansea (42)	Swansea	Crystal Palace	Swansea	West Brom	Swansea
West Ham (40)	West Brom	West Ham	West Brom	Hull	Crystal Palace
Aston Villa (38)	West Ham	Hull	West Ham	West Ham	Hull
Sunderland (38)	Aston Villa	Aston Villa	Aston Villa	Cardiff	West Ham
Hull (37)	Sunderland	West Brom	Sunderland	Crystal Palace	Fulham
West Brom (36)	Hull	Sunderland	Hull	Fulham	Norwich
Norwich (33)	Norwich	Fulham	Norwich	Norwich	Sunderland
Fulham (32)	Fulham	Norwich	Fulham	Sunderland	Aston Villa
Cardiff (30)	Cardiff	Cardiff	Cardiff	Aston Villa	Cardiff

4.6.3 Semi-Supervised Ranking

We illustrate here how, in a semi-supervised setting, one can interactively enforce some constraints on the retrieved ranking, using e.g. the semi-supervised seriation algorithm in Chapter 3. We compute rankings of England Football Premier League teams for season 2013-2014

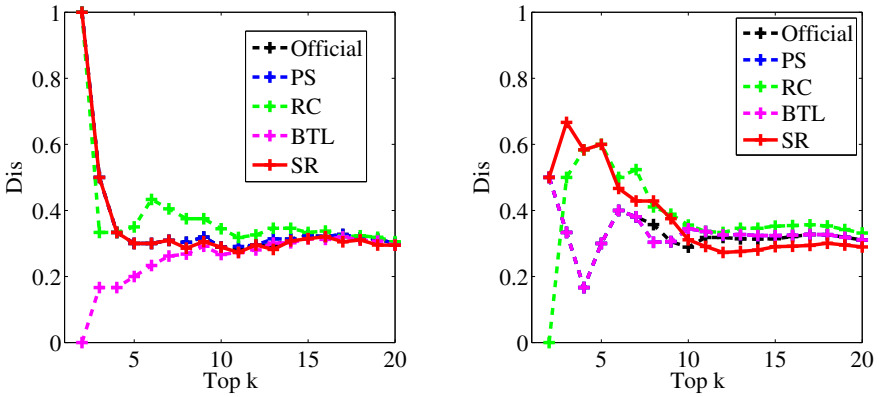


FIGURE 4.4: Percentage of upsets (i.e. disagreeing comparisons, lower is better) defined in (4.25), for various values of k and ranking methods, on England Premier League 2011-2012 season (*left*) and 2012-2013 season (*right*).

(*cf.* figure 4.4 for seasons 2011-2012 and 2012-2013). Comparisons are defined as the averaged outcome (win, loss, or tie) of home and away games for each pair of teams. As shown in Table 4.1, the top half of SerialRank ranking is very close to the official ranking calculated by sorting the sum of points for each team (3 points for a win, 1 point for a tie). However, there are significant variations in the bottom half, though the number of upsets is roughly the same as for the official ranking. To test semi-supervised ranking, suppose for example that we are not satisfied with the ranking of Aston Villa (last team when ranked by the spectral algorithm), we can explicitly enforce that Aston Villa appears before Cardiff, as in the official ranking. In the ranking based on the corresponding semi-supervised seriation problem, Aston Villa is not last anymore, though the number of disagreeing comparisons remains just as low (*cf.* Figure 4.3, *right*).

4.7 Discussion

We have formulated the problem of ranking from pairwise comparisons as a seriation problem, i.e. the problem of ordering from similarity information. By constructing an adequate similarity matrix, we applied a spectral relaxation for seriation to a variety of synthetic and real ranking datasets, showing competitive and in some cases superior performance compared to classical methods, especially in low noise environments. We derived performance bounds for this algorithm in the presence of corrupted and missing (ordinal) comparisons showing that SerialRank produces state-of-the art results for ranking based on ordinal comparisons, e.g. showing exact reconstruction w.h.p. when only $o(\sqrt{n})$ comparisons are missing. On the other hand, performance deteriorates when only a small fraction of comparisons are observed, or in the presence

of very high noise. In this scenario, we showed that local ordering errors can be bounded if the number of samples is of order $\Omega(n^{1.5}\mathrm{polylog}(n))$ which is significantly above the optimal bound of $\Omega(n \log n)$.

A few questions thus remain open, which we pose as future research directions. First of all, from a theoretical perspective, is it possible to obtain an ℓ_∞ bound on local perturbations of the ranking using only $\Omega(n\,\mathrm{polylog(n)})$ sampled pairs? Or, on the contrary, can we find a lower bound for spectral algorithms (i.e. perturbation arguments) imposing more than $\Omega(n\,\mathrm{polylog(n)})$ sampled pairs? Note that those questions hold for all current spectral ranking algorithms.

Another line of research concerns the generalization of spectral ordering methods to more flexible settings, e.g., enforcing structural or a priori constraints on the ranking. Hierarchical ranking, i.e. running the spectral algorithm on increasingly refined subsets of the original data should be explored too. Early experiments suggests this works quite well, but no bounds are available at this point.

Finally, it would be interesting to investigate how similarity measures could be tuned for specific applications in order to improve SerialRank predictive power, for instance to take into account more information than win/loss in sports tournaments. Additional experiments in this vein can be found in Cucuringu (2015).

4.8 Appendix

We now detail several complementary technical results.

4.8.1 Exact recovery results with missing entries

Here, as in Section 4.4, we study the impact of one missing comparison on SerialRank, then extend the result to multiple missing comparisons.

Proposition 4.25. *Given pairwise comparisons $C_{s,t} \in \{-1, 0, 1\}$ between items ranked according to their indices, suppose only one comparison $C_{i,j}$ is missing, with $j - i > 1$ (i.e., $C_{i,j} = 0$), then S^{match} defined in (4.3) remains* strict-R *and the point score vector remains strictly monotonic.*

Proof. We use the same proof technique as in Proposition 4.12. We write the true score and comparison matrix w and C, while the observations are written $\hat{w}$ and $\hat{C}$ respectively. This means in particular that $\hat{C}_{i,j} = 0$. To simplify notations we denote by S the similarity matrix S^{match} (respectively $\hat{S}$ when the similarity is computed from observations). We first study the impact of the missing comparison $C_{i,j}$ for $i < j$ on the point score vector $\hat{w}$. We have

$$\hat{w}_i = \sum_{k=1}^{n} \hat{C}_{k,i} = \sum_{k=1}^{n} C_{k,i} + \hat{C}_{j,i} - C_{j,i} = w_i + 1,$$

similarly $\hat{w}_j = w_j - 1$, whereas for $k \neq i, j$, $\hat{w}_k = w_k$. Hence, w is still strictly increasing if $j > i + 1$. If $j = i + 1$ there is a tie between w_i and w_{i+1}. Now we show that the similarity matrix defined in (4.3) is an R-matrix. Writing $\hat{S}$ in terms of S, we get

$$[\hat{C}\hat{C}^T]_{i,t} = \sum_{k \neq j} \left(\hat{C}_{i,k}\hat{C}_{t,k}\right) + \hat{C}_{i,j}\hat{C}_{t,j} = \sum_{k \neq j} (C_{i,k}C_{t,k}) = \begin{cases} [CC^T]_{i,t} - 1 & \text{if } t < j \\ \left[CC^T\right]_{i,t} + 1 & \text{if } t > j. \end{cases}$$

We thus get

$$\hat{S}_{i,t} = \begin{cases} S_{i,t} - \frac{1}{2} & \text{if } t < j \\ S_{i,t} + \frac{1}{2} & \text{if } t > j, \end{cases}$$

(remember there is a factor $1/2$ in the definition of S). Similarly we get for any $t \neq i$

$$\hat{S}_{j,t} = \begin{cases} S_{j,t} + \frac{1}{2} & \text{if } t < i \\ S_{j,t} - \frac{1}{2} & \text{if } t > i. \end{cases}$$

Finally, for the single corrupted index pair (i, j), we get

$$\hat{S}_{i,j} = \frac{1}{2}\left(n + \sum_{k \neq i,j}\left(\hat{C}_{i,k}\hat{C}_{j,k}\right) + \hat{C}_{i,i}\hat{C}_{j,i} + \hat{C}_{i,j}\hat{C}_{j,j}\right) = S_{i,j} - 0 + 0 = S_{i,j}.$$

For all other coefficients (s, t) such that $s, t \neq i, j$, we have $\hat{S}_{s,t} = S_{s,t}$. Meaning all rows or columns outside of i, j are left unchanged. We first observe that these last equations, together with our assumption that $j - i > 2$, mean that

$$\hat{S}_{s,t} \geq \hat{S}_{s+1,t} \quad \text{and} \quad \hat{S}_{s,t+1} \geq \hat{S}_{s,t}, \quad \text{for any } s < t$$

so $\hat{S}$ remains an R-matrix. To show uniqueness of the retrieved order, we need $j - i > 1$. Indeed, when $j - i > 1$ all these R constraints are strict, which means that $\hat{S}$ is still a strict R-matrix, hence the desired result. ■

We can extend this result to the case where multiple comparisons are missing.

Proposition 4.26. *Given pairwise comparisons $C_{s,t} \in \{-1, 0, 1\}$ between items ranked according to their indices, suppose m comparisons indexed $(i_1, j_1), \ldots, (i_m, j_m)$ are missing, i.e., $C_{i_l,j_j} = 0$ for $i = l, \ldots, m$. If the following condition (4.26) holds true,*

$$|s - t| > 1 \text{ for all } s \neq t \in \{i_1, \ldots, i_m, j_1, \ldots, j_m\} \tag{4.26}$$

then S^{match} defined in (4.3) remains strict-R *and the point score vector remains strictly monotonic.*

Proof. Proceed similarly as in the proof of Proposition 4.13, except that shifts are divided by two. ■

We also get the following corollary.

Corollary 4.27. *Given pairwise comparisons $C_{s,t} \in \{-1, 0, 1\}$ between items ranked according to their indices, suppose m comparisons indexed $(i_1, j_1), \ldots, (i_m, j_m)$ are either corrupted or missing. If condition (4.7) holds true then S^{match} defined in (4.3) remains* strict-R.

Proof. Proceed similarly as the proof of Proposition 4.13, except that shifts are divided by two for missing comparisons. ■

4.8.2 Standard theorems and technical lemmas used in spectral perturbation analysis (section 4.5)

We first recall Weyl's inequality and a simplified version of Davis-Kahan theorem which can be found in (Stewart and Sun, 1990; Stewart, 2001; Yu et al., 2015).

Theorem 4.28. (Weyl's inequality) *Consider a symmetric matrix A with eigenvalues $\lambda_1, \ldots, \lambda_n$ and $\tilde{A}$ a symmetric perturbation of A with eigenvalues $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n$,*

$$\max_i |\tilde{\lambda}_i - \lambda_i| \leq \|\tilde{A} - A\|_2.$$

Theorem 4.29. (Variant of Davis-Kahan theorem (Corollary 3 Yu et al., 2015)) *Let $A, \tilde{A} \in \mathbb{R}^n$ be symmetric, with eigenvalues $\lambda_1 \leq \ldots \leq \lambda_n$ and $\tilde{\lambda}_1 \leq \ldots \leq \tilde{\lambda}_n$ respectively. Fix $j \in \{1, \ldots, n\}$, and assume that $\min(\lambda_j - \lambda_{j-1}, \lambda_{j+1} - \lambda_j) > 0$, where $\lambda_{n+1} := \infty$ and $\lambda_0 := -\infty$. If $v, \tilde{v} \in \mathbb{R}^n$ satisfy $Av = \lambda_j v$ and $\tilde{A}\tilde{v} = \tilde{\lambda}_j \tilde{v}$, then*

$$\sin \Theta(\tilde{v}, v) \leq \frac{2\|\tilde{A} - A\|_2}{\min(\lambda_j - \lambda_{j-1}, \lambda_{j+1} - \lambda_j)}.$$

Moreover, if $\tilde{v}^T v \geq 0$, then

$$\|\tilde{v} - v\|_2 \leq \frac{2\sqrt{2}\|\tilde{A} - A\|_2}{\min(\lambda_j - \lambda_{j-1}, \lambda_{j+1} - \lambda_j)}.$$

When analyzing the perturbation of the Fiedler vector f, we may always reverse the sign of $\tilde{f}$ such that $\tilde{f}^T f \geq 0$ and obtain

$$\|\tilde{f} - f\|_2 \leq \frac{2\sqrt{2}\|\tilde{L} - L\|_2}{\min(\lambda_2 - \lambda_1, \lambda_3 - \lambda_2)}.$$

Lemma 4.30. *Let $r > 0$, for every $\mu \in (0, 1)$ and n large enough, if $q > \frac{\log^4 n}{\mu^2 (2p-1)^4 n}$, then*

$$\|(\tilde{S} - S)f\|_\infty \leq \frac{3\mu n^{3/2}}{\sqrt{\log n}}$$

with probability at least $1 - 2/n$.

Proof. The proof is very much similar to the proof of Lemma 4.15. Let $R = \tilde{S} - S$. We have

$$R_{ij} = \sum_{k=1}^{n} C_{ik} C_{jk} \left(\frac{B_{ik} B_{jk}}{q^2 (2p-1)^2} - 1 \right).$$

Therefore, let $\delta = Rf$

$$\delta_i = \sum_{j=1}^{n} R_{ij} f_j = \sum_{j=1}^{n} \sum_{k=1}^{n} C_{ik} C_{jk} \left(\frac{B_{ik} B_{jk}}{q^2 (2p-1)^2} - 1 \right) f_j.$$

Notice that we can arbitrarily fix the diagonal values of R to zeros. Indeed, the similarity between an element and itself should be a constant by convention, which leads to $R_{ii} = \tilde{S}_{ii} - S_{ii} = 0$ for all items i. Hence we could take $j \neq i$ in the definition of d_i, and we can consider B_{ik} independent of B_{jk} in the associated summation.

We first obtain a concentration inequality for each δ_i. We will then use a union bound to bound $\|\delta\|_\infty = \max |\delta_i|$. Notice that

$$\begin{aligned} \delta_i &= \sum_{j=1}^{n} \sum_{k=1}^{n} C_{ik} C_{jk} \left(\frac{B_{ik} B_{jk}}{q^2 (2p-1)^2} - 1 \right) f_j \\ &= \sum_{k=1}^{n} \left(\frac{C_{ik} B_{ik}}{q(2p-1)} \sum_{j=1}^{n} C_{jk} \left(\frac{B_{jk}}{q(2p-1)} - 1 \right) f_j \right) + \sum_{k=1}^{n} \sum_{j=1}^{n} C_{ik} C_{jk} \left(\frac{B_{ik}}{q(2p-1)} - 1 \right) f_j. \end{aligned}$$

The first term is quadratic while the second is linear, both terms have mean zero since the B_{ik} are independent of the B_{jk}. We begin by bounding the quadratic term. Let $X_{jk} = C_{jk}(\frac{1}{q(2p-1)} B_{jk} - 1) f_j$. We have

$$\mathbf{E}(X_{jk}) = f_j C_{jk} (\tfrac{qp - q(1-p)}{q(2p-1)} - 1) = 0,$$

$$\mathbf{var}(X_{jk}) = \frac{f_j^2 \, \mathbf{var}(B_{jk})}{q^2 (2p-1)^2} = \frac{f_j^2}{q^2 (2p-1)^2} (q - q^2 (2p-1)^2) \leq \frac{f_j^2}{q(2p-1)^2},$$

$$|X_{jk}| = |f_j| \left| \frac{B_{jk}}{q(2p-1)} - 1 \right| \leq \frac{2|f_j|}{q(2p-1)} \leq \frac{2\|f\|_\infty}{q(2p-1)^2}.$$

From corollary 4.19 $\|f\|_\infty \leq 2/\sqrt{n}$. Moreover $\sum_{j=0}^{n} f_j^2 = 1$ since f is an eigenvector. Hence, by applying Bernstein inequality we get for any $t > 0$

$$\mathbf{Prob} \left(\left| \sum_{j=1}^{n} X_{jk} \right| > t \right) \leq 2 \exp \left(\frac{-q(2p-1)^2 t^2}{2(1 + 2t/(3\sqrt{n}))} \right) \leq 2 \exp \left(\frac{-q(2p-1)^2 t^2 n}{2(n + \sqrt{n} t)} \right). \tag{4.27}$$

The rest of the proof is identical to the proof of Lemma 4.15, replacing t by $\sqrt{n} t$. ■

Lemma 4.31. *For every $\mu \in (0,1)$ and n large enough, if $q > \frac{\log^4 n}{\mu^2 (2p-1)^4 \sqrt{n}}$, then*

$$\|\tilde{f} - f\|_\infty \leq c \frac{\mu}{\sqrt{n \log n}}$$

with probability at least $1 - 2/n$*, where* c *is an absolute constant.*

Proof. Notice that by definition $\tilde{L}\tilde{f} = \tilde{\lambda}_2\tilde{f}$ and $Lf = \lambda_2 f$. Hence for $\tilde{\lambda}_2 > 0$

$$\begin{aligned}\tilde{f} - f &= \frac{\tilde{L}\tilde{f}}{\tilde{\lambda}_2} - f \\ &= \frac{\tilde{L}\tilde{f} - Lf}{\tilde{\lambda}_2} + \frac{(\lambda_2 - \tilde{\lambda}_2)f}{\tilde{\lambda}_2}.\end{aligned}$$

Moreover

$$\begin{aligned}\tilde{L}\tilde{f} - Lf &= (\mathbf{I} - \tilde{D}^{-1}\tilde{S})\tilde{f} - (\mathbf{I} - D^{-1}S)f \\ &= (\tilde{f} - f) + D^{-1}Sf - \tilde{D}^{-1}\tilde{S}\tilde{f} \\ &= (\tilde{f} - f) + D^{-1}Sf - \tilde{D}^{-1}\tilde{S}f + \tilde{D}^{-1}\tilde{S}f - \tilde{D}^{-1}\tilde{S}\tilde{f} \\ &= (\tilde{f} - f) + (D^{-1}S - \tilde{D}^{-1}\tilde{S})f + \tilde{D}^{-1}\tilde{S}(f - \tilde{f})\end{aligned}$$

Hence

$$(\mathbf{I}(\tilde{\lambda}_2 - 1) + \tilde{D}^{-1}\tilde{S})(\tilde{f} - f) = (D^{-1}S - \tilde{D}^{-1}\tilde{S} + (\lambda_2 - \tilde{\lambda}_2)\mathbf{I})f. \tag{4.28}$$

Writing S_i the i^{th} row of S and d_i the degree of row i, using the triangle inequality, we deduce that

$$|\tilde{f}_i - f_i| \leq \frac{1}{|\tilde{\lambda}_2 - 1|}\left(|(d_i^{-1}S_i - \tilde{d}_i^{-1}\tilde{S}_i)f| + |\lambda_2 - \tilde{\lambda}_2||f_i| + |\tilde{d}_i^{-1}\tilde{S}_i(\tilde{f} - f)|\right). \tag{4.29}$$

We will now bound each term separately. Define

$$\begin{aligned}&\text{Denom} = |\tilde{\lambda}_2 - 1|, \\ &\text{Num1} = |(d_i^{-1}S_i - \tilde{d}_i^{-1}\tilde{S}_i)f|, \\ &\text{Num2} = |\lambda_2 - \tilde{\lambda}_2||f_i|, \\ &\text{Num3} = |\tilde{d}_i^{-1}\tilde{S}_i(\tilde{f} - f)|.\end{aligned}$$

Bounding Denom First notice that using Weyl's inequality and equation (4.21) (*cf.* proof of Theorem 4.21), we have with probability at least $1-2/n$ $|\tilde{\lambda}_2 - \lambda_2| \leq \|L_R\|_2 \leq \frac{c\mu}{\sqrt{\log n}}$. Therefore there exists an absolute constant c such that with probability at least $1 - 2/n$

$$|\tilde{\lambda}_2 - 1| > c.$$

We now proceed with the numerator terms.

Bounding Num2 Using Weyl's inequality, corollary 4.19 and equation (4.21) (*cf.* proof of Theorem 4.21), we deduce that w.h.p.

$$|\lambda_2 - \tilde{\lambda}_2||f_i|| \leq \frac{c\mu}{\sqrt{n \log n}},$$

where c is an absolute constant.

Bounding Num1 We now bound $|d_i^{-1} S_i - \tilde{d}_i^{-1} \tilde{S}_i|$. We have

$$\begin{aligned} |(\tilde{d}_i^{-1} \tilde{S}_i - d_i^{-1} S_i) f| &= |(\tilde{d}_i^{-1} \tilde{S}_i - \tilde{d}_i^{-1} S_i + \tilde{d}_i^{-1} S_i - d_i^{-1} S_i) f| \\ &\leq |\tilde{d}_i^{-1}||(\tilde{S}_i - S_i) f| + |(\tilde{d}_i^{-1} - d_i^{-1}) S_i f|. \end{aligned}$$

Using equation (4.18) from the proof of Theorem 4.21, we have w.h.p. $|\tilde{d}_i^{-1} - d_i^{-1}| \leq \frac{c\mu}{n^2 \sqrt{\log n}}$. Moreover

$$|\tilde{d}_i^{-1}| \leq |\tilde{d}_i^{-1} - d_i^{-1}| + |d_i^{-1}| \leq \frac{c_1 \mu}{n^2 \sqrt{\log n}} + \frac{c_2}{n^2} \leq \frac{c}{n^2}$$

w.h.p., where c is an absolute constant. Therefore

$$|(\tilde{d}_i^{-1} \tilde{S}_i - d_i^{-1} S_i) f| \leq \frac{c\mu}{n^2 \sqrt{\log n}} |S_i f| + \frac{c}{n^2} |(\tilde{S}_i - S_i) f| \text{ w.h.p.} \tag{4.30}$$

Using the definition of S and corollary 4.19, we get

$$|S_i f| \leq \sum_{j=1}^{n} S_{ij} \max_i |f_i| \leq c \frac{n^2}{\sqrt{n}} \leq c n^{3/2}, \tag{4.31}$$

where c is an absolute constant. Using Lemma 4.22, we get

$$|(\tilde{S}_i - S_i) f| \leq \frac{3\mu n^{3/2}}{\sqrt{\log n}} \text{ w.h.p.} \tag{4.32}$$

Combining (4.30), (4.31) and (4.32) we deduce that there exists a constant c such that

$$|(\tilde{d}_i^{-1} \tilde{S}_i - d_i^{-1} S_i) f| \leq \frac{c\mu}{\sqrt{n \log n}} \text{ w.h.p.}$$

Bounding Num3 Finally we bound the remaining term $|\tilde{d}_i^{-1} \tilde{S}_i (\tilde{f} - f)|$. By Cauchy-Schwartz inequality we have,

$$|\tilde{d}_i^{-1} \tilde{S}_i (\tilde{f} - f)| \leq |\tilde{d}_i^{-1}| \|\tilde{S}_i\|_2 \|\tilde{f} - f\|_2.$$

Notice that

$$\|\tilde{S}_i\|_2 \leq \|S_i\|_2 + \|\tilde{S}_i - S_i\|_2 \leq \|S_i\|_2 + \|\tilde{S} - S\|_2.$$

Since $\|S_i\|_2^2 \leq \|S_1\|_2^2 \leq \frac{n(n+1)(2n+1)}{6}$ and $q > \frac{\log^4 n}{\mu^2(2p-1)^2\sqrt{n}}$ we deduce from Lemma 4.17 that w.h.p. $\|\tilde{S}_i\|_2 \leq \frac{c\mu n^{7/4}}{\sqrt{\log n}}$, where c is an absolute constant, for n large enough. Moreover, as shown above, $|\tilde{d}_i^{-1}| \leq \frac{c}{n^2}$ and we also get from Theorem 4.21 that $\|\tilde{f} - f\|_2 \leq \frac{c\mu}{n^{1/4}\sqrt{\log n}}$ w.h.p. Hence we have

$$|\tilde{d}_i^{-1}\tilde{S}_i(\tilde{f} - f)| \leq \frac{c\mu^2 n^{7/4}}{n^2 n^{1/4}(\log n)} \leq \frac{c\mu}{\sqrt{n \log n}} \text{ w.h.p.},$$

where c is an absolute constant. Combining bounds on the denominator and numerator terms yields the desired result. ■

4.8.3 Numerical experiments with normalized Laplacian

As shown in figure 4.5, results are very similar to those of SerialRank with unnormalized Laplacian. We lose a bit of performance in terms of robustness to corrupted comparisons.

4.8.4 Spectrum of the unnormalized Laplacian matrix

4.8.4.1 Asymptotic Fiedler value and Fiedler vector

We use results on the convergence of Laplacian operators to provide a description of the spectrum of the unnormalized Laplacian in SerialRank. Following the same analysis as in (Von Luxburg et al., 2008) we can prove that asymptotically, once normalized by n^2, apart from the first and second eigenvalue, the spectrum of the Laplacian matrix is contained in the interval $[0.5, 0.75]$. Moreover, we can characterize the eigenfunctions of the limit Laplacian operator by a differential equation, enabling to have an asymptotic approximation for the Fiedler vector.

Taking the same notations as in (Von Luxburg et al., 2008) we have here $k(x, y) = 1 - |x - y|$. The degree function is

$$d(x) = \int_0^1 k(x, y) d\,\mathbf{Prob}(y) = \int_0^1 k(x, y) d(y)$$

(samples are uniformly ranked). Simple calculations give

$$d(x) = -x^2 + x + 1/2.$$

We deduce that the range of d is $[0.5, 0.75]$. Interesting eigenvectors (i.e., here the second eigenvector) are not in this range. We can also characterize eigenfunctions f and corresponding

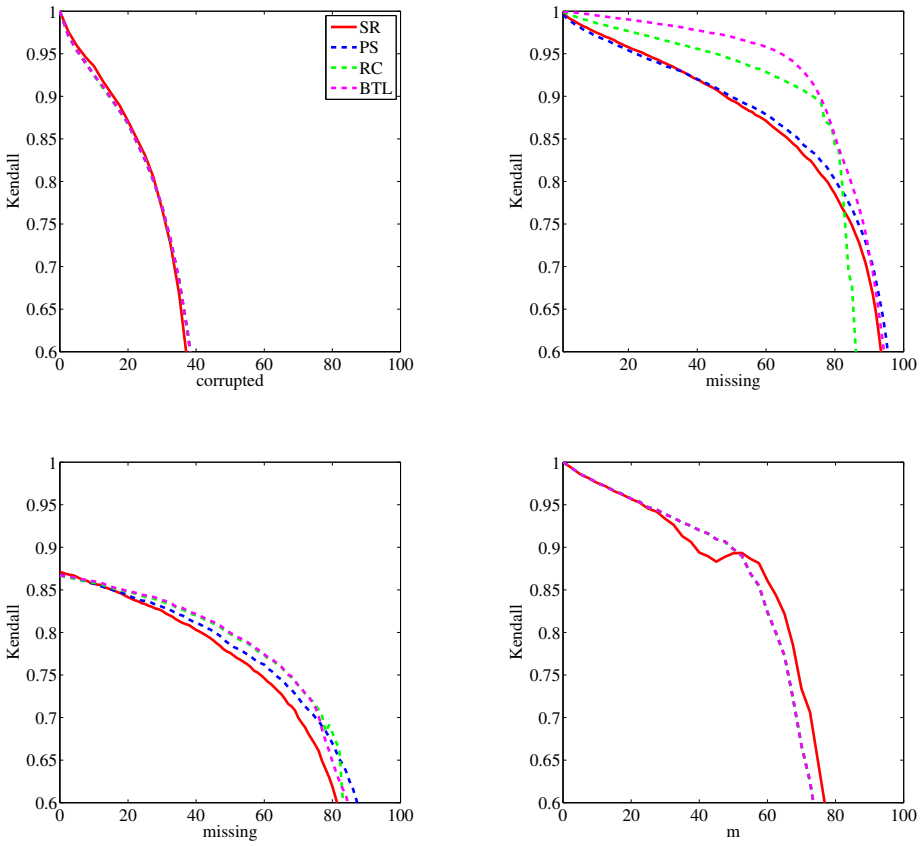


FIGURE 4.5: Kendall τ (higher is better) for SerialRank with normalized Laplacian (SR, full red line), row-sum (PS, (Wauthier et al., 2013) dashed blue line), rank centrality (RC (Negahban et al., 2012) dashed green line), and maximum likelihood (BTL (Bradley and Terry, 1952), dashed magenta line). In the first synthetic dataset, we vary the proportion of corrupted comparisons *(top left)*, the proportion of observed comparisons *(top right)* and the proportion of observed comparisons, with 20% of comparisons being corrupted *(bottom left)*. We also vary the parameter m in the second synthetic dataset *(bottom right)*.

eigenvalues λ by

$$\begin{aligned} & Uf(x) = \lambda f(x) \ \ \forall x \in [0,1] \\ \Leftrightarrow \quad & Mdf(x) - Sf(x) = \lambda f(x) \\ \Leftrightarrow \quad & d(x)f(x) - \int_0^1 k(x,y)f(y)d(y) = \lambda f(x) \\ \Leftrightarrow \quad & f(x)(-x^2 + x + 1/2) - \int_0^1 (1 - |x-y|)f(y)d(y) = \lambda f(x) \end{aligned}$$

Differentiating twice we get

$$f''(x)(1/2 - \lambda + x - x^2) + 2f'(x)(1 - 2x) = 0. \tag{4.33}$$

The asymptotic expression for the Fiedler vector is then a solution to this differential equation,

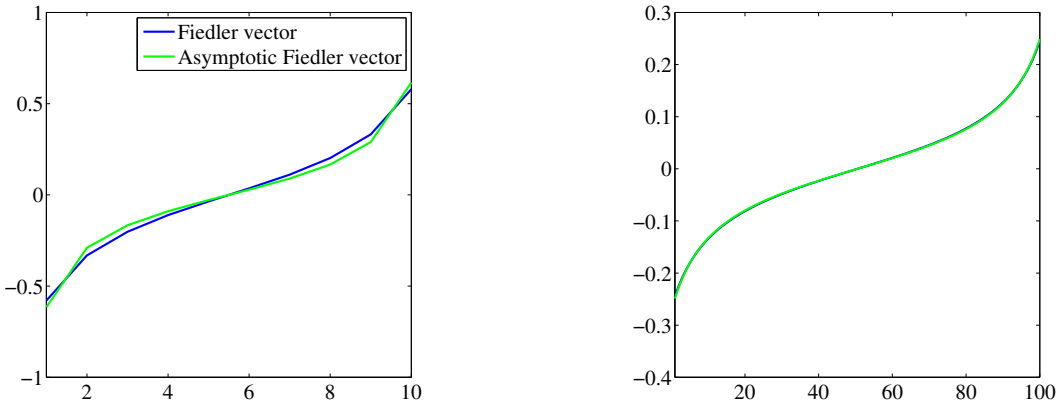


FIGURE 4.6: Comparison between the asymptotic analytical expression of the Fiedler vector and the numeric values obtained from eigenvalue decomposition, for $n = 10$ (*left*) and $n = 100$ (*right*).

with $\lambda < 0.5$. Let γ_1 and γ_2 be the roots of $(1/2 - \lambda + x - x^2)$ (with $\gamma_1 < \gamma_2$). We can suppose that $x \in (\gamma_1, \gamma_2)$ since the degree function is nonnegative. Simple calculations show that

$$f'(x) = \frac{A}{(x-\gamma_1)^2(x-\gamma_2)^2}$$

is solution to (4.33), where A is a constant. Now we note that

$$\begin{aligned}\frac{1}{(x-\gamma_1)^2(x-\gamma_2)^2} &= \frac{1}{(\gamma_1-\gamma_2)^2(\gamma_2-x)^2} + \frac{1}{(\gamma_1-\gamma_2)^2(\gamma_1-x)^2} \\ &\quad - \frac{2}{(\gamma_1-\gamma_2)^3(\gamma_2-x)} + \frac{2}{(\gamma_1-\gamma_2)^3(\gamma_1-x)}.\end{aligned}$$

We deduce that the solution f to (4.33) satisfies

$$f(x) = B + \frac{A}{(\gamma_1-\gamma_2)^2}\left(\frac{1}{\gamma_1-x} + \frac{1}{\gamma_2-x}\right) - \frac{2A}{(\gamma_1-\gamma_2)^3}\left(\log(x-\gamma_1) - \log(\gamma_2-x)\right),$$

where A and B are two constants. Since f is orthogonal to the unitary function for $x \in (0,1)$, we must have $f(1/2) = 0$, hence B=0 (we use the fact that $\gamma_1 = \frac{1-\sqrt{1+4\alpha}}{2}$ and $\gamma_2 = \frac{1+\sqrt{1+4\alpha}}{2}$, where $\alpha = 1/2 - \lambda$).

As shown in figure 4.6 , the asymptotic expression for the Fiedler vector is very accurate numerically, even for small values of n. The asymptotic Fiedler value is also very accurate (2 digits precision for $n = 10$, once normalized by n^2).

4.8.4.2 Bounding the eigengap

We now give two simple propositions on the Fiedler value and the third eigenvalue of the Laplacian matrix, which enable us to bound the eigengap between the second and the third eigenvalues.

Proposition 4.32. *Given all comparisons indexed by their true ranking, let λ_2 be the Fiedler value of S^{match}, we have*

$$\lambda_2 \leq \frac{2}{5}(n^2+1).$$

Proof. Consider the vector x whose elements are uniformly spaced and such that $x^T\mathbf{1} = 0$ and $\|x\|_2 = 1$. x is a feasible solution to the Fiedler eigenvalue minimization problem. Therefore,

$$\lambda_2 \leq x^T L x.$$

Simple calculations give $x^T L x = \frac{2}{5}(n^2+1)$. ■

Numerically the bound is very close to the true Fiedler value: $\lambda_2/n^2 \approx 0.39$ and $2/5 = 0.4$.

Proposition 4.33. *Given all comparisons indexed by their true ranking, the vector $v = [\alpha, -\beta, \ldots, -\beta, \alpha]^T$ where α and β are such that $v^T\mathbf{1} = 0$ and $\|v\|_2 = 1$ is an eigenvector of the Laplacian matrix L of S^{match} The corresponding eigenvalue is $\lambda = n(n+1)/2$.*

Proof. Check that $Lv = \lambda v$. ■

4.8.5 Other choices of similarities

The results in this paper shows that forming a similarity matrix (R-matrix) from pairwise preferences will produce a valid ranking algorithm. In what follows, we detail a few options extending the results of Section 4.2.2.

4.8.5.1 Cardinal comparisons

When input comparisons take continuous values between -1 and 1, several choice of similarities can be made. First possibility is to use S^{glm}. An other option is to directly provide $1 - \text{abs}(\text{C})$ as a similarity to SerialRank. This option has a much better computational cost.

4.8.5.2 Adjusting contrast in S^{match}

Instead of providing S^{match} to SerialRank, we can change the "contrast" of the similarity, i.e., take the similarity whose elements are powers of the elements of S^{match}.

$$S_{i,j}^{\text{contrast}} = (S_{i,j}^{\text{match}})^{\alpha}.$$

This construction gives slightly better results in terms of robustness to noise on synthetic datasets.

4.8.6 Hierarchical Ranking

In a large dataset, the goal may be to rank only a subset of top items. In this case, we can first perform spectral ranking, then refine the ranking of the top set of items using either the SerialRank algorithm on the top comparison submatrix, or another seriation algorithm such as the convex relaxation in Chapter 3. This last method also allows us to solve semi-supervised ranking problems, given additional information on the structure of the solution.

Chapter 5

Conclusion

We have proposed in this thesis new convex and spectral relaxations for the phase retrieval, seriation and ranking problems, providing both theoretical analysis and experimental validation.

In our first contribution, we have experimented algorithms to solve convex relaxation of the phase retrieval problem for molecular imaging. We have shown that exploiting structural assumptions on the signal and the observations, such as sparsity, smoothness or positivity, can significantly speed-up convergence and improve recovery performance. Extensive molecular imaging experiments were performed using simulated data from the Protein Data Bank (PDB).

In our second contribution, we have introduced new convex relaxations for the seriation problem. Besides being more robust to noise than the classical spectral relaxation, these convex relaxations also allow us to impose structural constraints on the solution, hence solve semi-supervised seriation problems. Numerical experiments on DNA de novo assembly gave promising results.

In our third contribution, we have formulated the problem of ranking from pairwise comparisons as a seriation problem. By constructing an adequate similarity matrix, we were able to apply the spectral relaxation of seriation on a variety of synthetic and real datasets, with competitive and in some cases superior performance compared to classical methods. We have performed a careful theoretical analysis of the algorithm in the presence of corrupted and missing comparisons. It appears, both theoretically and empirically, that SerialRank provides state-of-the art results for ranking based on ordinal comparisons, in the presence of limited noise and medium to high number of comparisons. On the other hand, performance deteriorates when only a very small number of observations are available, or in the presence of very high noise.

Several issues are still being investigated. On the first hand, experiments on molecular imaging and DNA assembly need to be performed in more realistic settings, with the help of experts in

the fields. Several collaborations have been initiated, notably with SLAC (Stanford University), and the Génoscope institute, thanks to my advisor Alexandre d'Aspremont.

- Ongoing work with Matthew Seaberg and Alexandre d'Aspremont (ENS Paris & SLAC) tries to reproduce experiments on molecular imaging in a real physical setting (no simulations). It will be very interesting to see which algorithms perform best in practice.
- Ongoing work with Antoine Recanati, Alexandre d'Aspremont (ENS Paris) and Thomas Bruls (Génoscope) is focused on how to improve the design of the similarity matrix in order to be more robust to repetitions in the DNA and high sequencing noise in reads.

On the other hand, several theoretical issues remain, with important practical implications. Notably, how to modify the spectral ranking algorithm proposed in Chapter 4 in order to get good ranking approximations when only a very small number of comparisons are available ($O(n \log n)$). Ongoing work with Mihai Cucuringu (UCLA) studies spectral solutions to the ranking problem that are very close to the one described in Chapter 4. Another line of research concerns the generalization of spectral methods to more flexible settings, e.g., enforcing structural or a priori constraints.

From a broader perspective, other approaches to solve the problems studied in this thesis have been recently proposed, and deserve much attention. Candes et al. (2015b) have detailed a non-convex algorithm with spectral initialization for phase retrieval. The iterative structure of their method makes it much more scalable than SDP relaxations, while preserving the same theoretical guarantees on the number of measurements needed for recovery. Lim and Wright (2014) have extended our convex relaxation of the seriation problem by using sorting networks representations of the permutohedron that are cheaper than representations of permutation matrices (Goemans, 2014). The use of phase retrieval algorithms to solve seriation problems is also being investigated (*cf.* Section 1.5). As for ranking, a formulation as a synchronization problem has been investigated by Cucuringu (2015), using related convex and spectral relaxations to solve it, with very good performance for cardinal comparisons.

Besides the study of the phase retrieval, seriation and ranking problems, new directions of research are already being pursued. A recent pre-print by Roulet et al. (2015) proposes fast algorithms to solve supervised learning problems such as regression or classification, while imposing features, classes, or samples to be clustered. The algorithms provide very good experimental results, though based on non-convex schemes. It remains to study their statistical and computational properties.

Bibliography

D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2), 2007.

Nir Ailon. Active learning ranking from pairwise preferences with almost optimal query complexity. In *NIPS*, pages 810–818, 2011.

Erling D. Andersen and Knud D. Andersen. The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. *High performance optimization*, 33:197–232, 2000.

J.E. Atkins, E.G. Boman, B. Hendrickson, et al. A spectral algorithm for seriation and the consecutive ones problem. *SIAM J. Comput.*, 28(1):297–310, 1998.

F. Bach and M. Jordan. Learning spectral clustering. *Adv. Neural Inf. Process. Syst*, 16:305–312, 2004.

Ed Barbeau. Perron's result and a decision on admissions tests. *Mathematics Magazine*, pages 12–22, 1986.

Stephen T. Barnard, Alex Pothen, and Horst Simon. A spectral algorithm for envelope reduction of sparse matrices. *Numerical linear algebra with applications*, 2(4):317–334, 1995.

Alexander Barvinok. Approximating orthogonal matrices by permutation matrices. *Pure and Applied Mathematics Quarterly*, 2(4):943–961, 2006.

Heinz H. Bauschke, Patrick L. Combettes, and D. Russell Luke. Phase retrieval, error reduction algorithm, and fienup variants: a view from convex optimization. *JOSA A*, 19(7):1334–1345, 2002.

Stephen R. Becker, Emmanuel J. Candès, and Michael C. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011.

A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization : analysis, algorithms, and engineering applications*. MPS-SIAM series on optimization. Society for Industrial and Applied Mathematics : Mathematical Programming Society, Philadelphia, PA, 2001.

Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic analysis on semigroups : theory of positive definite and related functions*, volume 100 of *Graduate texts in mathematics*. Springer-Verlag, New York, 1984.

H.M. Berman, T. Battistuz, TN Bhat, W.F. Bluhm, P.E. Bourne, K. Burkhardt, Z. Feng, G.L. Gilliland, L. Iype, S. Jain, et al. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907, 2002.

D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1998.

Avrim Blum, Goran Konjevod, R Ravi, and Santosh Vempala. Semidefinite relaxations for minimum bandwidth and other vertex ordering problems. *Theoretical Computer Science*, 235 (1):25–42, 2000.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, pages 324–345, 1952.

Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 268–276. Society for Industrial and Applied Mathematics, 2008.

O. Bunk, A. Diaz, F. Pfeiffer, C. David, B. Schmitt, D.K. Satapathy, and JF Veen. Diffractive imaging for periodic samples: retrieving one-dimensional concentration profiles across microfluidic channels. *Acta Crystallographica Section A: Foundations of Crystallography*, 63 (4):306–314, 2007.

Emmanuel J. Candes and Xiaodong Li. Solving quadratic equations via phaselift when there are about as many equations as unknowns. *Foundations of Computational Mathematics*, 14(5): 1017–1026, 2014.

Emmanuel J. Candes, Thomas Strohmer, and Vladislav Voroninski. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274, 2013.

Emmanuel J. Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval from coded diffraction patterns. *Applied and Computational Harmonic Analysis*, 2014.

Emmanuel J. Candes, Yonina C. Eldar, Thomas Strohmer, and Vladislav Voroninski. Phase retrieval via matrix completion. *SIAM Review*, 57(2):225–251, 2015a.

Emmanuel J. Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *Information Theory, IEEE Transactions on*, 61(4):1985–2007, 2015b.

A. Chai, M. Moscoso, and G. Papanicolaou. Array imaging using intensity-only measurements. *Inverse Problems*, 27:015005, 2011.

Moses Charikar, Mohammad Taghi Hajiaghayi, Howard Karloff, and Satish Rao. l_2^2 spreading metrics for vertex ordering problems. *Algorithmica*, 56(4):577–604, 2010.

Stéphan Clémençon and Jérémie Jakubowicz. Kantorovich distances between rankings with applications to rank aggregation. In *Machine Learning and Knowledge Discovery in Databases*, pages 248–263. Springer, 2010.

Jennifer Commins, Christina Toft, Mario A. Fares, et al. Computational biology methods and their application to the comparative genomics of endocellular symbiotic bacteria of insects. *Biological procedures online*, 11(1):52–78, 2009.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-interscience, 2012.

Mihai Cucuringu. Sync-rank: Robust ranking, constrained ranking and rank aggregation via eigenvector and semidefinite programming synchronization. *arXiv preprint arXiv:1504.01070*, 2015.

G. B. Dantzig. *Linear programming and extensions*. Princeton University press, 1963.

A. d'Aspremont and N. El Karoui. Weak recovery conditions from graph partitioning bounds and order statistics. *Mathematics of Operations Research*, 38(2):228–247, 2013.

Alexandre d'Aspremont and Stephen Boyd. Relaxations and randomized methods for nonconvex qcqps. *EE392o Class Notes, Stanford University*, 2003.

Jean-Charles de Borda. Mémoire sur les élections au scrutin. 1781.

Nicolas de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Imprimerie Royale, 1785.

Persi Diaconis and Ronald L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 262–268, 1977.

Martin Dierolf, Andreas Menzel, Pierre Thibault, Philipp Schneider, Cameron M Kewish, Roger Wepf, Oliver Bunk, and Franz Pfeiffer. Ptychographic x-ray computed tomography at the nanoscale. *Nature*, 467(7314):436–439, 2010.

Chris Ding and Xiaofeng He. Linearized cluster assignment via spectral ordering. In *Proceedings of the twenty-first international conference on Machine learning*, page 30. ACM, 2004.

John C. Duchi, Lester W. Mackey, and Michael I. Jordan. On the consistency of ranking algorithms. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 327–334, 2010.

John C. Duchi, Lester Mackey, Michael I. Jordan, et al. The asymptotics of ranking algorithms. *The Annals of Statistics*, 41(5):2292–2323, 2013.

C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. *Proceedings of the Tenth International World Wide Web Conference*, 2001a.

Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. ACM, 2001b.

P. Erdös and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.

P. Erdös and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.

Guy Even, Joseph Seffi Naor, Satish Rao, and Baruch Schieber. Divide-and-conquer approximation algorithms via spreading metrics. *Journal of the ACM (JACM)*, 47(4):585–616, 2000.

Uriel Feige. Approximating the bandwidth via volume respecting embeddings. *Journal of Computer and System Sciences*, 60(3):510–539, 2000.

Uriel Feige and James R. Lee. An improved approximation ratio for the minimum linear arrangement problem. *Information Processing Letters*, 101(1):26–29, 2007.

Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

J.R. Fienup. Phase retrieval algorithms: a comparison. *Applied Optics*, 21(15):2758–2769, 1982.

F. Fogel, R. Jenatton, F. Bach, and A. d'Aspremont. Convex relaxations for permutation problems. *NIPS 2013, arXiv:1306.4805*, 2013.

LR Ford and Delbert Ray Fulkerson. *Flows in networks*, volume 1962. Princeton Princeton University Press, 1962.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *The Journal of machine learning research*, 4:933–969, 2003.

A. Frieze and R. Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

D.R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific journal of mathematics*, 15(3):835, 1965.

Gemma C Garriga, Esa Junttila, and Heikki Mannila. Banded structure in binary matrices. *Knowledge and information systems*, 28(1):197–226, 2011.

Alan George and Alex Pothen. An analysis of spectral envelope reduction via quadratic assignment problems. *SIAM Journal on Matrix Analysis and Applications*, 18(3):706–732, 1997.

R. Gerchberg and W. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

Michel X. Goemans. Smallest compact formulation for the permutahedron. *Mathematical Programming*, pages 1–7, 2014.

M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.

Joseph Goodman. Introduction to fourier optics. 2008.

D. Griffin and J. Lim. Signal estimation from modified short-time fourier transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(2):236–243, 1984.

Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer, 1988.

Lars Hagen and Andrew B. Kahng. New spectral methods for ratio cut partitioning and clustering. *Computer-aided design of integrated circuits and systems, ieee transactions on*, 11(9): 1074–1085, 1992.

R.W. Harrison. Phase problem in crystallography. *JOSA A*, 10(5):1046–1055, 1993.

C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior–point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.

Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill$^{\text{TM}}$: A bayesian skill rating system. In *Advances in Neural Information Processing Systems*, pages 569–576, 2006.

Frank Roy Hodson. *The La Tène cemetery at Münsingen-Rain: catalogue and relative chronology*, volume 5. Stämpfli, 1968.

L. Huang, D. Yan, M.I. Jordan, and N. Taft. Spectral Clustering with Perturbed Data. *Advances in Neural Information Processing Systems (NIPS)*, 2008.

Peter J Huber. Pairwise comparison and ranking: optimum properties of the row sum procedure. *The annals of mathematical statistics*, pages 511–520, 1963.

David R Hunter. MM algorithms for generalized bradley-terry models. *Annals of Statistics*, pages 384–406, 2004.

Kevin G Jamieson and Robert D Nowak. Active ranking using pairwise comparisons. In *NIPS*, volume 24, pages 2240–2248, 2011.

Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, and Yinyu Ye. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, 127(1):203–244, 2011.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

I. Johnson, K. Jefimovs, O. Bunk, C. David, M. Dierolf, J. Gray, D. Renker, and F. Pfeiffer. Coherent diffractive imaging using phase front modifications. *Physical review letters*, 100 (15):155503, 2008.

Norman L. Johnson and Samuel Kotz. *Distributions in statistics: Continuous multivariate distributions*. Wiley, 1972.

N. K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

James P. Keener. The perron-frobenius theorem and the ranking of football teams. *SIAM review*, 35(1):80–93, 1993.

David G. Kendall. Abundance matrices and seriation in archaeology. *Probability Theory and Related Fields*, 17(2):104–112, 1971.

Maurice G. Kendall and B. Babington Smith. On the method of paired comparisons. *Biometrika*, 31(3-4):324–345, 1940.

Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 95–103. ACM, 2007.

L. G. Khachiyan. A polynomial algorithm in linear programming (in Russian). *Doklady Akedamii Nauk SSSR*, 244:1093–1096, 1979.

J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46: 604–632, 1999.

J. Kuczynski and H. Wozniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM J. Matrix Anal. Appl*, 13(4):1094–1122, 1992.

Monique Laurent and Matteo Seminaroti. The quadratic assignment problem is easy for robinsonian matrices with toeplitz structure. *Operations Research Letters*, 43(1):103–109, 2015.

Eugene L. Lawler. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963.

Innar Liiv. Seriation and matrix reordering methods: An historical overview. *Statistical analysis and data mining*, 3(2):70–91, 2010.

Cong Han Lim and Stephen Wright. Beyond the birkhoff polytope: Convex relaxations for vector permutation problems. In *Advances in Neural Information Processing Systems*, pages 2168–2176, 2014.

L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

RD Luce. *Individual choice behavior*. Wiley, 1959.

Filipe Maia. Spsim. 2013.

João Meidanis, Oscar Porto, and Guilherme P. Telles. On the consecutive ones property. *Discrete Applied Mathematics*, 88(1):325–354, 1998.

J. Miao, T. Ishikawa, Q. Shen, and T. Earnest. Extending x-ray crystallography to allow the imaging of noncrystalline materials, cells, and single protein complexes. *Annu. Rev. Phys. Chem.*, 59:387–410, 2008.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pairwise comparisons. In *NIPS*, pages 2483–2491, 2012.

A. Nemirovski. Sums of random symmetric matrices and quadratic optimization under orthogonality constraints. *Mathematical programming*, 109(2):283–317, 2007.

Arkadi Nemirovski and Uriel Rothblum. On complexity of matrix scaling. *Linear Algebra and its Applications*, 302:435–460, 1999.

A. Nemirovskii and D. Yudin. Problem complexity and method efficiency in optimization. *Nauka (published in English by John Wiley, Chichester, 1983)*, 1979.

Y. Nesterov. *Global quadratic optimization via conic relaxation*. Number 9860. CORE Discussion Paper, 1998.

Y. Nesterov. *Introductory Lectures on Convex Optimization*. Springer, 2003.

Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*. Society for Industrial and Applied Mathematics, Philadelphia, 1994.

A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, page 849. MIT Press, 2002.

L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Stanford CS Technical Report*, 1998.

L. Portugal, F. Bastos, J. Júdice, J. Paixao, and T. Terlaky. An investigation of interior-point algorithms for the linear transportation problem. *SIAM Journal on Scientific Computing*, 17 (5):1202–1223, 1996.

Arun Rajkumar and Shivani Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proceedings of the 31st International Conference on Machine Learning*, pages 118–126, 2014.

Satish Rao and Andréa W. Richa. New approximation techniques for some linear ordering problems. *SIAM Journal on Computing*, 34(2):388–404, 2005.

William S. Robinson. A method for chronologically ordering archaeological deposits. *American antiquity*, 16(4):293–301, 1951.

Vincent Roulet, Fajwel Fogel, Alexandre d'Aspremont, and Francis Bach. Supervised clustering in the data cube. *arXiv preprint arXiv:1506.04908*, 2015.

Thomas L. Saaty. The analytic hierarchy process: planning, priority setting, resources allocation. *New York: McGraw*, 1980.

Thomas L. Saaty. Decision-making with the ahp: Why is the principal eigenvector necessary. *European journal of operational research*, 145(1):85–91, 2003.

William W. Schapire, Robert E. Cohen, and Yoram Singer. Learning to order things. In *Advances in Neural Information Processing Systems 10: Proceedings of the 1997 Conference*, volume 10, page 451. MIT Press, 1998.

Ohad Shamir and Naftali Tishby. Spectral clustering on a budget. In *International Conference on Artificial Intelligence and Statistics*, pages 661–669, 2011.

Yoav Shechtman, Yonina C. Eldar, Oren Cohen, Henry N. Chapman, Jianwei Miao, and Mordechai Segev. Phase retrieval with application to optical imaging. *arXiv preprint arXiv:1402.7350*, 2014.

WF Sheppard. On the calculation of the double integral expressing normal correlation. *Transactions of the Cambridge Philosophical Society*, 19:23–66, 1900.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

N.Z. Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25:1–11, 1987.

Eric Sibony, Stéphan Clemençon, and Jérémie Jakubowicz. Mra-based statistical learning from incomplete rankings. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1432–1441, 2015.

ACM SIGKDD. Netflix. In *Proceedings of kdd cup and workshop*, 2007.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

Anthony Man-Cho So. Moment inequalities for sums of random matrices and their applications in optimization. *Mathematical programming*, 130(1):125–151, 2011.

G.W. Stewart. *Matrix Algorithms Vol. II: Eigensystems*. Society for Industrial Mathematics, 2001.

G.W. Stewart and J. Sun. *Matrix perturbation theory*. Academic Press, 1990.

Mechthild Stoer and Frank Wagner. A simple min-cut algorithm. *Journal of the ACM (JACM)*, 44(4):585–591, 1997.

K. C. Toh, M. J. Todd, and R. H. Tutuncu. SDPT3 – a MATLAB software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

Sebastiano Vigna. Spectral ranking. *arXiv preprint arXiv:0912.0238*, 2009.

Milan Vojnovic. Contest theory: Incentive mechanisms and ranking methods. Cambridge University Press, 2015.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.

Niko Vuokko. Consecutive ones property and spectral ordering. In *Proceedings of the 10th SIAM International Conference on Data Mining (SDM'10)*, pages 350–360, 2010.

Dorothea Wagner and Frank Wagner. *Between min cut and graph bisection*. Springer, 1993.

I. Waldspurger and S. Mallat. Time-frequency phase recovery. *Working paper*, 2012.

Irène Waldspurger, Alexandre d'Aspremont, and Stéphane Mallat. Phase recovery, maxcut and complex semidefinite programming. *Mathematical Programming*, 149(1-2):47–81, 2015.

Fabian L. Wauthier, Michael I. Jordan, and Nebojsa Jojic. Efficient ranking from pairwise comparisons. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

K.Q. Weinberger and L.K. Saul. Unsupervised Learning of Image Manifolds by Semidefinite Programming. *International Journal of Computer Vision*, 70(1):77–90, 2006.

Yi Yu, Tengyao Wang, and Richard J. Samworth. A useful variant of the davis–kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015.

Ernst Zermelo. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, 1929.

Qing Zhao, Stefan E Karisch, Franz Rendl, and Henry Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2 (1):71–109, 1998.