Diss. ETH No. 17004

Dieter Mitsche

Spectral Methods for Reconstruction Problems

2007

Diss. ETH No. 17004

Spectral Methods for Reconstruction Problems

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZÜRICH

for the degree of
Doctor of Sciences

presented by
DIETER MITSCHE

born in Klagenfurt
citizen of Austria

accepted on the recommendation of
Prof. Dr. Emo Welzl, ETH Zürich, examiner
Prof. Josep Díaz, Universitat Politècnica de Catalunya, co-examiner
Dr. Joachim Giesen, MPI für Informatik, Saarbrücken, co-examiner

2007

Abstract

"Spectral methods" capture generally the class of algorithms which cast their input data as a matrix and then employ eigenvalue and eigenvector techniques from linear algebra. Empirically, spectral methods have been shown to perform successfully in a variety of domains, e.g., text classification [DDL$^+$90], website ranking [PB98, Kle99]. On the other hand, not many theoretical guarantees have been given for spectral algorithms, and there remains a lot of work to be done to obtain a real understanding why these approaches work as well as they do.

In this thesis we use the framework of reconstruction problems to gain such an understanding of these approaches: the basic idea of reconstruction problems is given some input data which are generated according to some model, the task is to recover some of the model parameters. Clearly, algorithms for these problems do not know the hidden structure and are supposed to reconstruct the structure with the plain information of the data. One main advantage of reconstruction problems is that they allow to compare and to validate algorithms: typically, in reconstruction problems there are a few parameters that measure the difficulty of the reconstruction task. An algorithm is better, if it can provably solve the problem for a wider range of parameters (note that the generating data model is known for the purpose of the analysis of the algorithm). Moreover, many algorithms that have been proven to perform well on reconstruction problems (e.g., in computer graphics, see [Dey04]) also work nicely on real world data.

We provide data models and design spectral algorithms for which we show that they often provably reconstruct the model parameters. In particular, we formulate the partitioning problem and the preference elicitation problem as reconstruction problems and provide theoretical guarantees for the correct reconstruction of the hidden structure using our algorithms.

Zusammenfassung

"Spektrale Methoden" umfassen die Klasse von Algorithmen, die ihre Eingabedaten in Matrixform darstellen und dann Eigenwert- bzw. Eigenvektortechniken der linearen Algebra anwenden. In zahlreichen Arbeiten wurde der empirische Erfolg spektraler Methoden aufgezeigt, unter anderem in den Bereichen Textklassifizierung [DDL+90] sowie Ranking von Webseiten [PB98, Kle99]. Auf der anderen Seite gibt es wesentlich weniger Arbeiten, die theoretisch den Erfolg spektraler Methoden erklären. Es fehlt ein wirkliches Verständnis, warum diese Methoden so gut funktionieren.

In dieser Arbeit versuchen wir, mit Hilfe von Rekonstruktionsproblemen ein solches Verständnis spektraler Methoden zu gewinnen. Rekonstruktionsprobleme können so zusammengefasst werden: gegeben seien Eingabedaten, die einem bestimmten Datenmodell folgen; die Aufgabe ist es, einige der Modellparameter ohne Kenntnis des zugrundeliegenden Modells zu rekonstruieren. Algorithmen, die auf solchen Daten arbeiten, kennen die zugrundeliegende Struktur nicht und sollen diese ohne zusätzliche Information, lediglich auf Grund der Eingabedaten, herausfinden. Ein wesentlicher Vorteil von Rekonstruktionsproblemen ist, dass sie es ermöglichen, Algorithmen hinsichtlich ihrer Qualität zu bewerten und zu vergleichen: typischerweise gibt es in Rekonstruktionsproblemen einige Parameter, die die Schwierigkeit der Rekonstruktionsaufgabe messen. Ein Algorithmus ist besser, wenn er für einen grösseren Wertebereich der Parameter beweisbar das Rekonstruktionsproblem lösen kann (man beachte, dass das generierende Datenmodell zum Zweck der Analyse des Algorithmus als bekannt vorausgesetzt werden kann). Darüber hinaus hat sich in zahlreichen Anwendungen (z.B. Computergraphik, vgl. [Dey04]) gezeigt, dass Algorithmen, die sich (beweisbar) gut auf Rekonstruktionsproblemen verhalten, auch auf realen Daten gut funktionieren.

In der Arbeit präsentieren wir Datenmodelle und entwerfen spektrale Algorithmen, von denen wir beweisen können, dass sie oft die Modellparameter rekonstruieren. Im Speziellen formulieren wir das Partitionierungsproblem von Objekten sowie das Präferenzerhebungsproblem als Rekonstruktionsprobleme und geben für die hier vorgestellten Algorithmen theoretische Garantien.

Contents

Chapter 1

Introduction

Reconstruction problems are a general class of problems of the following type: given data obeying a certain model, the task is to reconstruct the model parameters. In other words, we are given some input data of which we assume that they contain some latent structure which we try to recover. The algorithms clearly do not know the hidden structure and try to recover the structure using the information in the data only. Turning to reconstruction problems allows to compare and validate algorithms whose quality otherwise is difficult to assess. An important example of a reconstruction problem is the surface reconstruction problem in computer graphics [Dey04]: given a finite sample from the surface of some solid, compute a surface from the sample that has the same topology as the original surface and is geometrically close to it. Obviously the surface reconstruction problem can only be solved if the surface and the sample obey certain conditions, i.e., if they follow a certain model. However it turned out that many of the algorithms for which guarantees can be proven in a restricted model also work nicely on real world data that almost never meet the conditions of the model. Here we try to do something similar for partitioning problems and the preference elicitation problem, namely, turning them into reconstruction problems by providing a reasonable model. We strongly believe that the framework of reconstruction problems has further applications beyond the scope of partitioning, preference elicitation and surface reconstruction problems.

The results of this thesis are primarily obtained using spectral meth-

ods. Spectral algorithms basically work as follows: at first the input data are transformed into a matrix M (in such a way that one column corresponds to one data point). Then, the top k eigenvalues and eigenvectors (or in general singular values and singular vectors) of M are computed. In this thesis we work primarily with the projector onto the space spanned by the top k eigenvectors. The idea is that if M is "close" (in terms of a certain matrix norm) to a rank k matrix then the top k eigenvalues and eigenvectors contain most information which is contained in M. In particular, the restriction to the top k eigenvectors captures the structured part of M and leaves out the additional noise on top of this structure. Finally, spectral algorithms use these eigenvectors (in our case the projector) as a basis for the reconstruction and recover the structure hidden in M by comparing different data points according to the (dis)similarity of their corresponding entries in these eigenvectors (in our case the projector, respectively).

Empirically, spectral algorithms have been shown to perform well in several domains, e.g., text classification [DDL+90], website ranking [PB98, Kle99]. On the theoretical side, however, much less is known so far why these algorithms perform so well. In this thesis we provide data models for reconstruction problems in which we can prove that our spectral algorithms perform well.

1.1 Summary and organization

The rest of the thesis is divided into three chapters, according to the general theme of the results stated. Here we give a short summary for each chapter.

1.1.1 Preliminaries

In Chapter 2 we recapitulate some concepts and results from discrete probability theory and from linear algebra which are used throughout the thesis. We also introduce notation used later on. The concepts and results are mostly standard and can be found in the following literature:

- Discrete probability theory: [Wel02], [AS00]
- Linear algebra: [Str88], [SS90], [Gut05]

1.1.2 Reconstructing planted partitions

Chapter 3 can be considered as the main part of this thesis. In this chapter we present three different spectral algorithms for partitioning data. The algorithms are analyzed in the so called planted partition model which was introduced by [McS01]. The basic idea of this model is the formulation of the partitioning problem as a reconstruction problem: since there is no single criterion to assess the quality of a partitioning algorithm we assume a partition to be given (planted) and measure the algorithm's quality by its ability to reconstruct the given partition. For the first algorithm, presented in section 2.3, we can prove the best reconstruction guarantees of all three algorithms, but the running time of the algorithm is not polynomial in the size of the input data. The second algorithm (section 2.4) comes with slightly weaker guarantees (still improving before known bounds), and its running time is polynomial in the size of the input data. Finally, the third algorithm (section 2.5) is also polynomial and gives improved guarantees with respect to a different correctness criterion. Moreover, we prove in section 2.6, that if the number of partitions exceeds a certain value, no algorithm will be able to reconstruct the planted partitions with high probability.

The results of this chapter are joint work with Joachim Giesen, and are presented in [GM05b], [GM05c] and [GM05a].

1.1.3 Collaborative ranking

In Chapter 4 we present a spectral algorithm for the collaborative ranking problem: given a set of products and a set of users, each user is asked a usually small subset of all possible product pairs $\{x, y\}$ and has to decide whether he prefers product x over y or the other way around. Assuming that all users belong to a certain user type, the problem is to reconstruct

- the user type to which each user belongs
- the typical ranking of all products of any user type

In this chapter we first show how this problem can be represented in a matrix form and then present a spectral algorithm to solve this problem. As in Chapter 3, we formulate this problem as a reconstruction problem

and introduce a model in which our algorithm can be analyzed rigorously.

The results of this chapter are joint work with Joachim Giesen and Eva Schuberth, and are presented in [GMS07].

Chapter 2

Preliminaries

In this chapter we will give a short presentation of concepts we will use throughout this thesis and which include some topics from linear algebra and discrete probability theory. The goal is not to make a complete survey, but nevertheless we still attempt to make it as self-contained as possible.

Notation. Throughout the thesis $\log$ always denotes the natural logarithm. $\|.\|_2$ denotes the L_2 matrix norm, i.e., $\|M\|_2 := \max_{|x|=1} |Mx|$. We also use the Frobenius norm $\|.\|_F$, which is defined as $\|M\|_F := \sqrt{\sum_{i,j} M_{ij}^2}$. For asymptotic analysis of algorithms we use the Landau symbols: for two functions $f, g : \mathbb{R} \to \mathbb{R}$, we have $f(n) \in O(g(n))$, if there exists a constant $c_1 \in \mathbb{R}^+$, such that $\forall n \geq n_0, f(n) \leq c_1 g(n)$. Also, $f(n) \in \Omega(g(n))$, if there exists a constant $c_2 \in \mathbb{R}^+$, such that $\forall n \geq n_0, f(n) \geq c_2 g(n)$. If both $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$, then $f(n) \in \Theta(g(n))$. Next, $f(n) \in o(g(n))$ ($f(n) \in \omega(g(n))$, respectively), if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ ($\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$, respectively). Finally we say that an event holds asymptotically almost surely, or just a.a.s., if it holds with probability tending to 1 as n tends to infinity.

2.1 Discrete probability

Definition 1 (Probability space). *A* $\textit{probability space}$ *is a pair* $(\Omega, \Pr)$ *where* Ω *is a set and* $\Pr$ *is a mapping* $\Omega \to \mathbb{R}_0^+$ *such that*

$$\sum_{\omega \in \Omega} \Pr[\omega] = 1.$$

Every subset A *of* Ω *is called an* $\textit{event}$*, and the mapping* $\Pr$ *is extended to events by setting*

$$\Pr[A] := \sum_{\omega \in A} \Pr[\omega].$$

The elements in Ω *are called* $\textit{elementary events}$*. If* Ω *is finite and* $\Pr[\omega] = 1/|\Omega|$ *for all* $\omega \in \Omega$*, then* $\Pr$ *is called* $\textit{uniform distribution}$ *on* Ω*. We use* Ω^+ *for the set of elementary events with positive probability, i.e.,*

$$\Omega^+ := \{\omega \in \Omega \mid \Pr[\omega] > 0\}.$$

Definition 2 (Random variable). *Given a probability space* $(\Omega, \Pr)$*, a* $\textit{random variable}$ *is a real-valued function defined on the elementary events of a probability space, i.e.,*

$$X : \Omega \to \mathbb{R}.$$

If A *is an event, then*

$$\omega \mapsto [\omega \in A]$$

is the $\textit{indicator variable}$ *for event* A*, where for a statement* S *we have* $[S] := 1$ *if* S *is true and* $[S] := 0$ *if* S *is false.*

Definition 3 (Independence). *A collection* $X_i, 1 \le i \le n$ *of random variables on a common probability space is called* $\textit{mutually independent}$ *or simply* $\textit{independent}$ *if*

$$\Pr[X_{i_1} = x_{i_1} \wedge \ldots \wedge X_{i_k} = x_{i_k}] = \Pr[X_{i_1} = x_{i_1}] \ldots \Pr[X_{i_k} = x_{i_k}]$$

for all $k \in \{2, \ldots, n\}$*,* $1 \le i_1 \le \ldots \le i_n$*, and* $(x_{i_1}, \ldots, x_{i_k}) \in \mathbb{R}^k$*.*

Definition 4 (Expectation and variance). *Let* X *be a random real-valued variable. The* $\textit{expectation}$ *of* X *is defined as*

$$E[X] := \sum_{x \in X(\Omega^+)} x \Pr[X = x]$$

provided this infinite sum exists. The variance *of* X *is defined as*

$$\mathrm{var}[X] := \mathrm{E}[[X - \mathrm{E}[X]]^2] = \mathrm{E}[X^2] - \mathrm{E}[X]^2.$$

Fact 1 (Linearity of Expectation). *Let* X *and* Y *be random variables defined on a common probability space and let* $c \in \mathbb{R}$*. Then*

$$\mathrm{E}[cX] = c\mathrm{E}[X] \text{ and } \mathrm{E}[X + Y] = \mathrm{E}[X] + \mathrm{E}[Y],$$

provided $\mathrm{E}[X]$ *and* $\mathrm{E}[Y]$ *exist.*

Lemma 1 (Chebyshev's inequality). *Let* X *be a random variable with finite variance* $\mathrm{var}[X]$*. Then for any* $\lambda \in \mathbb{R}^+$*,*

$$\Pr[|X - \mathrm{E}[X]| \geq \lambda\sqrt{\mathrm{var}[X]}] \leq \frac{1}{\lambda^2}.$$

Definition 5 (Conditional probability). *Let* A *and* B *be events in a probability space with* $\Pr[B] > 0$*. The* conditional probability *of* A*, given* B*, is defined to be*

$$\Pr[A|B] := \frac{\Pr[A \cap B]}{\Pr[B]}.$$

Lemma 2 (Chernoff bound). *Let* $X = \sum_{i=1}^{n} X_i$ *where the* X_i*'s are mutually independent* $\{0, 1\}$*-valued random variables with* $\Pr[X_i = 1] = p_i$ *such that* $\mathrm{E}[X]$ *is positive. Then for any* $\delta \in (0, 1]$*,*

$$\Pr[X < (1 - \delta)\mathrm{E}[X]] \leq e^{-\delta^2 \mathrm{E}[X]/2}$$

and for any $\xi \in \mathbb{R}^+$

$$\Pr[X > (1 + \xi)\mathrm{E}[X]] \leq \left(\frac{e^{\xi}}{(1 + \xi)^{1+\xi}}\right)^{\mathrm{E}[X]}.$$

For our purposes the following (weaker, but more concise) corollary is sufficient.

Corollary 1. *For* X *and* X_i *as in Lemma 2 and* $0 < \delta < 1$ *we have*

$$\Pr[X < (1 - \delta)\mathrm{E}[X]] \leq e^{-\delta^2 \mathrm{E}[X]/4}$$

and

$$\Pr[X > (1 + \delta)\mathrm{E}[X]] \leq e^{-\delta^2 \mathrm{E}[X]/4}.$$

Proof. The lower tail follows trivially from Lemma 2.
For the upper tail, note that $(\frac{e^{\delta}}{(1+\delta)^{1+\delta}})^{E[X]} = (e^{\delta-(1+\delta)\log(1+\delta)})^{E[X]}$.
Since the Taylor expansion of $\log(1+\delta)$ is $\sum_{i\geq 1} \frac{\delta^i}{i}(-1)^{i+1}$, we get (just considering the exponent) that

$$\delta - (1+\delta)\log(1+\delta) = \sum_{i\geq 2} \frac{\delta^i}{i(i-1)}(-1)^{i+1}.$$

For $0 < \delta < 2e-1$ this series is less than $-\delta^2/4$, which yields the desired bound. □

2.2 Linear algebra

Definition 6 (Vector space). *A* vector space *over a field F is a nonempty set V together with two operations* vector addition $v, w \in V \mapsto v+w \in V$ *and* scalar multiplication $\alpha \in F, v \in V \mapsto \alpha v \in V$ *satisfying the following axioms:*

- *vector addition forms an abelian group (i.e., it is associative, commutative, has an identity element and for every element an inverse element)*
- *for all $\alpha \in F, v, w \in V$, $\alpha(v+w) = \alpha v + \alpha w$.*
- *for all $\alpha, \beta \in F, v \in V$, $(\alpha+\beta)v = \alpha v + \beta v$.*
- *for all $\alpha, \beta \in F, v \in V$, $\alpha(\beta v) = (\alpha\beta)v$.*
- *for all $v \in V$, we have $1v = v$, where 1 is the multiplicative identity of F.*

Definition 7 (Subspace). *A nonempty set U of a vector space V over a field F is called a* subspace, *if for all $u, v \in U$ and all $\alpha \in F$,*

$$x + y \in U, \text{ and } \alpha x \in U.$$

Fact 2. *Every subspace is a vector space.*

Definition 8 (Linear independence and basis). *A set of vectors $v_1, \ldots, v_m$ of a vector space V over F is called* linearly independent *, if for any $\gamma_i \in F$, $\sum_{i=1}^{m} \gamma_i v_i = \mathbf{o}$ implies that $\gamma_1 = \ldots = \gamma_m = 0$. A set of linearly independent vectors $v_1, \ldots, v_m$ is called a* basis *of V, if any $v \in V$ can be written as $\sum_{i=1}^{m} \gamma_i v_i$ for some $\gamma_i \in F$.*

Definition 9 (Dimension). *The number of basis vectors (in every basis) of a vector space V is called the* $dimension$ *of V.*

Definition 10 (Linear mapping). *A mapping $M : V \to W, v \mapsto Mv$ between two vector spaces V and W over F is called* $linear$*, if for any $v, w \in V$ and any $\alpha \in F$,*

$$M(v + w) = Mv + Mw, \;\; M(\alpha v) = \alpha(Mv).$$

In this thesis we primarily consider matrices which are linear mappings M between the same vector space V, i.e., $M : V \to V$, $v \mapsto Mv$.

Definition 11 (Eigenvalue and Eigenvector). *The number $\lambda \in F$ is called* $eigenvalue$ *of a matrix $M : V \mapsto V$, if there exists an* $eigenvector$ *$v \in V$, $v \neq \mathbf{o}$, such that*

$$Mv = \lambda v.$$

The subspace spanned by k eigenvectors $v_1, \ldots, v_k$ with corresponding eigenvalues $\lambda_1, \ldots, \lambda_k$ is called the $eigenspace$ *corresponding to $\lambda_1, \ldots, \lambda_k$.*

Definition 12 (Eigenbasis). *A basis of eigenvectors of M is called* $eigenbasis$ *of M.*

In the following the underlying field F is always $\mathbb{R}$. A matrix M is called real, if $m_{ij} \in \mathbb{R}$ for all i, j and is called symmetric if $m_{ij} = m_{ji}$ for all i, j.

Fact 3. *Any real symmetric $n \times n$-matrix M has n real eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ and $\mathbb{R}^n$ has a corresponding eigenbasis. The eigenbasis can be chosen to be orthogonal, i.e., we have vectors $v_1, \ldots, v_n$ such that $Mv_i = \lambda_i v_i$, $i = 1, \ldots, n$ and $v_i^T v_j = 0$ for all $i \neq j$.*

In this thesis the eigenvalues of a real symmetric $n \times n$-matrix M are always assumed to be ordered such that $\lambda_1(M) \geq \ldots \geq \lambda_n(M)$ (or simply $\lambda_1 \geq \ldots \geq \lambda_n$ if M is understood).

Fact 4. *For any real symmetric matrix $n \times n$-matrix M,*

$$\|M\|_F \leq \sqrt{\mathrm{rk}(M)}\|M\|_2,$$

where $\mathrm{rk}(M)$ is the $rank$ *of the matrix M, i.e., the number of linearly independent columns of M.*

The next theorem (see [Vu05] and also [FK81] for previous work) is central for our investigations.

Theorem 1 (Vu [Vu05]). *Let $m_{ij}, 1 \leq i \leq j \leq n$, be independent (but not necessarily identical) random variables with the properties*

- $|m_{ij}| \leq K$, *for all* $1 \leq i \leq j \leq n$,
- $E[m_{ij}] = 0$, *for all* $1 \leq i < j \leq n$,
- $var[m_{ij}] \leq \sigma^2$, *for all* $1 \leq i < j \leq n$,

where K *and* σ *are fixed. Define* $m_{ji} := m_{ij}$ *and consider the matrix* $M = (m_{ij})$. *Then there is a positive constant* $c = c(\sigma, K)$ *such that*

$$\lambda_1(M) \leq 2\sigma\sqrt{n} + cn^{1/4}\log n$$

holds asymptotically almost surely.

Remark 1. *A short glance at the proof of Theorem 1 tells us that also*

$$E[\lambda_1(M)] \leq 2\sigma\sqrt{n} + cn^{1/4}\log n,$$

since Theorem 1 is proven by applying Wigner's trace method to bound the expected number of walks of a certain length.

Lemma 3 (Alon, Krivelevich and Vu [AKV02]). *For* M *as in Theorem 1 and* $t \in \omega(1)$ *we have*

$$Pr[|\lambda_1(M) - E[\lambda_1(M)]| \geq t] \leq e^{-(1-o(1))t^2/32}.$$

Corollary 2. *For* M *as in Theorem 1,*

$$\lambda_1(M) \leq 4\sigma\sqrt{n}$$

with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$.

Proof. By the remark following Theorem 1,

$$E[\lambda_1(M)] \leq 2\sigma\sqrt{n} + cn^{1/4}\log n.$$

Thus, by Lemma 3, for $t = (1 - o(1))2\sigma\sqrt{n}$,

$$\begin{aligned} & Pr\left[\lambda_1(M) \geq 4\sigma\sqrt{n}\right] \\ \leq\ & Pr\left[\lambda_1(M) \geq E[\lambda_1(M)] + (1 - o(1))\,2\sigma\sqrt{n}\right] \\ \leq\ & Pr\left[|\lambda_1(M) - E[\lambda_1(M)]| \geq (1 - o(1))\,2\sigma\sqrt{n}\right] \\ \leq\ & e^{-(1-o(1))\sigma^2 n/8}. \end{aligned}$$

□

Theorem 2 (Weyl). *Let M and $\hat{M}$ be two symmetric matrices $\in \mathbb{R}^{n\times n}$ with corresponding eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ and $\hat{\lambda}_1 \geq \ldots \geq \hat{\lambda}_n$, respectively. Then*

$$\max\{\left|\lambda_i - \hat{\lambda}_i\right| \mid i \in \{1,\ldots,n\}\} \leq \|M - \hat{M}\|_2.$$

Definition 13 (Spectral separation). *The spectral separation $\delta_k(M)$ of a symmetric matrix $M \in \mathbb{R}^{n\times n}$ with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ is defined as*

$$\delta_k(M) := \lambda_k - \lambda_{k+1}.$$

Definition 14 (Projection matrix). *The projection matrix P_M^k onto the space spanned by the k largest eigenvalues of a real symmetric matrix M is defined as*

$$P_M^k := \sum_{i=1}^{k} v_i v_i^T,$$

where v_i is the eigenvector corresponding to the i'th largest eigenvalue of M.

Theorem 3 (Stewart [SS90]). *Let M and $\hat{M}$ be two symmetric matrices $\in \mathbb{R}^{n\times n}$ and let P (and $\hat{P}$, respectively) be the projection matrices onto the space spanned by the k largest eigenvalues of M ($\hat{M}$, respectively). Then*

$$\|P - \hat{P}\|_2 \leq \frac{2\|M - \hat{M}\|_2}{\delta_k(M) - 2\|M - \hat{M}\|_2}$$

if $\delta_k(M) > 4\|M - \hat{M}\|_2$.

Chapter 3

Reconstructing planted partitions

3.1 Introduction

The partition reconstruction problem that we study here is related to the k-partition problem. In the latter problem the task is to partition the vertices of a given graph into k equally sized classes such that the number of edges between the classes is minimized. This problem is already NP-hard for $k = 2$, i.e., in the graph bisection case [GJS76]. Thus researchers, see for example [CK01, BS04] and the references therein, started to analyze the problem in specialized but from an application point of view still meaningful graph families – especially families of random graphs. The random graph families typically assume a given partition of the vertices of the graph aka planted partition, which is obscured by random noise. In general the planted partition need not be an optimal solution of the k-partition problem. Nevertheless, we want to assess the quality of a partitioning algorithm not in terms of the probability that it computes or well approximates an optimal k-partition but in terms of its ability to reconstruct a planted partition. The intuition is that for a "meaningful" random graph family the planted partition is with high probability also optimal. The random graph families usually are parameterized, e.g., by the number of vertices n and number of classes k. Two measures to assess the quality of a partitioning algorithm

in terms of these parameters are:

(1) the probability that the algorithm can reconstruct a planted partition (perfect reconstruction), and

(2) the number of items that the algorithm misclassifies (with a suited definition of misclassification).

The best studied random graph family for the partition reconstruction problem is the following: an edge in the graph appears with probability p if its two incident vertices belong to the same planted class and with probability $q < p$ otherwise, independently from all other edges. The probabilities p and q control the density / sparsity of the graph and can depend on the number of vertices n. In general there is a trade-off between the sparsity of the graph and the number of classes k that can be reconstructed, i.e., the sparser the graph the less classes can be reconstructed correctly and vice versa the larger k the denser the graph has to be in order to reconstruct the classes correctly. Here we assume that p and q are fixed, i.e., we deal with dense graphs only. That leaves only k as a free parameter. We believe that this should be enough to assess the power of most partitioning algorithms.

Related work. In this model the most powerful efficient (polynomial in n) algorithms known so far are the algorithm of Shamir and Tsur [ST02], which builds on ideas of Condon and Karp [CK01], and the algorithm of McSherry [McS01]. Both algorithms can asymptotically almost surely reconstruct correctly up to $k = O(\sqrt{n/\log n})$ planted classes. The algorithm of McSherry falls into the category of spectral clustering algorithms that make use of the eigenvalues and eigenvectors of the adjacency matrix of the input graph. The use of spectral methods for clustering has become increasingly popular in recent years. The vast majority of the literature points out the experimental success of spectral methods, see for example the review by Meila et al. [MV]. On the theoretical side much less is known about the reasons why spectral algorithms perform well in partitioning problems. In 1987 Boppana [Bop87] presented a spectral algorithm for recovering the optimal bisection of a graph. Much later Alon et al. [AKS98] showed how the entries in the second eigenvector of the adjacency matrix of a graph can be used to find a hidden clique of size $\Omega(\sqrt{n})$ in a random graph. Spielman and Teng [ST96] showed how bounded degree planar graphs and d-dimensional meshes can be partitioned using the signs of the entries

in the second eigenvector of the adjacency matrix of the graph or mesh, respectively.

3.2 The model and some spectral properties

3.2.1 The planted partition model

In this section we introduce the planted partition reconstruction problem formally and define two quality measures that can be used to compare different partitioning algorithms. We first introduce the $A(\varphi, p, q)$ distribution, see also McSherry [McS01].

$A(\varphi, p, q)$ **distribution.** Given a surjective function $\varphi : \{1, \ldots, n\} \to \{1, \ldots, k\}$ and fixed probabilities $p, q \in (0, 1)$ with $p > q$. The $A(\varphi, p, q)$ distribution is a distribution on the set of $n \times n$ symmetric, 0-1 matrices with zero trace. Let $\hat{A} = (\hat{a}_{ij})$ be a matrix drawn from this distribution. We have $\hat{a}_{ij} = 0$ if $i = j$ and for $i \neq j$,

$$\begin{array}{lll} \Pr(\hat{a}_{ij} = 1) = p & \text{if } \varphi(i) = \varphi(j) \\ \Pr(\hat{a}_{ij} = 0) = 1 - p & \text{if } \varphi(i) = \varphi(j) \\ \Pr(\hat{a}_{ij} = 1) = q & \text{if } \varphi(i) \neq \varphi(j) \\ \Pr(\hat{a}_{ij} = 0) = 1 - q & \text{if } \varphi(i) \neq \varphi(j), \end{array}$$

independently. The *matrix of expectations* $A = (a_{ij})$ corresponding to the $A(\varphi, p, q)$ distribution is given as

$$\begin{array}{ll} a_{ij} = 0 & \text{if } i = j \\ a_{ij} = p & \text{if } \varphi(i) = \varphi(j) \text{ and } i \neq j \\ a_{ij} = q & \text{if } \varphi(i) \neq \varphi(j) \end{array}$$

Since $\sigma^2 := \max\{p(1-p), q(1-q)\} \leq 0.25$, by Corollary 2 we immediately get the following observation.

Observation 1. *For A and $\hat{A}$ as before*

$$\|A - \hat{A}\|_2 \leq 2\sqrt{n}$$

with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$.

Planted partition reconstruction problem. Given a matrix $\hat{A}$ drawn from the $A(\varphi, p, q)$ distribution. Assume that all classes $\varphi^{-1}(l), l \in \{1, \ldots, k\}$ have the same size n/k (we assume w.l.o.g. that n is an integer multiple of k). Then the function φ is called a *partition function*. The planted partition reconstruction problem asks to reconstruct φ up to a permutation of $\{1, \ldots, k\}$ only from $\hat{A}$ (up to permutations of $\{1, \ldots, k\}$).

Quality of a reconstruction algorithm. A planted partition reconstruction algorithm takes a matrix $\hat{A}$ drawn from the distribution $A(\varphi, p, q)$ as input and outputs a function $\psi : \{1, \ldots, n\} \to \{1, \ldots, k'\}$. There are two natural measures to assess the quality of the reconstruction algorithm.

(1) The probability of perfect reconstruction, i.e.,

$$\Pr\left[\varphi = \psi \text{ up to a permutation of } \{1, \ldots, k\}\right].$$

(2) The distribution of the number of elements in $\{1, \ldots, n\}$ misclassified by the algorithm. The definition for the number of misclassifications used here (see also Meila et al. [MV]) is as the size of a maximum weight matching on the weighted, complete bipartite graph whose vertices are the classes $\varphi^{-1}(i), i \in \{1, \ldots, k\}$, and the classes $\psi^{-1}(j), j \in \{1, \ldots, k'\}$, produced by the algorithm. The weight of the edge $\{\varphi^{-1}(i), \psi^{-1}(j)\}$ is $|\varphi^{-1}(i) \cap \psi^{-1}(j)|$, i.e., the size of the intersection of the classes. The matching gives a pairing of the classes defined by φ and ψ. Assume without loss of generality that always $\varphi^{-1}(i)$ and $\psi^{-1}(i)$ are paired. Then the number of misclassifications is given as

$$n - \sum_{i=1}^{\min\{k,k'\}} |\varphi^{-1}(i) \cap \psi^{-1}(i)|.$$

3.2.2 Spectral properties

Here we are concerned with two types of real symmetric matrices. First, any matrix $\hat{A}$ drawn from an $A(\varphi, p, q)$ distribution. Second, the matrix A of expectations corresponding to the distribution $A(\varphi, p, q)$.

We want to denote the eigenvalues of $\hat{A}$ by $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \ldots \geq \hat{\lambda}_n$ and the vectors of a corresponding orthonormal eigenbasis of $\mathbb{R}^n$ (which exists by Fact 3) by $v_1, \ldots, v_n$.

For the sake of the analysis we want to assume here without loss of generality that the matrix A of expectations has a block diagonal structure, i.e., the elements in the i'th class have indices in $\{\frac{n}{k}(i-1)+1, \ldots, \frac{n}{k}i\}$ (all indices are in $\{1, \ldots, n\}$). It is easy to verify that the eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ of A are $(\frac{n}{k}-1)p+(n-\frac{n}{k})q$, $\frac{n}{k}(p-q)-p$ and $-p$ with corresponding multiplicities 1, $k-1$ and $n-k$, respectively. Hence, $\delta_k(A) = \frac{n}{k}(p-q)$. A possible orthonormal basis of the eigenspace corresponding to the k largest eigenvalues of A is u_i, $i = 1, \ldots, k$, whose j'th coordinates are given as follows,

$$u_{ij} = \begin{cases} \sqrt{\frac{k}{n}}, & j \in \{\frac{n}{k}(i-1)+1, \ldots, \frac{n}{k}i\} \\ 0, & \text{else.} \end{cases}$$

Denote by P ($\hat{P}$, respectively) the projection matrix that projects any vector in $\mathbb{R}^n$ onto the eigenspace spanned by the k eigenvectors corresponding to the k largest eigenvalues of the matrix A ($\hat{A}$, respectively). The entries of P can be characterized explicitly: its entries are given as

$$p_{ij} = \begin{cases} \frac{k}{n}, & \varphi(i) = \varphi(j) \\ 0, & \varphi(i) \neq \varphi(j) \end{cases}$$

For the analysis of our algorithms we need the following three lemmas that relate the eigenvalues of $\hat{A}$ and the parameters p, q and k.

Lemma 4. *All the k largest eigenvalues of $\hat{A}$ are larger than $2\sqrt{n}$ and all the $n-k$ smallest eigenvalues of $\hat{A}$ are smaller than $2\sqrt{n}$ with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$ provided that n is sufficiently large and $k < \frac{p-q}{8}\sqrt{n}$.*

Proof. Plugging in our assumption that $k < \frac{p-q}{8}\sqrt{n}$ gives that the k largest eigenvalues of A are larger than $8\sqrt{n} - p > 4\sqrt{n}$. By Observation 1 we have $\|A - \hat{A}\|_2 \leq 2\sqrt{n}$ with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$. Now it follows from Theorem 2 that the k largest eigenvalues of $\hat{A}$ are larger than $2\sqrt{n}$ with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$. Since the $n-k$ smallest eigenvalues of A are $-p$ it also follows that the $n-k$ smallest eigenvalues of $\hat{A}$ are smaller than $2\sqrt{n}$ with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$. □

Lemma 5. *With probability at least* $1 - e^{-(1-o(1))\sigma^2 n/8}$,

$$\begin{aligned} q &\leq \frac{k}{k-1}\frac{\hat{\lambda}_1}{n} + \frac{k}{k-1}\frac{2}{\sqrt{n}} =: q^+ \quad \textit{and} \\ q &\geq \frac{k}{k-1}\frac{\hat{\lambda}_1}{n} - \frac{k}{k-1}\left(\frac{2}{\sqrt{n}} + \frac{1}{k}\right) := q^-. \end{aligned}$$

and with the same probability

$$\begin{aligned} p &\leq \frac{k\hat{\lambda}_2 + \frac{k}{k-1}\hat{\lambda}_1}{n-k} + \frac{2k\sqrt{n}}{(n-k)(k-1)} + \frac{2k\sqrt{n}}{n-k} := p^+ \quad \textit{and} \\ p &\geq \frac{k\hat{\lambda}_2 + \frac{k}{k-1}\hat{\lambda}_1}{n-k} - \frac{2k\sqrt{n}}{(n-k)(k-1)} - \frac{n}{(n-k)(k-1)} - \frac{2k\sqrt{n}}{n-k} := p^-. \end{aligned}$$

Proof. We know that $\lambda_1 = (\frac{n}{k}-1)p+(n-\frac{n}{k})q$. By Theorem 2 and Observation 1, respectively, it holds $|\lambda_1 - \hat{\lambda}_1| \leq 2\sqrt{n}$ with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$. Thus with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$,

$$\hat{\lambda}_1 \in \left[\left(\frac{n}{k}-1\right)p + \left(n-\frac{n}{k}\right)q - 2\sqrt{n},\ \left(\frac{n}{k}-1\right)p + \left(n-\frac{n}{k}\right)q + 2\sqrt{n}\right].$$

From this we get with the same probability (using $0 \leq p \leq 1$)

$$q \leq \frac{k}{k-1}\frac{\hat{\lambda}_1}{n} + \frac{k}{k-1}\frac{2}{\sqrt{n}} \quad \text{and} \quad q \geq \frac{k}{k-1}\frac{\hat{\lambda}_1}{n} - \frac{k}{k-1}\left(\frac{2}{\sqrt{n}} + \frac{1}{k}\right),$$

which proves the statement for q.

Again by Theorem 2 and Observation 1, respectively, it holds

$$\hat{\lambda}_2 \in \left[\frac{n}{k}(p-q) - p - 2\sqrt{n},\ \frac{n}{k}(p-q) - p + 2\sqrt{n}\right]$$

with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$. From this we get with the same probability (using our previously established estimates of q)

$$\begin{aligned} p &\leq \frac{k\hat{\lambda}_2 + \frac{k}{k-1}\hat{\lambda}_1}{n-k} + \frac{2k\sqrt{n}}{(n-k)(k-1)} + \frac{2k\sqrt{n}}{n-k} \quad \text{and} \\ p &\geq \frac{k\hat{\lambda}_2 + \frac{k}{k-1}\hat{\lambda}_1}{n-k} - \frac{2k\sqrt{n}}{(n-k)(k-1)} - \frac{n}{(n-k)(k-1)} - \frac{2k\sqrt{n}}{n-k}, \end{aligned}$$

which proves the statement for p. □

Lemma 6. *With probability at least* $1 - e^{-(1-o(1))\sigma^2 n/8}$ *it holds*

$$\frac{n}{k}(p-q) - p - 2\sqrt{n} \leq \hat{\lambda}_2 \text{ and } \hat{\lambda}_2 \frac{k}{n} - \frac{2k}{\sqrt{n}} \leq p - q,$$

provided n *is sufficiently large.*

Proof. It holds $\lambda_2 = \frac{n}{k}(p-q) - p$. By combining Theorem 2 and Observation 1 we get that with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$ it holds

$$\hat{\lambda}_2 \in \left[\frac{n}{k}(p-q) - p - 2\sqrt{n},\ \frac{n}{k}(p-q) - p + 2\sqrt{n}\right].$$

Hence with the same probability

$$\hat{\lambda}_2 \frac{k}{n} - \frac{2k}{\sqrt{n}} \leq p - q \leq \hat{\lambda}_2 \frac{k}{n} + \frac{2k}{\sqrt{n}} + \frac{k}{n},$$

where we used $p \leq 1$ for the upper bound and $p \geq 0$ for the lower bound. □

In the following three sections we design three spectral algorithms for the planted partition reconstruction problem.

The first algorithm in section 3.3 gives the best reconstruction guarantees among all three algorithms, but its running time is not necessarily polynomial in n. It runs in time $C^{k/2}\,\mathrm{poly}(n)$, for some constant $C > 0$. We prove that this algorithm asymptotically almost surely reconstructs a planted partition if $k \leq c\sqrt{n}$, where c is another sufficiently small constant. Note that the running time adapts to the difficulty of the problem, i.e., it takes longer if k gets larger. For $k \leq c\sqrt{n}$ the running time is subexponential in the size of the input.

The second algorithm in section 3.4 asymptotically almost surely reconstructs a planted partition with k classes if $k \leq c\sqrt{n}/\log\log n$ for some constant c in polynomial time. At the core of this algorithm is a spectral algorithm with weaker guarantees. This algorithm uses a small random submatrix of the adjacency matrix of the input graph to correctly reconstruct a large fraction of the planted classes. Iterating this algorithm allows us to reconstruct the remaining classes. Doing this in a naive way would allow us to correctly reconstruct up to $k = O(\sqrt{n/\log n})$ classes, which would exactly match the bounds obtained in the papers of [ST02] and [McS01]. In order to boost the

power of the algorithm we prune from the small random submatrix the algorithm is currently working with all entries that correspond to classes already reconstructed in earlier iterations. This means that the number of entries in this matrix corresponding to a not yet reconstructed class is increasing relatively to the size of the matrix. This makes the reconstruction problem easier and thus the relative fraction of classes that the algorithm reconstructs increases in every iteration.

Finally, the third algorithm in section 3.5 builds on similar ideas as the second one though we cannot provide as good guarantees for this algorithm when it comes to perfect reconstruction. Instead we get for this algorithm that the relative number of misclassifications for $k = o(\sqrt{n})$ goes to zero asymptotically almost surely when n goes to infinity – for polynomial-time algorithms this further extends the range of k for which non-trivial guarantees can be given.

3.3 Perfect reconstruction in superpolynomial time

In this section we address the first quality measure, namely perfect reconstruction. The algorithm given here is not guaranteed to run in polynomial time (in n). In the first part of the section we give a pseudocode together with some explanation of the algorithm. Since the running time is not polynomial, the second subsection is devoted to the running time analysis of the algorithm. The third subsection then deals with the correctness of the algorithm.

3.3.1 The algorithm

For convenience, we assume here that the input to the algorithm is a $4n \times 4n$ matrix $\hat{A}$ drawn from the $A(\varphi, p, q)$ distribution, where φ is a surjective function from $\{1, \ldots, 4n\} \to \{1, \ldots, k\}$.

GRIDRECONSTRUCT($\hat{A}$)
1 $k :=$ number of eigenvalues $\hat{\lambda}_i$ of $\hat{A}$ that are larger than $4\sqrt{n}$.
2 $\alpha := c_0/\sqrt{k}$.
3 Randomly partition $\{1, \ldots, 4n\}$ into four subsets I_{11}, I_{12}, I_{21}

and I_{22} of equal size.
4 **for** $i, j = 1, 2$ **do**
5 $\hat{A}_{ij}$:= restriction of $\hat{A}$ to index set I_{ij}.
6 **end for**

In the first six lines of the algorithm we do some preprocessing. We compute the eigenvalues of $\hat{A}$ and use them in line 1 to estimate the number of planted partitions. According to Lemma 4 our estimate k is asymptotically almost surely the correct number of planted partitions. In line 2 we use the value of k to set the value α, which is an essential parameter of the algorithm; c_0 is a small constant > 0.

In line 3 we randomly partition the set $\{1, \dots, 4n\}$ into four equally sized subsets. One way to compute such a random partition is to compute a random permutation π of $\{1, \dots, 4n\}$ and assign

$$I_{11} = \{\pi(1), \dots, \pi(n)\}, \qquad I_{12} = \{\pi(n+1), \dots, \pi(2n)\},$$
$$I_{21} = \{\pi(2n+1), \dots, \pi(3n)\}, \quad I_{22} = \{\pi(3n+1), \dots, \pi(4n)\}.$$

For the sake of the analysis we will use a slightly different method: we first put every vertex into one of the four parts with equal probability $\frac{1}{4}$ (independently from all other vertices) and later redistribute randomly chosen elements to make the partition sizes equal. In Lemma 8 below we formalize this. The only reason to partition $\{1, \dots, 4n\}$ into four sets is for the sake of the analysis where we need independence at some point. The main idea behind our algorithm needs only a partitioning into two sets.

In lines 4 to 6 we compute the restrictions of the matrix $\hat{A}$ to the index sets I_{ij}. Note that the $\hat{A}_{ij}$ are $n \times n$ matrices.

The following lines 7 to 30 make up the main part of the algorithm.

7 **for** $i, j = 1, 2$ **do**
8 $l := 1$
9 $\{v_1, \dots, v_k\}$:= orthonormal eigenbasis of the eigenspace ($\subset \mathbb{R}^n$) of $\hat{A}_{ij}$ that corresponds to the k largest eigenvalues.
10 **for** all $(\lambda_1, \dots, \lambda_k) \in (\alpha\mathbb{Z})^k$ with $\sum_{s=1}^{k} \lambda_s^2 \leq 1$ **do**
11 $\hat{C}_l^{ij} := \emptyset$
12 $v := \sum_{s=1}^{k} \lambda_s v_s \,/\, \left|\sum_{s=1}^{k} \lambda_s v_s\right|$
13 I := subset of the index set I_{ij} that corresponds to the

n/k largest coordinates of v (break ties arbitrarily).
14 **for** all $t \in I_{i(j \bmod 2+1)}$ **do**
15 **if** $\sum_{s \in I} \hat{a}_{st} \geq t(n, k, \hat{\lambda}_1, \hat{\lambda}_2)$ **do**
16 $\hat{C}_l^{ij} := \hat{C}_l^{ij} \cup \{t\}$
17 **end if**
18 **end for**
19 **if** $|\hat{C}_l^{ij}| \leq \frac{3}{4}\left(1 + 3(kn^{-3/4})^{1/3}\right)\frac{n}{k}$ **do**
20 $l := l - 1$
21 **else**
22 **for** $1 \leq l' < l$ **do**
23 **if** $\hat{C}_l^{ij} \cap \hat{C}_{l'}^{ij} \neq \emptyset$ **do**
24 $\hat{C}_{l'}^{ij} := \hat{C}_{l'}^{ij} \cup \hat{C}_l^{ij}$; $l := l - 1$; **break**
25 **end if**
26 **end for**
27 **end if**
28 **end for**
29 $l := l + 1$
30 **end for**

The idea in the main part of the algorithm is to sample the unit ball of the eigenspace of a matrix $\hat{A}_{ij}$ on a grid with grid spacing α. The intuition behind this approach is that for every indicator vector of a class there is a vector in the sample that approximates the indicator vector well. We basically search through all grid vectors in the unit ball to find the ones that are good approximations of characteristic vectors. Every vector in the sample is used to form a class $\hat{C}_l^{ij}$ of indices in $I_{i(j \bmod 2+1)}$. That is, for the algorithm we pair the index sets I_{11} with I_{12} and I_{21} with I_{22}. Thus the vectors of the matrix A_{ij}, which corresponds to the index set I_{ij}, are used to reconstruct the classes in the partner index set $I_{i(j \bmod 2+1)}$. The unit ball is sampled in line 12 and elements are taken into class $\hat{C}_l^{ij}$ in lines 15 to 17 using a threshold test. Note that the threshold value $t(n, k, \hat{\lambda}_1, \hat{\lambda}_2)$ is a function of values that all can be computed from $\hat{A}$.

This way we would form too many classes and elements that belong to one class would be scattered over several classes in the reconstruction. To prevent this we reject a class $\hat{C}_l^{ij}$ in lines 19 to 21 if it contains too few elements. We know that the correct reconstruction has to contain roughly n/k elements. In lines 22 to 26 we check if the set $\hat{C}_l^{ij}$ is a

(partial) reconstruction that already has been partially reconstructed. If this is the case then there should exist $\hat{C}_{l'}^{ij}$ with $l' < l$ such that $\hat{C}_l^{ij} \cap \hat{C}_{l'}^{ij} \neq \emptyset$. We combine the partial reconstructions in line 24 and store the result in the set $\hat{C}_{l'}^{ij}$. With the break statement in line 24 we leave the for-loop enclosed by lines 22 and 26.

In lines 31 to 49 we postprocess the reconstructions that we got in the main part of the algorithm.

```
31  for all l ∈ {1,...,k} do
32    Ĉ_l := Ĉ_l^11
33    for all l_1 ∈ {1,...,k} do
34      if Σ_{i∈Ĉ_l^11} Σ_{j∈Ĉ_{l_1}^21} â_ij > s(n,k,λ̂_1,λ̂_2) do
35        Ĉ_l := Ĉ_l ∪ Ĉ_{l_1}^21
36        for all l_2 ∈ {1,...,k} do
37          if Σ_{i∈C_{l_1}^21} Σ_{j∈Ĉ_{l_2}^12} â_ij > s(n,k,λ̂_1,λ̂_2) do
38            Ĉ_l := Ĉ_l ∪ Ĉ_{l_2}^12
39            for all l_3 ∈ {1,...,k} do
40              if Σ_{i∈C_{l_2}^12} Σ_{j∈Ĉ_{l_3}^22} â_ij > s(n,k,λ̂_1,λ̂_2) do
41                Ĉ_l := Ĉ_l ∪ Ĉ_{l_3}^22
42              end if
43            end for
44          end if
45        end for
46      end if
47    end for
48  end for
49  return all Ĉ_l
```

After the main part of the algorithm the reconstruction of a class C_l is distributed into four sets corresponding to the four index sets I_{ij}. The purpose of the postprocessing is to unite these four parts. This is again done using thresholding with threshold value $s(n,k,\hat{\lambda}_1,\hat{\lambda}_2)$, which again can be computed from $\hat{A}$. In line 49 we finally return the computed reconstructions.

3.3.2 Running time analysis

The running time of the algorithm GRIDRECONSTRUCT is essentially bounded by the number of points $(\lambda_1, \ldots, \lambda_k) \in (\alpha\mathbb{Z})^k$ with $\sum_{s=1}^{k} \lambda_s^2 \leq 1$.

Lemma 7. *The number of points* $(\lambda_1, \ldots, \lambda_k) \in (\alpha\mathbb{Z})^k$ *with* $\sum_{s=1}^{k} \lambda_s^2 \leq 1$ *is asymptotically bounded by* $\frac{1}{\sqrt{\pi k}} \left(\frac{\pi 2e}{c_0^2}\right)^{k/2}$.

Proof. We want to bound the number of points $(\lambda_1, \ldots, \lambda_k) \in (\alpha\mathbb{Z})^k$ that are contained in the k-dimensional ball centered at the origin with radius 1. This number is asymptotically the number of such points in the cube $[-1, 1]^k$ times the volume of the k-dimensional unit ball divided by the volume of the cube, which is 2^k. The volume of the k-dimensional unit ball is

$$\frac{\pi^{k/2}}{k/2\,\Gamma(k/2)}$$

and the number of points $(\lambda_1, \ldots, \lambda_k) \in (\alpha\mathbb{Z})^k$ that are contained in $[-1, 1]^k$ is $(2/\alpha)^k$. Plugging in $\alpha = c_0/\sqrt{k}$ and using Stirling's formula ($\Gamma(x) \sim \sqrt{2\pi} e^{-x} x^{x-1/2}$) asymptotically gives for the number of points in the unit ball

$$\begin{aligned} \frac{\pi^{k/2}}{k/2\,\Gamma(k/2)\,2^k} \frac{2^k k^{k/2}}{c_0^k} &= \frac{(k\pi)^{k/2}}{k/2\,\Gamma(k/2)\,c_0^k} \sim \frac{1}{\sqrt{2\pi}} \frac{(2\pi e k)^{k/2}}{\sqrt{k/2}\,(k c_0^2)^{k/2}} \\ &= \frac{1}{\sqrt{\pi k}} \left(\frac{2\pi e}{c_0^2}\right)^{k/2}. \end{aligned}$$

□

Remark 2. The time needed for the preprocessing in the algorithm is polynomially bounded in n. The same holds for the postprocessing. The time we have to spend on the main part is polynomial in n for every point in the intersection of $(\alpha\mathbb{Z})^k$ with the k-dimensional unit ball. That is, the running time of the whole algorithm is asymptotically bounded by

$$\frac{1}{\sqrt{\pi k}} \left(\frac{2\pi e}{c_0^2}\right)^{k/2} \mathrm{poly}(n) \;=\; C^{k/2}\,\mathrm{poly}(n),$$

for some constant $C > 0$.

3.3.3 Correctness proof

In this subsection we prove the following

Theorem 4. *A.a.s., the algorithm* GRIDRECONSTRUCT *reconstructs all* k *classes correctly, provided* $k \leq c\sqrt{n}$ *for a small enough constant* $c > 0$.

In the following A_{ij} and $\hat{A}_{ij}$ always refer to the restrictions of the matrices A and $\hat{A}$, respectively, to the index set I_{ij}. The corresponding projectors onto the space spanned by the k largest eigenvectors are denoted as P_{ij} and $\hat{P}_{ij}$. Also, $C_l^{ij} = C_l \cap I_{ij}$ is the restriction of C_l to I_{ij}.

Lemma 8. *Let* $l \in \{1, \ldots, k\}$. *The size of* C_l^{ij} *is contained in the interval*

$$\left[\left(1 - 3(kn^{-3/4})^{1/3}\right)\frac{n}{k}, \left(1 + 3(kn^{-3/4})^{1/3}\right)\frac{n}{k}\right]$$

with probability at least $1 - e^{-c'n^{1/4}}$ *for some* $c' > 0$.

Proof. As stated in the description of the algorithm, we will proceed in two steps. In the first step we put every element in C_l into a fixed I_{ij} with probability $\frac{1}{4}$, independently from all other vertices. Define X_s as the random variable which is 1 if the s'th element of partition l is put into I_{ij} and 0 otherwise. Let $X := \sum_{s=1}^{4n/k} X_s$ denote the random variable $|C_l^{ij}|$. We get $E[X] = \frac{n}{k}$, and since the X_s are independent, we get using Chernoff's bound (Corollary 1) for any $0 < \delta < 1$

$$\Pr\left[X \leq (1-\delta)E[X]\right] < e^{-\frac{\delta^2}{4}E[X]} = e^{-\delta^2 n/4k}$$

and similarly

$$\Pr\left[X \geq (1+\delta)E[X]\right] < e^{-\frac{\delta^2}{4}E[X]} = e^{-\delta^2 n/4k}.$$

By the same Chernoff bound argument we obtain $(1-\delta)n \leq |I_{ij}| \leq (1+\delta)n$ with probability at least $1 - 2e^{-\delta^2 n/4}$ (for any $0 < \delta < 1$).
In the second step we move some elements between the different sets $I_{i'j'}$, $i', j' = 1, 2$, to make them all of the same size. One way to achieve this is the following: take all sets $I_{i'j'}$, $i', j' = 1, 2$, with more than n elements and redistribute $|I_{i'j'}| - n$ randomly chosen elements from these

sets $I_{i'j'}$ among all other sets $I_{i'j'}$ with $|I_{i'j'}| < n$ such that all sets $I_{i'j'}$, $i', j' = 1, 2$, have exactly n elements after redistribution.
Assuming that both X and $|I_{ij}|$ take values between the stated upper and lower bounds, we redistribute in the second step at most δn many elements for I_{ij}. Conditioning under these facts we can bound the probability that one element from I_{ij} to be redistributed belongs to C_l by $\frac{(1+\delta)n/k}{n}$ from above, since in every step there are at most $(1+\delta)n/k$ elements of C_l^{ij} left, and there are more than n elements left in I_{ij} in total (otherwise nothing is redistributed).
Let Y_s be the indicator variable which is 1 if the s'th element of C_l^{ij} is redistributed and 0 otherwise. We also define an auxiliary indicator variable Z_s which dominates Y_s, i.e., $Y_s = 1$ implies $Z_s = 1$, but Z_s has exactly probability $\frac{(1+\delta)n/k}{n}$ to happen: if $\Pr[Y_s = 1] < \frac{(1+\delta)n/k}{n}$ we (virtually) add some "dummy" elements to I_{ij} (belonging to C_l as well as belonging to the complement of C_l) in every step by some arbitrary but deterministic rule such that $\Pr[Z_s = 1]$ is exactly the desired value. Furthermore, let Z_s be defined for $s = 1, \ldots, (1+\delta)n/k$, whereas the actual size of C_l^{ij} might be smaller. Let $Y = \sum_{s=1}^{|C_l^{ij}|} Y_s$ denote the number of redistributed elements of C_l^{ij} and set $Z = \sum_{s=1}^{(1+\delta)n/k} Z_s$. Clearly, $Z \geq Y$. Provided that the conditions from above hold we assume the worst case that indeed δn elements have to be redistributed to decrease the size of I_{ij} down to n. We get that $E[Z] = \frac{(1+\delta)n}{kn}\delta n = \delta(1+\delta)\frac{n}{k}$. Since the Z_s are all independent, by Chernoff's bound,

$$\begin{aligned} \Pr[Z \geq (1+\delta)E[Z]] &\leq e^{-\delta^2 E[Z]/4} \\ &= e^{-\delta^2 \delta(1+\delta)n/4k}. \end{aligned}$$

This upper bound holds also for Y.
Combining the probabilities, we obtain that after the second step, with probability at least $1 - e^{-c'\delta^3 n/k}$ for any $0 < \delta < 0.1$ and some constant $c' > 0$,

$$\begin{aligned} \frac{n}{k}(1-3\delta) &< \frac{n}{k}((1-\delta) - \delta(1+\delta)^2) \\ &\leq |C_l^{ij}| \\ &\leq \frac{n}{k}((1+\delta) + \delta(1+\delta)^2) \\ &< \frac{n}{k}(1+3\delta). \end{aligned}$$

Setting $\delta = (kn^{-3/4})^{1/3}$ gives the stated result. □

Lemma 9. *The spectral separation $\delta_k(A_{ij})$ is at least*

$$\left(1 - 3(kn^{-3/4})^{1/3}\right) \frac{n}{k}(p-q)$$

with probability at least $1 - ke^{-c'n^{1/4}}$.

Proof. Let A_q be the $n \times n$-matrix with q everywhere except on the main diagonal where it is zero and let A^l be the $n \times n$-matrix, which has entry $p - q$ at positions whose indices both are mapped to the value l by φ (except on the main diagonal where it is zero), and zero otherwise. It holds

$$A_{ij} = A_q + \sum_{l=1}^{k} A^l.$$

By Theorem 2, we have for the k'th largest eigenvalues $\lambda_k(A_{ij})$ of A_{ij},

$$\lambda_k(A_{ij}) \geq \lambda_k \left(\sum_{l=1}^{k} A^l \right) + \lambda_n(A_q).$$

By construction the spectrum of the sum $\sum_{l=1}^{k} A^l$ of matrices is the union of the spectra of the matrices A^l. The spectrum of one such matrix is $(p-q)(\mathrm{rk}(A^l) - 1)$ with multiplicity one, $-(p-q)$ with multiplicity $\mathrm{rk}(A^l) - 1$ and zero with multiplicity $n - \mathrm{rk}(A^l)$. Furthermore, $\lambda_n(A_q) = -q$. By Lemma 8, with probability at least $1 - e^{-c'n^{1/4}}$,

$$\mathrm{rk}(A^l) \geq \left(1 - 3(kn^{-3/4})^{1/3}\right) \frac{n}{k}.$$

Hence taking a union bound we have that with probability at least $1 - ke^{-c'n^{1/4}}$,

$$\lambda_k \left(\sum_{l=1}^{k} A^l \right) \geq \left(1 - 3(kn^{-3/4})^{1/3}\right) \frac{n}{k}(p-q) - (p-q),$$

and thus

$$\lambda_k(A_{ij}) \geq \left(1 - 3(kn^{-3/4})^{1/3}\right) \frac{n}{k}(p-q) - p$$

with the same probability. On the other hand it is easy to see that

$$\lambda_{k+1}(A_{ij}) = \ldots = \lambda_n(A_{ij}) = -p,$$

by explicitly constructing $n - k$ orthogonal eigenvectors for the eigenvalue $-p$. Hence

$$\delta_k(A_{ij}) \geq \left(1 - 3(kn^{-3/4})^{1/3}\right) \frac{n}{k}(p - q)$$

with probability at least $1 - ke^{-c'n^{1/4}}$. □

Lemma 10. *For every unit vector $v \in \mathbb{R}^n$ with $P_{ij}v = v$ the angle θ between v and $\hat{P}_{ij}v$ is bounded by*

$$\theta < \arccos\left(\sqrt{1-\varepsilon}\right), \text{ and } \varepsilon = \frac{4\sqrt{n}}{\left(1 - 3(kn^{-3/4})^{1/3}\right)\frac{n}{k}(p-q) - 4\sqrt{n}}.$$

with probability at least $\left(1 - e^{-(1-o(1))\sigma^2 n/8}\right)\left(1 - ke^{-c'n^{1/4}}\right)$.

Proof. Theorem 3, Observation 1 and Lemma 9 together imply that

$$|(P_{ij} - \hat{P}_{ij})v|^2 < \varepsilon$$

with probability at least $\left(1 - e^{-(1-o(1))\sigma^2 n/8}\right)\left(1 - ke^{-c'n^{1/4}}\right)$. It follows

$$\begin{aligned} \varepsilon &> 1 + |\hat{P}_{ij}v|^2 - 2v^T\hat{P}_{ij}v \\ &= 1 + |\hat{P}_{ij}v|^2 - 2|\hat{P}_{ij}v|\cos\theta, \end{aligned}$$

which in turn gives

$$\cos\theta > \frac{1 + |\hat{P}_{ij}v|^2 - \varepsilon}{2|\hat{P}_{ij}v|}.$$

As a function of $|\hat{P}_{ij}v|$ the cosine of θ is minimized at $|\hat{P}_{ij}v| = \sqrt{1-\varepsilon}$. Thus we have $\cos\theta > \sqrt{1-\varepsilon}$, which gives that the stated bound on θ holds with the stated probability. □

Lemma 11. *For every vector w in the image of the projector $\hat{P}_{ij}$ there is a vector v computed in line 12 of the algorithm* GRIDRECONSTRUCT *such that the angle θ between w and v is bounded by* $\theta < \arccos\left(1 - \alpha\sqrt{k}/2\right)$.

Proof. Observe that every v constructed in the algorithm is contained in the image of $\hat{P}_{ij}$, because it is a linear combination of an eigenbasis corresponding to the k largest eigenvalues of $\hat{A}_{ij}$ and this eigenbasis spans the image of $\hat{P}_{ij}$. Since we are only interested in angles we can assume that w is a unit vector. By construction w must have a vector v' with

$$v' = \sum_{s=1}^{k} \lambda_s v_s \quad \text{with} \quad (\lambda_1, \ldots, \lambda_k) \in (\alpha\mathbb{Z})^k \text{ and } \sum_{s=1}^{k} \lambda_s^2 \leq 1,$$

at Euclidean distance at most $\alpha\sqrt{k}/2$. Scaling the vector v' to get v does not change the angle between w and v'. Using the law of cosines we get for the angle θ between w and v'

$$\begin{aligned} \cos\theta &> \frac{1 + \left(1 - \alpha\sqrt{k}/2\right)^2 - \alpha^2 k/4}{2} \\ &= 1 - \alpha\sqrt{k}/2, \end{aligned}$$

which gives the bound on θ. □

Lemma 12. *For any $l \in \{1, \ldots, k\}$ there is a vector v computed in line 12 of the algorithm* GRIDRECONSTRUCT *such that with probability at least $\left(1 - e^{-(1-o(1))\sigma^2 n/8}\right)\left(1 - ke^{-c'n^{1/4}}\right)$ at least*

$$\left(1 - 4(1 - \cos\beta)\left(1 + 3(kn^{-3/4})^{1/3}\right)\right)\frac{n}{k}$$

of the indices corresponding to the n/k largest entries in v are mapped to l by φ, where $\beta = \arccos\left(1 - \alpha\sqrt{k}/2\right) + \arccos\left(\sqrt{1-\varepsilon}\right)$ and ε as in Lemma 10.

Proof. Let $c_l \in \mathbb{R}^n$ be the normalized characteristic vector of the class C_l^{ij}. By construction it holds $P_{ij}c_l = c_l$. Thus the angle between c_l and $\hat{P}_{ij}c_l$ is bounded by $\arccos(\sqrt{1-\varepsilon})$ with probability at least $\left(1 - e^{-(1-o(1))\sigma^2 n/8}\right)\left(1 - ke^{-c'n^{1/4}}\right)$ by Lemma 10. For the vector $\hat{P}_{ij}c_l$ there exists by Lemma 11 a vector as constructed in line 12 of the algorithm such that the angle between $\hat{P}_{ij}c_l$ and v is bounded by

$\arccos\left(1-\alpha\sqrt{k}/2\right)$. Using the triangle inequality for angles we thus get

$$c_l^T v \geq \cos\left(\arccos\left(1-\alpha\sqrt{k}/2\right)+\arccos\left(\sqrt{1-\varepsilon}\right)\right) \;=\; \cos\beta.$$

Since c_l and v are both unit vectors we can get an upper bound on the length of $|c_l - v|$ from the lower bound on the dot product $c_l^T v$. First we decompose v into the projection of v onto c_l and the orthogonal complement $v^\perp$ of this projection. Since v is a unit vector we have $1 = (c_l^T v)^2 + |v^\perp|^2$. Thus $|v^\perp|^2$ is bounded from above by $1-(\cos\beta)^2$. Also, $|(c_l^T v)c_l - c_l|^2$ is bounded from above by $(1-\cos\beta)^2$ since c_l is a unit vector. Combining the two inequalities we get

$$|v-c_l|^2 = |v^\perp|^2+|(c_l^T v)c_l-c_l|^2 \leq 1-(\cos\beta)^2+(1-\cos\beta)^2 = 2(1-\cos\beta).$$

Let $x = |C_l^{ij}|$ be the size of C_l^{ij} and let y be the number of indices whose corresponding entries in the vector v are among the x largest, but that are not mapped to l by φ. The number y is maximized under the upper bound on $|v-c_l|^2$ if the entries that are "large" in c_l but are "small" in v have a value just smaller than $\frac{1}{2}\sqrt{1/x}$ in v and if the entries whose value is 0 in c_l, but "large" in v, have a value just larger than $\frac{1}{2}\sqrt{1/x}$ in v and if all other entries coincide. For such a vector v it follows $|v-c_l|^2 = \frac{2y}{4x}$, which implies

$$y \leq 4x(1-\cos\beta).$$

Since by Lemma 8 it holds $x \leq \left(1+3(kn^{-3/4})^{1/3}\right)\frac{n}{k}$ with probability at least $1-e^{-c'n^{1/4}}$ we have

$$y \leq 4(1-\cos\beta)\left(1+3(kn^{-3/4})^{1/3}\right)\frac{n}{k}$$

with probability at least $1-ke^{-c'n^{1/4}}$ (taking a union bound). This is also an upper bound on the number of indices whose corresponding entries in the vector v are among the n/k largest, but that are not mapped to l by φ. □

Remark 3. If $k \leq c\sqrt{n}$ and $\alpha = c_0/\sqrt{n}$ then for sufficiently small constants c and $c_0 = c_0(c)$ and for large enough n

$$\left(1-4(1-\cos\beta)\left(1+3(kn^{-3/4})^{1/3}\right)\right) \geq \frac{3}{4}.$$

That is, asymptotically almost surely at least 3/4 of the indices corresponding to the n/k largest entries in v are mapped to l by φ.

Lemma 13. *Let ν_{ij} be a unit vector constructed in round (i,j) in line 12 of the algorithm* GRIDRECONSTRUCT. *Let I be subset of the index set I_{ij} that corresponds to the n/k largest entries in ν.*

If at least $\nu n/k$ of the indices in I are mapped to the same element $l \in \{1,\dots,k\}$ by φ then for $t \in C_l^{i(j \bmod 2+1)}$ it holds

$$\sum_{s\in I} \hat{a}_{st} \;\geq\; (\nu p + (1-\nu)q)\frac{n}{k}(1-\delta)$$

with probability at least $1 - e^{-\frac{\delta^2}{4}(\nu p+(1-\nu)q)\frac{n}{k}}$ (for any $0 < \delta < 1$). If at most $\mu n/k$ of the indices in I are mapped to the same element $l \in \{1,\dots,k\}$ by φ then for $t \in C_l^{i(j \bmod 2+1)}$ it holds

$$\sum_{s\in I} \hat{a}_{st} \;\leq\; (\mu p + (1-\mu)q)\frac{n}{k}(1+\delta)$$

with probability at least $1 - e^{-\frac{q\delta^2}{4}\frac{n}{k}}$ (for any $0 < \delta < 1$).

Proof. The entries $\hat{a}_{st}$ of $\hat{A}$, where $t \in C_l^{i(j \bmod 2+1)}$ is fixed and $s \in I$, are independent Poisson variables for which

$$\begin{array}{lll}
\Pr(\hat{a}_{st} = 1) \;=\; p & \text{if } \varphi(s) = l \\
\Pr(\hat{a}_{st} = 0) \;=\; 1-p & \text{if } \varphi(s) = l \\
\Pr(\hat{a}_{st} = 1) \;=\; q & \text{if } \varphi(s) \neq l \\
\Pr(\hat{a}_{st} = 0) \;=\; 1-q & \text{if } \varphi(s) \neq l.
\end{array}$$

Let X be the random variable

$$X = \sum_{s\in I} \hat{a}_{st}$$

with $t \in I^l_{i(j \bmod 2+1)}$. If at least $\nu n/k$ of the indices in I are mapped to the same element $l \in \{1,\dots,k\}$ by φ then

$$E[X] \geq (\nu p + (1-\nu)q)\frac{n}{k}$$

and we get using Chernoff's bound (Corollary 1)

$$\begin{aligned}
\Pr\left[X \leq (1-\delta)(\nu p + (1-\nu)q)\frac{n}{k}\right] \;&\leq\; P\left[X \leq (1-\delta)E[X]\right] \\
&<\; e^{-\frac{\delta^2}{4}E[X]} \\
&<\; e^{-\frac{\delta^2}{4}(\nu p+(1-\nu)q)\frac{n}{k}}
\end{aligned}$$

for any $0 < \delta < 1$.
If at most $\mu n/k$ of the indices in I are mapped to the same element $l \in \{1, \ldots, k\}$ by φ then

$$q\frac{n}{k} \leq E[X] \leq (\mu p + (1-\mu)q)\frac{n}{k}$$

and we get using Chernoff's bound

$$\begin{aligned} Pr\left[X \geq (1+\delta)(\mu p + (1-\mu)q)\frac{n}{k}\right] &\leq P\left[X \geq (1+\delta)E[X]\right] \\ &< e^{-\frac{\delta^2}{4}E[X]} \\ &< e^{-\frac{q\delta^2}{4}\frac{n}{k}} \end{aligned}$$

for any $0 < \delta < 1$. □

Remark 4. If we choose $\nu = 3/4$ and $\mu = 2/3$ and let $0 < \delta < \frac{(\nu-\mu)(p-q)}{(\nu+\mu)(p-q)+2q}$ then

$$(\nu p + (1-\nu)q)\frac{n}{k}(1-\delta) > (\mu p + (1-\mu)q)\frac{n}{k}(1+\delta)$$

asymptotically almost surely. That is, if we choose the threshold in line 15 of the algorithm GridReconstruct in the interior of the interval

$$\left[\mu p + (1-\mu)q)\frac{n}{k}(1+\delta), (\nu p + (1-\nu)q)\frac{n}{k}(1-\delta)\right]$$

then asymptotically almost surely the test in line 15 is only passed for vectors v (as constructed in line 12 of the algorithm) that have at least $\frac{2}{3}\frac{n}{k}$ indices corresponding to the n/k largest entries in v that are mapped to the same element by φ. Assume this element is $l \in \{1, \ldots, k\}$. The elements that pass the test (and are subsequently put into a class) are all mapped to l by φ. The only problem is that it is possible that for some vector v only some of the elements that are mapped to l by φ and that take the test also pass it. But from Remark 3 we know that for every $l \in \{1, \ldots, k\}$ there is a vector v such that asymptotically almost surely (taking a union bound) all elements that are mapped to l by φ and that take the test also pass it. That is the reason for the postprocessing in lines 19 to 28 of the algorithm. It remains to show how a good threshold value can be found. Only obstacle to that is that we do not know the values of p and q when running the algorithm.

Remark 5. Asymptotically almost surely we can approximate q arbitrarily well by $\frac{k}{k-1}\frac{\hat{\lambda}_1}{n}$ for growing n if $k \in \omega(1)$, see Lemma 5. That is not the case for p. If $k = c\sqrt{n}$ then we can approximate p asymptotically by $\frac{k\hat{\lambda}_2 + \frac{k}{k-1}\hat{\lambda}_1}{n-k}$ only up to a constant that depends on c. But for sufficiently small c if we choose

$$0 < \delta < \frac{\left(\frac{3}{4}p^- + \frac{1}{4}q^-\right) - \left(\frac{2}{3}p^+ + \frac{1}{3}q^+\right)}{\left(\frac{3}{4}p^+ + \frac{1}{4}q^+\right) + \left(\frac{2}{3}p^+ + \frac{1}{3}q^+\right)},$$

which is positive for sufficiently large n, the algorithm GRIDRECONSTRUCT asymptotically almost surely reconstructs for $k \leq c\sqrt{n}$ (and $k \in \omega(1)$) all the classes C_l^{ij} for all $l \in \{1, \dots, k\}$ (up to a permutation of $\{1, \dots, k\}$) if we choose the threshold

$$t(n, k, \hat{\lambda}_1, \hat{\lambda}_2) = \left(\frac{2}{3}p^+ + \frac{1}{3}q^+\right)\frac{n}{k}(1+\delta)$$

in line 15 of the algorithm.

Lemma 14. *Let $0 < \delta < 1$ be a constant. It holds asymptotically almost surely*

$$\sum_{i \in \hat{C}_l^{11}} \sum_{j \in \hat{C}_{l'}^{21}} \hat{a}_{ij} \geq \left(1 - 3(kn^{-3/4})^{1/3}\right)^2 \frac{n^2}{k^2}(1-\delta)p$$

if $\varphi(l) = \varphi(l')$, and it holds asymptotically almost surely

$$\sum_{i \in \hat{C}_l^{11}} \hat{\sum}_{j \in \hat{C}_{l'}^{21}} \leq \left(1 + 3(kn^{-3/4})^{1/3}\right)^2 \frac{n^2}{k^2}(1+\delta)q$$

if $\varphi(l) \neq \varphi(l')$.

Proof. By construction the following two events are independent.

(1) For all $l = 1, \dots, k$ the index sets $\hat{C}_l^{11}$ are correct reconstructions of classes $C_{l'}^{11}$.

(2) For all $l = 1, \dots, k$ the index sets $\hat{C}_l^{21}$ are correct reconstructions of classes $C_{l'}^{21}$.

By Remark 5 both events happen asymptotically almost surely. By Lemma 8 the sizes of all the classes $\hat{C}_l^{11}$ and $\hat{C}_{l'}^{21}$ are a.a.s. contained in the interval

$$\left[\left(1-3(kn^{-3/4})^{1/3}\right)\frac{n}{k},\left(1+3(kn^{-3/4})^{1/3}\right)\frac{n}{k}\right].$$

Let X be the following sum of independent Poisson variables

$$\sum_{i\in\hat{C}_l^{11}}\sum_{j\in\hat{C}_{l'}^{21}}\hat{a}_{ij}.$$

Asymptotically almost surely (using Lemma 8) we have for the expectation of X,

$$E[X]\geq\left(1-3(kn^{-3/4})^{1/3}\right)^2\frac{n^2}{k^2}p$$

if $\varphi(l)=\varphi(l')$, and

$$E[X]\leq\left(1+3(kn^{-3/4})^{1/3}\right)^2\frac{n^2}{k^2}q$$

if $\varphi(l)\neq\varphi(l')$. Using Chernoff's bound we find that a.a.s.

$$X\geq\left(1-3(kn^{-3/4})^{1/3}\right)^2\frac{n^2}{k^2}(1-\delta)p$$

if $\varphi(l)=\varphi(l')$, and

$$X\leq\left(1+3(kn^{-3/4})^{1/3}\right)^2\frac{n^2}{k^2}(1+\delta)q$$

if $\varphi(l)\neq\varphi(l')$. □

Remark 6. Analogous results hold for the index sets $\hat{C}_l^{21}$ and $\hat{C}_{l'}^{12}$ with $l,l'\in\{1,\dots,k\}$ and for the index sets $\hat{C}_l^{12}$ and $\hat{C}_{l'}^{22}$ with $l,l'\in\{1,\dots,k\}$.

Proof of Theorem 4. If $\delta<(p-q)/(p+q)$ and n sufficiently large then

$$\left(1-3(kn^{-3/4})^{1/3}\right)^2\frac{n^2}{k^2}(1-\delta)p>\left(1+3(kn^{-3/4})^{1/3}\right)^2\frac{n^2}{k^2}(1+\delta)q.$$

Again in order to use this result to derive a computable threshold value we have to approximate the unknown probabilities p and q by $p^{\pm}$ and $q^{\pm}$, which are functions of the known (or almost surely known) quantities $n,k,\hat{\lambda}_1$ and $\hat{\lambda}_2$ (see Lemma 5). If we choose

$$\delta<(p^{-}-q^{+})/(p^{+}+q^{+})$$

and the threshold

$$s(n,k,\hat{\lambda}_1,\hat{\lambda}_2) = \left(1+3(kn^{-3/4})^{1/3}\right)^2 \frac{n^2}{k^2}(1+\delta)q^+$$

then the algorithm GRIDRECONSTRUCT asymptotically almost surely finds the correct reconstruction of the planted partition (up to a permutation of $\{1,\ldots,k\}$). □

3.4 Perfect reconstruction in polynomial time

In this section we also address the first quality measure, namely, perfect reconstruction. In the first part of this section we present a polynomial-time algorithm which is then analyzed in a second part.

3.4.1 The algorithm

Here is the description of our first polynomial-time algorithm to solve the planted partition reconstruction problem.

BOOSTEDRECONSTRUCT($\hat{A}$)
1 $\hat{k}, k' :=$ number of eigenvalues $\hat{\lambda}_i$ of $\hat{A}$ that are larger than $2\sqrt{n}$.
2 $\hat{p} := \frac{\hat{k}}{\hat{k}-1}\frac{\hat{\lambda}_1}{n}$
3 $\hat{q} := \frac{\hat{k}\hat{\lambda}_2 + \frac{\hat{k}}{\hat{k}-1}\hat{\lambda}_1}{n-\hat{k}}$
4 $m := n/c\log\log n$
5 Randomly partition $\{1,\ldots,n\}$ into $c\log\log n$ equal size subsets I_i.
6 $i := 1$; $\mathcal{C} := \emptyset$
7 **while** $i < c\log\log n - 1$ **do**
8 $\hat{A}_i :=$ restriction of $\hat{A}$ to I_i.
9 $\hat{P}_i :=$ projector onto the space spanned by the k' largest eigenvectors of $\hat{A}_i$.
10 $m_i := |I_i|$
11 **for** $j := 1$ **to** m_i **do**
12 **for** $l := 1$ **to** m_i **do**
13 $(c_j)_l := \begin{cases} 1 & : \ (\hat{P}_i)_{lj} \geq \frac{\hat{k}}{2m} \\ 0 & : \ \text{otherwise} \end{cases}$
14 **end for**

15 **if** $0.89\frac{m}{k} \le |c_j|^2 \le 1.11\frac{m}{k}$ **do**
16 mark the element of rank j in I_i.
17 **end if**
18 **end for**
19 **while** $\{l \in I_i : l \text{ marked}\} \neq \emptyset$ **do**
20 choose arbitrary $l \in \{l' \in I_i : l' \text{ marked}\}$ and unmark l.
21 $C_1 := \left\{l' \in I_i : l' \text{ marked}, {c_{rk_i(l)}}^T c_{rk_i(l')} \ge 0.79\frac{m}{k}\right\}$
22 **if** $0.89\frac{m}{k} \le |C_1| \le 1.11\frac{m}{k}$ **do**
23 $C_2 := \left\{l \in I_{i+1} : \sum_{l' \in C_1} \hat{a}_{ll'} \ge \frac{3}{4}|C_1|\hat{p} + \frac{1}{4}|C_1|\hat{q}\right\}$
24 **if** $|C_2| \ge \frac{3}{4}\frac{m}{k}$ **do**
25 $C_3 := \left\{l \in I_{i+2} : \sum_{l' \in C_2} \hat{a}_{ll'} \ge \frac{3}{4}|C_2|\hat{p} + \frac{1}{4}|C_2|\hat{q}\right\}$
26 $C_4 := \left\{l \in I_{i+3} : \sum_{l' \in C_3} \hat{a}_{ll'} \ge \frac{3}{4}|C_3|\hat{p} + \frac{1}{4}|C_3|\hat{q}\right\}$
27 $C_1 := \left\{l \in I_i \quad : \sum_{l' \in C_3} \hat{a}_{ll'} \ge \frac{3}{4}|C_3|\hat{p} + \frac{1}{4}|C_3|\hat{q}\right\}$
28 $C_2 := \left\{l \in I_{i+1} : \sum_{l' \in C_4} \hat{a}_{ll'} \ge \frac{3}{4}|C_4|\hat{p} + \frac{1}{4}|C_4|\hat{q}\right\}$
29 $C := C_1 \cup C_2 \cup C_3 \cup C_4$
30 **for** $j := 1$ **to** $c \log\log n$ **do**
31 **if** $j \notin \{i, i+1, i+2, i+3\}$ **do**
32 $C := C \cup \left\{l \in I_j : \sum_{l' \in C} \hat{a}_{ll'} \ge \frac{3}{4}|C|p + \frac{1}{4}|C|q\right\}$
33 **end if**
34 $I_j := I_j \setminus (I_j \cap C)$
35 **end for**
36 $\mathcal{C} := \mathcal{C} \cup \{C\}; \quad k' := k' - 1$
37 **end if**
38 **end if**
39 **end while**
40 $i := i + 4$
41 **end while**
42 **return** $\mathcal{C}$

In a nutshell the algorithm BOOSTEDRECONSTRUCT works as follows: in lines 1 to 6 we do some preprocessing. The main loop of the algorithm is enclosed by lines 7 and 41. In lines 8 to 18 we compute binary vectors c_j of size m_i that potentially are close in Hamming distance to characteristic vectors of planted classes C_ℓ restricted to the index set I_i. Note that very likely these vectors are not exactly characteristic vectors of planted classes. To account for this we have to work with four index sets $I_i, \ldots, I_{i+3}$. In lines 19 to 39 we use the vectors c_j to reconstruct whole classes, i.e., not just the restriction to $I_i \cup \ldots \cup I_{i+3}$.

Removing reconstructed elements from not yet processed index sets I_j in line 34 boosts the performance of the algorithm as a potential source of inter class noise between an already reconstructed and a not yet reconstructed class gets removed. That is, in subsequent iterations we expect to get (seen as a relative fraction of all vectors) more vectors that are close to characteristic vectors of restricted planted classes. Our subsequent analysis shows that this is indeed the case.

The running time of the algorithm is determined by $c \log\log n$ times the time to compute the k' largest eigenvectors of an $m \times m$ matrix. In the following we give a more detailed description of the algorithm.

PREPROCESSING: In line 1 we estimate the number of classes k by the number of eigenvalues of $\hat{A}$ larger than $2\sqrt{n}$, which by Lemma 4 is a.a.s. correct. In lines 2 and 3 we estimate the parameters p and q of the $A(\varphi, p, q)$ distribution from which the matrix $\hat{A}$ is drawn. By Lemma 5 these estimates almost surely converge to the true values when n goes to infinity, provided $k = o(\sqrt{n})$. In line 5 we randomly partition the index set $\{1, \ldots, n\}$ into $c \log\log n$ sets of equal size. We choose the constant c such that m and $c \log\log n$ are integers and $c \log\log n$ is divisible by 4. In line 6 we initialize the $\mathcal{C}$ that is going to contain the reconstructed classes.

MAIN LOOP: In line 9 we compute the projector $\hat{P}_i$ onto the space spanned by the eigenvectors that correspond to the k' largest eigenvalues of $\hat{A}_i$. In the variable k' we store the number of classes that still has to be reconstructed. In lines 11 to 18 we compute for every column of $\hat{P}_i$ a binary vector c_j. If we did this for the projector derived from the unperturbed adjacency matrix A_i then these vectors c_j would be the characteristic vectors of planted classes C_ℓ restricted to the index set I_i. Thus for small noise most of the vectors should be close to characteristic vectors. In lines 15 to 17 we discard c_j if it does not have the right size in order to be close to a characteristic vector. In the loop enclosed by lines 19 and 39 the actual reconstruction takes place. In lines 20 and 21 we use the binary vectors that belong to marked elements in I_i to compute a first rough reconstruction C_1 of some class C_ℓ restricted to I_i. In order to do so we have to use the rank function $\mathrm{rk}_i(\cdot)$, which gives the rank of an element in I_i, to map the elements of I_i into $\{1, \ldots, m_i\}$, i.e., the set that contains the column indices for $\hat{P}_i$. The intuition behind the computations in line 21 is that if $c_{\mathrm{rk}_i(l)}$ and $c_{\mathrm{rk}_i(l')}$ are close in Hamming distance to the characteristic vector of the same class C_ℓ restricted to I_i then their dot product should be large. Note that very likely C_1

does not contain all elements from $C_\ell \cap I_i$ and may contain elements from other classes than C_ℓ. If C_1 contains roughly as many elements as we expect to be in $C_\ell \cap I_i$ then it likely contains many elements from C_ℓ and few elements from other classes. In this case we compute in line 23 a set C_2 that contains the elements from I_{i+1} whose corresponding columns in $\hat{A}$ have many entries that are 1 at row indices in C_1. We will show that with high probability C_2 contains only elements from $C_\ell \cap I_{i+1}$, but maybe not all these elements. If C_2 is large enough then we can use it in line 23 to compute (as we computed C_2 from C_1) a set $C_3 \subset I_{i+2}$, for which we can show that asymptotically almost surely $C_3 = C_\ell \cap I_{i+2}$. Similarly we compute C_4 and re-compute C_1 from C_3 and re-compute C_2 from C_4. We will show that asymptotically almost surely it also holds that $C_1 = C_\ell \cap I_i$, $C_2 = C_\ell \cap I_{i+1}$ and $C_4 = C_\ell \cap I_{i+3}$. The reason for computing four sets is that we need some probabilistic independence in the analysis of the algorithm. In practice it probably is sufficient to re-compute C_1 from C_2 and afterwards C_2 from C_1. Similarly we will show that we compute asymptotically almost surely the set $C_\ell \cap I_j$ in line 32 and put it into the reconstruction C of C_ℓ. In line 34 we remove all elements in C from the sets I_j. In line 36 we add C to the set of reconstructed classes.

3.4.2 Analysis of the algorithm

In the following we use the notation as in the algorithm BOOSTEDRECONSTRUCT and let $C_{\ell i} = C_\ell \cap I_i$ for I_i as computed in line 5 of the algorithm. Here is a short outline of the proof, which is by induction on the number of iterations of the algorithm. The induction is anchored in Lemmas 17 and 18 and the induction step is proven in Theorem 5 and Corollary 4. But at first we state two technical lemmas, whose proofs are analogous to proofs of similar statements in the previous section, and derive a simple corollary from them.

Lemma 15. *The size of $C_{\ell 1}$ is contained in the interval*

$$\left[(1-\delta)\frac{m_1}{k}, (1+\delta)\frac{m_1}{k}\right] \quad \textit{with} \quad \delta := 3(km_1^{-3/4})^{1/3}.$$

with probability at least $1 - e^{-c'' n^{1/4}}$ for some $c'' > 0$.

Proof. See the proof of Lemma 8. □

Lemma 16. *The spectral separation $\delta_k(A_1)$ is a.a.s. at least*

$$(1-\delta)\frac{m_1}{k}(p-q),$$

where δ as in Lemma 15.

Proof. See the proof of Lemma 9. □

Corollary 3. *If $k < c'\sqrt{m}$ then a.a.s.*

$$\|P_1 - \hat{P}_1\|_2^2 < \left((1-\delta)(p-q)\frac{\sqrt{m}}{4k} - 1\right)^{-1} =: \varepsilon,$$

where δ as in Lemma 15.

Proof. Follows immediately by combining Theorem 3, Observation 1 and Lemma 16. □

In the following we will use δ and ε from Lemma 15 and Corollary 3, respectively. Note that δ goes to 0.

Dangerous element. Let $x \in C_{\ell i}$ and $r_i(x)$ be the rank of x in I_i. The element x is called *dangerous* if the $r_i(x)$'th column in the matrix $P_i - \hat{P}_i$ has more than $\frac{1}{10}\frac{m}{k}$ entries whose absolute value is at least $\frac{1}{3}\frac{k}{m}$.

Safe class. A class C_ℓ is called *safe* with respect to I_i if it satisfies the following two conditions:

(1) At most $\frac{1}{10}\frac{m}{k}$ elements of $C_{\ell i}$ are dangerous.

(2) At most $\frac{1}{10}\frac{m}{k}$ columns of $\hat{P}_i$ whose index is the rank $r_i(\cdot)$ of an element in $I_i \setminus C_\ell$ have at least $\frac{1}{10}\frac{m}{k}$ entries of value at least $\frac{1}{3}\frac{k}{m}$ at row indices that are the ranks of elements in $C_{\ell i}$.

Lemma 17. *If $k < c'\sqrt{m}$ then a.a.s. there are at least $(1 - 3600\varepsilon)k$ classes C_ℓ that are safe with respect to I_1, provided that c' is small enough such that $3600\varepsilon \ll 1$.*

Proof. The event whose probability we want to bound from above here can also be stated as: there are at most $3600\varepsilon k$ classes C_ℓ that are not safe with respect to I_1. Here it is easier to work with the latter formulation of the event. For a class to be not safe either condition (1) or condition (2) in the definition of a safe class has to be violated.

For condition (1), observe that for every dangerous element $x \in C_{\ell 1}$ the contribution of the $r_1(x)$'th column to the Frobenius norm $\|P_1 - \hat{P}_1\|_F^2$ is at least $(\frac{k}{3m})^2 \frac{m}{10k} = \frac{k}{90m}$. Hence, for a class C_ℓ to violate condition (1), the contribution to $\|P_1 - \hat{P}_1\|_F^2$ is at least $\frac{k}{90m}\frac{m}{10k} = \frac{1}{900}$. Since we have chosen c' sufficiently small Corollary 3 gives $\|P_1 - \hat{P}_1\|_2 < \varepsilon < 1$ a.a.s. This implies that a.a.s. $\|P_1 - \hat{P}_1\|_2^2 \leq \|P_1 - \hat{P}_1\|_2$. By Fact 4 applied to $P_1 - \hat{P}_1$ and given that the rank $\mathrm{rk}(P_1 - \hat{P}_1)$ is at most $2k$ with the probability stated in Observation 1, we have for the Frobenius norm $\|P_1 - \hat{P}_1\|_F^2 \leq 2\varepsilon k$ a.a.s. Thus we obtain for the number y of classes violating condition (1), $\frac{1}{900}y \leq 2\varepsilon k$ and equivalently $y \leq 1800\varepsilon k$ a.a.s.

For condition (2), let C_ℓ be a class that violates condition (2). That is, there are at least $\frac{m}{10k}$ elements $x \in I_1 \setminus C_\ell$ such that the columns with index rank $r_1(x)$ each have at least $\frac{m}{10k}$ entries of value at least $\frac{k}{3m}$ at row indices that are the ranks of elements in $C_{\ell 1}$. Hence the contribution of these columns to the Frobenius norm $\|P_1 - \hat{P}_1\|_F^2$ is at least $(\frac{k}{3m})^2 \frac{m}{10k}\frac{m}{10k} = 1/900$. If we denote by y the number of classes violating condition (2), we get using as above that $\|P_1 - \hat{P}_1\|_F^2 \leq 2\varepsilon k$ a.a.s. that $\frac{1}{900}y \leq 2\varepsilon k$ and equivalently $y \leq 1800\varepsilon k$ a.a.s.

In combination we get that a.a.s. there are at most $3600\varepsilon k$ classes which are not safe with respect to I_1. This proves the statement of the lemma. □

Lemma 18. *A.a.s., the following holds, provided* $k < c'\sqrt{m}$*:*

(1) Every class C_ℓ *that is safe with respect to* I_1 *is reconstructed correctly.*

(2) Every class C_ℓ *that is not safe with respect to* I_1 *is either reconstructed correctly or none of its elements gets removed from any index set* I_i*.*

Proof. The proof is a case analysis depending on the element l chosen in line 20 of the algorithm, where $l \in C_{\ell 1}$ for some $\ell \in \{1, \ldots, k\}$.

Case 1. Assume that C_ℓ is safe with respect to I_1 and that l is not dangerous. By Lemma 15, a.a.s. $(1-\delta)\frac{m}{k} \leq |C_{\ell 1}| \leq (1+\delta)\frac{m}{k}$. Hence, a.a.s., by the definition of dangerous in line 21 of the algorithm all other not dangerous elements in $C_{\ell 1}$ are put into C_1 and all other not dangerous elements in $I_1 \setminus C_{\ell 1}$ are not put into C_1. Also, by the definitions of dangerous element and safe class, a.a.s. there are at most $\frac{m}{10k}$ elements $l' \in I_1 \setminus C_{\ell 1}$ for which it holds that ${c_{rk(l)}}^T c_{rk(l')} \geq 0.79\frac{m}{k}$. Hence, a.a.s. $0.89\frac{m}{k} \leq |C_1| \leq 1.11\frac{m}{k}$ and C_1 contains at least $0.89\frac{m}{k}$ elements of C_ℓ. That implies that a.a.s. the fraction of elements in C_1 belonging to C_ℓ is strictly larger than $\frac{3}{4}$. Since the entries in $\hat{A}$ with index in $I_1 \times I_2$ all are independent Poisson trials we get by using Chernoff's bound that a.a.s. every column of $\hat{A}$ with index in $C_{\ell 2}$ has more than $\frac{3}{4}|C_1|p + \frac{1}{4}|C_1|q$ entries that are 1 at row indices in C_1. By Lemma 5 and our assumption that $k < c'\sqrt{m}$ we can approximate p and q asymptotically arbitrarily well by $\hat{p}$ and $\hat{q}$, respectively. Hence we even have that this number of 1 entries is a.a.s. larger than $\frac{3}{4}|C_1|\hat{p} + \frac{1}{4}|C_1|\hat{q}$. Likewise, every column with index in $I_2 \setminus C_\ell$ a.a.s. has less than $\frac{3}{4}|C_1|\hat{p} + \frac{1}{4}|C_1|\hat{q}$ entries that are 1 at row indices in C_1. Hence, a.a.s. $C_2 = C_{\ell 2}$. Applying Lemma 15 to $C_{\ell 2}$, we get that a.a.s. $|C_2| \geq \frac{3}{4}\frac{m}{k}$. Applying a similar Chernoff bound argument as for $C_{\ell 2}$ we get that a.a.s. in lines 25-28 (note that by construction all entries of $\hat{A}$ that we use in these computations are probabilistically independent of the ones that we used in previous computations) and in line 32 of the algorithm all elements of $C_{\ell j}, j = 1, \ldots, c \log\log n$ and no other elements are put into C. Thus C is a.a.s. the correct reconstruction of C_ℓ.

Case 2. Assume that C_ℓ is safe with respect to I_1 and that l is dangerous. In this case it is possible that only some elements in $C_{\ell 1}$ are put into C_1 in line 21 of the algorithm, and it is also possible that elements in $I_1 \setminus C_{\ell 1}$ are put into C_1. If not $0.89\frac{m}{k} \leq |C_1| \leq 1.11\frac{m}{k}$ then no new class is produced by the algorithm. Otherwise, by using Chernoff's bound for independent Poisson trials and the approximation guarantee for p and q by $\hat{p}$ and $\hat{q}$, respectively, we have that if a column of $\hat{A}$ with index $x \in I_2$ such that $\varphi(x) = \ell'$ for some $\ell' \in \{1, \ldots, k\}$ has more than $\frac{3}{4}|C_1|\hat{p} + \frac{1}{4}|C_1|\hat{q}$ entries 1 at row indices in C_1 then a.a.s. at least $\frac{2}{3}$ of the elements in C_1 are also mapped to ℓ' by φ. Hence, a.a.s. there is at most one class ℓ' whose elements in I_2 are put into C_2 in line 23 of the algorithm. Thus, C_2 a.a.s. contains only elements of $C_{\ell' 2}$. If $|C_2| \geq \frac{3}{4}\frac{m}{k}$, again using Chernoff's bound, C_3 a.a.s. contains all elements of $C_{\ell' 3}$ and no other elements. Similarly we can argue for

C_4, C_1, C_2 and C. Thus C is a.a.s. the correct reconstruction of $C_{\ell'}$, where ℓ' may be different from ℓ.

Case 3. Assume that C_ℓ is not safe with respect to I_1 and that l is not dangerous. By the same arguments that we used in the analysis of the first case C_1 as computed in line 21 a.a.s. contains only elements of $C_{\ell 1}$. Thus, if the test in line 22 is passed then as in the analysis of the first case a.a.s. $C = C_\ell$. Otherwise no new class is computed.

Case 4. Assume that C_ℓ is not safe with respect to I_1 and that l is dangerous. The analysis of this case is the same as for the second case.

In any case whenever a new class C is computed by the algorithm then a.a.s. $C = C_\ell$ for some $\ell \in \{1, \dots, k\}$, i.e., we reconstruct C_ℓ correctly. Furthermore, any safe class C_ℓ gets reconstructed since at some point l chosen in line 20 has to be a not dangerous element in $C_{\ell 1}$, i.e., we are in the first case. □

Lemma 18 basically states that after the first iteration of the outer while-loop of the algorithm all classes C_ℓ that are safe with respect to I_1 (but possibly even more classes) are reconstructed correctly. Thus we get from Lemma 17 that after the first iteration of the outer while-loop a.a.s. at least $(1 - 3600\varepsilon)k$ classes are reconstructed correctly, or equivalently a.a.s. at most $3600\varepsilon k$ classes remain to be reconstructed.

Theorem 5. *A.a.s., after the j'th iteration of the outer while loop of the algorithm* BOOSTEDRECONSTRUCT *there remain at most*

$$(3600\varepsilon')^{2(1.5^j-1)}k \quad \textit{with} \quad \varepsilon' = \frac{4\sqrt{m(1+\delta)}}{\frac{m}{k}(p-q)(1-\delta) - 4\sqrt{m(1+\delta)}}$$

classes to be reconstructed and all other classes have been reconstructed correctly, provided $k < c'\sqrt{m}$ and c' sufficiently small.

Proof. The proof is by induction on j. For $j = 1$, by Lemma 17 and Lemma 18, after the first step, a.a.s. at most $3600\varepsilon k \leq 3600\varepsilon' k$ classes remain to be reconstructed. Thus the statement of the theorem holds for $j = 1$. The induction hypothesis is that after the $(j-1)$'th iteration of the outer while-loop a.a.s. at most $k' \leq (3600\varepsilon')^{2(1.5^{j-1}-1)}k =: k''$ classes remain to be reconstructed and all other classes have been reconstructed correctly. Now we study the j'th iteration of the outer while-loop. The value of i within this iteration is $i = 4(j-1)+1$. By the induction hypothesis we have a.a.s. that for any so far not reconstructed

class C_ℓ it holds that $C_{\ell i} = C_\ell \cap I_i$. This did not change during the previous iterations of the outer while-loop, i.e., the only elements removed from I_i belong to correctly reconstructed classes. Thus we can apply Lemma 15 to get a.a.s. $(1-\delta)\frac{m}{k} \leq |C_{\ell i}| \leq (1+\delta)\frac{m}{k}$. This implies that a.a.s. we have

$$k'(1-\delta)\frac{m}{k} \leq m_i = |I_i| \leq k''(1+\delta)\frac{m}{k}.$$

If we adapt the proof of Lemma 16 with m_i instead of m_1 and k' instead of k then a.a.s. we find for the spectral separation $\delta_{k'}(A_i)$, where A_i is the restriction of A to I_i,

$$\delta_{k'}(A_i) \geq (1-\delta)\frac{m}{k}(p-q).$$

Let P_i be the projector onto the space spanned by the k' largest eigenvectors of A_i. Applying Theorem 3 and Observation 1 to the matrix $P_i - \hat{P}_i$, we get a.a.s. by plugging in the bounds on m_i and k',

$$\begin{aligned}
\|P_i - \hat{P}_i\|_2 &\leq \frac{4\sqrt{m_i}}{\delta_{k'}(A_i) - 4\sqrt{m_i}} \\
&\leq \frac{4\sqrt{k''(1+\delta)\frac{m}{k}}}{\frac{m}{k}(p-q)(1-\delta) - 4\sqrt{k''(1+\delta)\frac{m}{k}}} \\
&= \frac{4\sqrt{(3600\varepsilon')^{2(1.5^{j-1}-1)}m(1+\delta)}}{\frac{m}{k}(p-q)(1-\delta) - 4\sqrt{(3600\varepsilon')^{2(1.5^{j-1}-1)}m(1+\delta)}} \\
&\leq \frac{4\sqrt{(3600\varepsilon')^{2(1.5^{j-1}-1)}m(1+\delta)}}{\frac{m}{k}(p-q)(1-\delta) - 4\sqrt{m(1+\delta)}} \\
&= \varepsilon'\sqrt{(3600\varepsilon')^{2(1.5^{j-1}-1)}}.
\end{aligned}$$

The last expression is less than 1 if $k < c'\sqrt{m}$ and c' is sufficiently small. Thus we also have a.a.s.

$$\|P_i - \hat{P}_i\|_2^2 \leq \varepsilon'\sqrt{(3600\varepsilon')^{2(1.5^{j-1}-1)}}.$$

With the probability that Observation 1 holds we have that the rank of

$P_i - \hat{P}_i$ is at most $2k''$. Thus we get a.a.s. for the Frobenius norm

$$\begin{aligned}
\|P_i - \hat{P}_i\|_F^2 &\leq \mathrm{rk}(P_i - \hat{P}_i)\|P_i - \hat{P}_i\|_2^2 \\
&\leq 2(3600\varepsilon')^{2(1.5^{j-1}-1)}k\varepsilon'\sqrt{(3600\varepsilon')^{2(1.5^{j-1}-1)}} \\
&= 2\varepsilon' k\left((3600\varepsilon')^{2(1.5^{j-1}-1)}\right)^{1.5} \\
&= 2\varepsilon' k\left((3600\varepsilon')^{2(1.5^{j}-1.5)}\right).
\end{aligned}$$

We use this bound to show that a.a.s. at most $1800(3600\varepsilon')^{2(1.5^j-1.5)}\varepsilon' k$ not yet reconstructed classes violate condition (1) of a safe class with respect to I_i and at most $1800(3600\varepsilon')^{2(1.5^j-1.5)}\varepsilon' k$ classes violate condition (2). By a similar argument as in the proof of Lemma 17 we can show that both events occur a.a.s., because if any of the two events does not hold we get

$$\begin{aligned}
\|P_i - \hat{P}_i\|_F^2 &> 1800(3600\varepsilon')^{2(1.5^j-1.5)}\varepsilon' k\left(\frac{k}{3m}\right)^2 \frac{m}{10k}\frac{m}{10k} \\
&= 2\varepsilon' k(3600\varepsilon')^{2(1.5^j-1.5)},
\end{aligned}$$

which a.a.s. does not happen. Hence, a.a.s. after the $(j-1)$'th iteration of the outer while-loop at most $(3600\varepsilon')^{2(1.5^j-1)}k$ of the not yet reconstructed classes are not safe with respect to I_i. We can now argue as in the proof of Lemma 18 that a.a.s. all not yet reconstructed classes that are safe with respect to I_i get correctly reconstructed in the j'th iteration of the outer while-loop. A not safe class a.a.s. either also gets correctly reconstructed or none of its elements gets removed from any index set $I_\iota, \iota = 1, \ldots, c\log\log n$. That is, a.a.s. after the j'th iteration of the outer while-loop at most $(3600\varepsilon')^{2(1.5^j-1)}k$ classes remain to be reconstructed and all other classes have been reconstructed correctly. That is, what we had to prove. □

Note that in the statement of Theorem 5 the boosting reflects itself in the double exponential dependence of the decrease factor on j. Without boosting we would only have a single exponential dependence on j. A careful look into the proof of Theorem 5 and its preceding lemmas shows that the statement holds not just a.a.s., but it holds with probability $1 - e^{-cn^\alpha}$ for some $c, \alpha > 0$. Therefore, for the j as in Theorem 5 we

can also plug in the value $j = \Theta(\log\log n)$ and we obtain the following corollary.

Corollary 4. *A.a.s., the algorithm* BOOSTEDRECONSTRUCT *reconstructs all* k *classes correctly, provided* $k \leq c' \frac{\sqrt{n}}{\log\log n}$ *for a small enough constant* c'.

Proof. By Theorem 5, we have that after the $\left(\frac{c}{4}\log\log n\right)$'th iteration of the outer while-loop, a.a.s. the number of not yet reconstructed classes is at most $(3600\varepsilon')^{2(1.5^{\frac{c}{4}\log\log n}-1)}k$. If c' is small enough then we have $3600\varepsilon' \ll 1$ and

$$\lim_{n\to\infty} (3600\varepsilon')^{2(1.5^{\frac{c}{4}\log\log n}-1)}k = 0.$$

Hence, for n sufficiently large, $(3600\varepsilon')^{2(1.5^{\frac{c}{4}\log\log n}-1)}k$ is strictly less than 1. Since the number of not yet reconstructed classes is an integer, it has to be 0 for n large enough. Thus a.a.s. all classes are reconstructed correctly. □

3.5 Bounding the number of misclassifications

In contrast to the previous two sections, in this section we address the second quality measure, namely, bounding the number of misclassifications. As before, we present a polynomial-time algorithm in the first part of this section and its analysis in the second part.

3.5.1 The algorithm

Here is the description of our second polynomial-time spectral algorithm to solve the planted partition reconstruction problem.

BOUNDEDERRORRECONSTRUCT($\hat{A}$)
1 k' := number of eigenvalues of $\hat{A}$ that are larger than $2\sqrt{n}$.
2 $\hat{P}$:= projection matrix computed from the k' largest eigenvectors $v_1, \ldots, v_{k'}$ of $\hat{A}$.
3 **for** $i = 1$ **to** n **do**
4 R_i := set of row indices which are among the $\frac{n}{k'}$ largest entries

of the i'th column of $\hat{P}$.
5 **for** $j = 1$ **to** n **do**
6 $c_{ij} := \begin{cases} 1, & j \in R_i \\ 0, & \text{else} \end{cases}$
7 **end for**
8 $c_i := (c_{i1}, \ldots, c_{in})^T$
9 **end for**
10 $I := \{1, \ldots, n\}$; $l := 1$
11 **while** exists an unmarked index $i \in I$ **do**
12 $C_l := \emptyset$
13 **for each** $j \in I$ **do**
14 **if** $c_i^T c_j > \frac{4n}{5k'}$ **do**
15 $C_l := C_l \cup \{j\}$
16 **end if**
17 **end for**
18 **if** $|C_l| \geq \left(1 - \sqrt{\frac{320\sqrt{n}}{\hat{\lambda}_2 - 6\sqrt{n}}}\right) \frac{n}{k'}$ **do**
19 $I := I \setminus C_l$; $l := l + 1$
20 **else**
21 mark index i.
22 **end if**
23 **end while**
24 $C_l := I$
25 **return** $C_1, \ldots, C_l$

In line 1 the number of planted classes k' is estimated. The estimate is motivated by Lemma 4. In line 2 the projection matrix $\hat{P}$ that belongs to $\hat{A}$ is computed. From line 3 to line 9 for each column i of $\hat{A}$ a vector $c_i \in \{0,1\}^n$ with exactly $\frac{n}{k'}$ entries that are 1 is computed. In lines 10 to 24 the actual partitioning takes place. Roughly speaking, two indices i, j are put into the same class if the Hamming distance of the corresponding vectors c_i and c_j is small (test in line 14). A class as created in lines 12 to 17 is not allowed to be too small (test in line 18), otherwise its elements get distributed into other classes that are going to be constructed in future executions of the body of the while-loop. Notice that the algorithm runs in time polynomial in n and only makes use of quantities that can be deduced from $\hat{A}$, i.e., it does not need to know the values of p, q and k.

3.5.2 Analysis of the algorithm

Stable vector. A vector $c_i \in \{0,1\}^n$ as produced by the algorithm BOUNDEDERRORRECONSTRUCT is called *stable* with respect to φ if more than $\frac{9}{10}$ of the indices in $\{1,\dots,n\}$ that correspond to entries that are 1 in c_i are mapped by φ to $\varphi(i)$, i.e., all these elements belong to the same class. A vector c_i is called *unstable* if it is not stable.

Lemma 19. *In the algorithm* BOUNDEDERRORRECONSTRUCT *with probability at least* $1 - e^{-(1-o(1))\sigma^2 n/8}$ *at most*

$$\frac{320n}{\frac{\sqrt{n}}{k}(p-q)-4}$$

vectors $c_i \in \{0,1\}^n$ *are constructed that are unstable if* $k < \frac{p-q}{8}\sqrt{n}$.

Proof. Let x be the number of unstable vectors c_i that are computed within the algorithm BOUNDEDERRORRECONSTRUCT from the projection matrix $\hat{P}$. If c_i is an unstable vector then at least $\frac{n}{10k}$ of the $\frac{n}{k}$ largest entries in the i'th column of $\hat{P}$ correspond to row indices $j \in \{1,\dots,n\}$ such that the entries p_{ij} in P are zero, i.e., these entries are not among the $\frac{n}{k}$ largest entries in the i'th column of P. That is, at least $x\frac{n}{10k}$ of the large entries in P become small entries in $\hat{P}$, i.e., they do no longer belong to the $\frac{n}{k}$ largest entries in their column. We denote the number of such entries by y and can bound it by using the Frobenius norm of the matrix $P - \hat{P}$. In order to bound the Frobenius norm of $P - \hat{P}$ we first bound its L_2 norm,

$$\begin{aligned}\|P-\hat{P}\|_2 &\leq \frac{2\|A-\hat{A}\|_2}{\delta_k(A)-2\|A-\hat{A}\|_2} = \frac{2\|A-\hat{A}\|_2}{\frac{n}{k}(p-q)-2\|A-\hat{A}\|_2} \\ &\leq \frac{4\sqrt{n}}{\frac{n}{k}(p-q)-4\sqrt{n}} = \frac{4}{\frac{\sqrt{n}}{k}(p-q)-4} < 1,\end{aligned}$$

where we use Theorem 3 in the first inequality, Definition 13 in the first equality, Observation 1 in the second inequality and our assumption on k in the last inequality. Note, that the second inequality only holds with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$. From Lemma 4 it follows that the k' chosen in line 1 of the algorithm BOUNDEDERRORRECONSTRUCT is exactly k with probability at least $1-e^{-(1-o(1))\sigma^2 n/8}$. Hence the rank

of $P - \hat{P}$ is at most $2k$ with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$. That gives

$$\|P - \hat{P}\|_F^2 \leq 2k\|P - \hat{P}\|_2^2 < 2k\|P - \hat{P}\|_2 \leq \frac{8k}{\frac{\sqrt{n}}{k}(p-q) - 4},$$

where the first inequality only holds with probability at least

$$1 - e^{-(1-o(1))\sigma^2 n/8}.$$

In order for a large entry in P to become a small entry in $\hat{P}$ this entry must become at least as small in $\hat{P}$ as some other entry in the same column which is zero in P. The number of large/small pairs in a column of P that become small/large pairs in $\hat{P}$ can be maximized for a given bound on the Frobenius norm $\|P - \hat{P}\|_F^2$ if the large entry, which is $\frac{k}{n}$ in P, and the small entry, which is zero in P, both become $\frac{k}{2n}$ in $\hat{P}$. By this argument the number y of such pairs can be bounded from above by

$$(\frac{k}{2n})^2 y \leq \frac{8k}{\frac{\sqrt{n}}{k}(p-q) - 4}, \quad \text{that is} \quad y \leq \frac{32\frac{n^2}{k}}{\frac{\sqrt{n}}{k}(p-q) - 4}.$$

Putting everything together we get

$$x\frac{n}{10k} \leq y \leq \frac{32\frac{n^2}{k}}{\frac{\sqrt{n}}{k}(p-q) - 4}, \quad \text{that is} \quad x \leq \frac{320n}{\frac{\sqrt{n}}{k}(p-q) - 4}.$$

This inequality holds with the same probability that the bound on the Frobenius norm $\|P - \hat{P}\|_F^2$ holds. The latter probability is at least $1 - e^{-(1-o(1))\sigma^2 n/8}$. □

Notation. To shorten our exposition we set in the following

$$\alpha = \frac{320}{\frac{\sqrt{n}}{k}(p-q) - 4}.$$

Lemma 20. *With probability at least* $1 - e^{-(1-o(1))\sigma^2 n/8}$ *in at least* $(1 - \sqrt{\alpha})k$ *classes we have at least* $(1 - \sqrt{\alpha})\frac{n}{k}$ *associated stable vectors if* $k < \frac{p-q}{8}\sqrt{n}$.

Proof. The proof of the lemma is equivalent to showing that the complementary event that there are less than $(1-\sqrt{\alpha})k$ classes with at least $(1-\sqrt{\alpha})\frac{n}{k}$ associated stable vectors occurs with probability at most $e^{-(1-o(1))\sigma^2 n/8}$. In case of the complementary event there are more than $\sqrt{\alpha}k$ classes, which contain more than $\sqrt{\alpha}\frac{n}{k}$ unstable vectors each. Thus we get in total more than $\sqrt{\alpha}k\sqrt{\alpha}\frac{n}{k} = \alpha n$ unstable vectors. By Lemma 19, however, this happens only with probability at most $e^{-(1-o(1))\sigma^2 n/8}$. □

Covered vectors and split classes. A vector c_i *covers* a vector c_j and vice versa if $c_i^T c_j > \frac{4n}{5k}$. A class $C = \varphi^{-1}(l), l \in \{1,\dots,k\}$ is split by an unstable vector c_j if there exists a stable vector c_i with $i \in C$ that is covered by c_j. An unstable vector c_h *almost splits* C if it does not split C, but there exists an unstable vector c_j which splits C and covers c_h.

Lemma 21. *Every unstable vector can split or almost split at most one class.*

Proof. Let c_j be an unstable vector and let c_i be a stable vector covered by c_j, i.e., $c_i^T c_j > \frac{4n}{5k}$. By the definition of stable vectors more than $\frac{9}{10}\frac{n}{k}$ of the indices corresponding to the one entries in c_i are mapped by φ to $\varphi(i)$. That is, at least

$$\left(\frac{4}{5}-\frac{1}{10}\right)\frac{n}{k} = \frac{7}{10}\frac{n}{k} > \frac{n}{2k}$$

of the indices corresponding to the one entries of c_j are mapped by φ to $\varphi(i)$. That shows there cannot be another stable vector c_h with $\varphi(i) \neq \varphi(h)$ covered by c_j. Thus c_j can split at most one class.

It remains to show that an unstable vector can almost split at most one class. Let c_j be an unstable vector and let c_i be an unstable vector that splits a class and is covered by c_j. That is, $c_i^T c_j > \frac{4n}{5k}$ and there exists a stable vector c_h such that at least $\frac{7}{10}\frac{n}{k}$ of the indices corresponding to the one entries of c_i are mapped by φ to $\varphi(h)$. That is, more than

$$\left(\frac{7}{10}-\frac{1}{5}\right)\frac{n}{k} = \frac{n}{2k}$$

of the indices corresponding to the one entries of c_j are mapped by φ to $\varphi(h)$. That shows that c_j can split or almost split at most one class. □

Lemma 22. *With probability at least* $1 - e^{-(1-o(1))\sigma^2 n/8}$ *at most* $\sqrt{\alpha}k$ *classes are split or almost split by more than* $\sqrt{\alpha}\frac{n}{k}$ *unstable vectors, provided* $k < \frac{p-q}{8}\sqrt{n}$.

Proof. The proof of the lemma is equivalent to showing that the complementary event that more than $\sqrt{\alpha}k$ classes are split or almost split by more than $\sqrt{\alpha}\frac{n}{k}$ unstable vectors occurs with probability at most $e^{-(1-o(1))\sigma^2 n/8}$. If this is the case, since by Lemma 21 every unstable vector splits or almost splits at most one class, the total number of unstable vectors that split or almost split a class is more than

$$\sqrt{\alpha}k\sqrt{\alpha}\frac{n}{k} = \alpha n.$$

That is, the number of unstable vectors is more than αn. By Lemma 19, however, this happens only with probability at most $e^{-(1-o(1))\sigma^2 n/8}$. □

Stable class. A class $\varphi^{-1}(m), m \in \{1, \ldots, k\}$ is called stable if it contains more than $(1 - \sqrt{\alpha})\frac{n}{k}$ indices of stable vectors and if it is split or almost split by at most $\sqrt{\alpha}\frac{n}{k}$ unstable vectors.

Lemma 23. *For any stable class* $C = \varphi^{-1}(m), m \in \{1, \ldots, k\}$*, with probability at least* $1 - e^{-(1-o(1))\sigma^2 n/8}$*, the algorithm* BOUNDEDERRORRECONSTRUCT *outputs a class* C_l *that contains at least*

$$\left(1 - \sqrt{\alpha} - \sqrt{2\alpha}\right)\frac{n}{k}$$

indices in C *corresponding to stable vectors, provided* $k < \frac{p-q}{16}\sqrt{n}$.

Proof. Let c_i be vector which is used in line 14 of the algorithm BOUNDEDERRORRECONSTRUCT to create a class C_l. A stable vector c_j with $j \in C$ can only be put into the class C_l if either c_i is another stable vector with $i \in C$ or if c_i is an unstable vector that splits C. We discuss the two cases now.

Assume c_i is a stable vector with $i \in C$. Then all stable vectors c_h with $h \in C$ will also be drawn into C_l since we have

$$c_i^T c_h > \left(1 - 2\frac{1}{10}\right)\frac{n}{k} = \frac{4n}{5k}.$$

That is, C_l will contain all stable vectors whose index is in C. It remains to show that C_l will pass the test in line 18 of the algorithm. But this follows from our definition of stable class and

$$\begin{aligned} 1-\sqrt{\alpha} &= 1-\sqrt{\frac{320}{\frac{\sqrt{n}}{k}(p-q)-4}} \\ &\geq 1-\sqrt{\frac{320}{\frac{\sqrt{n}}{k}\left(\hat{\lambda}_2\frac{k}{n}-\frac{2k}{\sqrt{n}}\right)-4}} \\ &= 1-\sqrt{\frac{320\sqrt{n}}{\hat{\lambda}_2-6\sqrt{n}}}, \end{aligned}$$

where we used the lower bound on $p-q$ from Lemma 6.

Now assume that c_i is an unstable vector that splits C. Then c_i can draw some of the stable vectors whose index is in C into C_l and it can draw some unstable vectors that either split or almost split C. Assume that C_l passes the test in line 18. Since by the definition of a stable class it can draw at most $\sqrt{\alpha}\frac{n}{k}$ unstable vectors, it has to draw at least

$$\left(1-\sqrt{\frac{320\sqrt{n}}{\hat{\lambda}_2-6\sqrt{n}}}\right)\frac{n}{k}-\sqrt{\alpha}\frac{n}{k}$$

stable vectors with an index in C. Using that with probability at least $1-e^{-(1-o(1))\sigma^2 n/8}$ it holds

$$\frac{n}{k}(p-q)-p-2\sqrt{n}\leq\hat{\lambda}_2,$$

see Lemma 6, we find that with the same probability

$$\begin{aligned} 1-\sqrt{\frac{320\sqrt{n}}{\hat{\lambda}_2-6\sqrt{n}}} &\geq 1-\sqrt{\frac{320}{\frac{\sqrt{n}}{k}(p-q)-\frac{p}{\sqrt{n}}-8}} \\ &\geq 1-\sqrt{\frac{320}{\frac{1}{2}\left(\frac{\sqrt{n}}{k}(p-q)-4\right)}} = 1-\sqrt{2\alpha}, \end{aligned}$$

where we used

$$\frac{1}{2}\frac{\sqrt{n}}{k}(p-q)\geq\frac{p}{\sqrt{n}}+6,$$

which follows from $k < \frac{p-q}{16}\sqrt{n}$. Combining everything we get that C_l with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$ contains at least

$$\left(1 - \sqrt{\alpha} - \sqrt{2\alpha}\right) \frac{n}{k}$$

indices in C corresponding to stable vectors. □

Theorem 6. *With probability at least* $1 - e^{-(1-o(1))\sigma^2 n/8}$ *at most*

$$\left(3\sqrt{\alpha} + \sqrt{2\alpha}\right) n$$

indices are misclassified by the algorithm BOUNDEDERRORRECONSTRUCT *provided* $k < \frac{p-q}{16}\sqrt{n}$.

Proof. By combining Lemmas 20 and 22 we get with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$ at least

$$\left(1 - \sqrt{\alpha} - \sqrt{\alpha}\right) k = \left(1 - 2\sqrt{\alpha}\right) k$$

stable classes. From each stable class at least

$$\left(1 - \sqrt{\alpha} - \sqrt{2\alpha}\right) \frac{n}{k}$$

indices of stable vectors are grouped together by the algorithm with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$, see Lemma 23. Thus in total with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$ at least

$$\left(1 - \sqrt{\alpha} - \sqrt{2\alpha}\right) \frac{n}{k} \left(1 - 2\sqrt{\alpha}\right) k > \left(1 - 3\sqrt{\alpha} - \sqrt{2\alpha}\right) n$$

indices of stable vectors are grouped together correctly. Hence, by the definition of the number of misclassifications via a maximum weight matching, with probability at least $1 - e^{-(1-o(1))\sigma^2 n/8}$ at most

$$\left(3\sqrt{\alpha} + \sqrt{2\alpha}\right) n$$

elements are misclassified. □

Discussion. Note that the theorem is non-trivial only if

$$k < \frac{p-q}{3524 + 1920\sqrt{2}}\sqrt{n}.$$

The theorem implies that if $k = o(\sqrt{n})$ then the relative number of misclassifications goes to zero asymptotically almost surely as n goes to infinity. That is the first non-trivial polynomial-time result for $k = \omega\left(\frac{\sqrt{n}}{\log\log n}\right)$. But also in the case $k = c\sqrt{n}$ for a small constant c the theorem provides useful information. It basically says that on average the percentage of elements per class that get misclassified by the algorithm becomes arbitrarily small if c is small enough.

3.6 A lower bound for minimality of k-partition

In this section we give, in contrast to the previous three sections, some sort of "negative" result: we show that for a certain value of k there is no algorithm that can recover the planted k-partition a.a.s. Intuitively, it is clear that it becomes harder to recover the planted k-partition when the difference between p and q gets smaller and when k gets bigger. If we assume as before, that both p and q are constant values, $p > q$ (and thus the difference $p - q$ is in this case also constant), the only parameter left to control the difficulty of the problem is k. If the planted k-partition is not minimal anymore, one can not hope to find an algorithm that recovers this planted k-partition asymptotically almost surely. To prove this statement in a precise way, we will make use of the following theorem ([Bol01], Theorem 1.5):

Theorem 7. *Let* $pn \geq 1$ *and* $m := pn + h < n$*, where* $h > 0$*. Define* $\beta := \frac{1}{12m} + \frac{1}{12(n-m)}$ *and* $\eta := \frac{1}{\sqrt{2\pi p(1-p)n}}$*. Then*

$$\binom{n}{m} p^m (1-p)^{n-m} > \eta\, e^{-\frac{h^2}{2p(1-p)n} - \frac{h^3}{2(1-p)^2 n^2} - \frac{h^4}{3p^3 n^3} - \frac{h}{2pn} - \beta}.$$

In particular, we will prove the following theorem:

Theorem 8. *Let* p *and* q *to be fixed constants,* $p > q$*. For* $k \geq c\frac{n}{\log n}$ *(for some constant* $c = c(p,q) > 0$*), a.a.s. the planted* k*-partition is not a minimum* k*-partition anymore.*

Proof. The proof follows the idea of [CO06] where it is proven that if $p(n)$ and $q(n)$ satisfy $p(n) - q(n) = o(\sqrt{\frac{p \log n}{n}})$, a planted bisection is a.a.s. not a minimal bisection anymore. To keep the exposition of the proof simpler, we fix $p = 0.5$ and $q = 0.25$ and prove Theorem 8 for this special case. For this choice of p and q, we can prove the theorem with the particular value of $c = 7$ (no attempt has been made to optimize c). For other fixed values of p and q, only constants change. Assume w.l.o.g. that $k \in \omega(1) \cap o(n)$.

To prove Theorem 8, define the indicator variable C_{ij} to be 1 if two vertices v_i and v_j are such that $v_i \in P_r$, $v_j \in P_s$, $P_r \neq P_s$, and if additionally the swap of vertex v_i with vertex v_j decreases the total number of edges going between vertices of different partition classes. We have

$$\Pr[C_{ij} = 1] \geq \Pr[E_1 = 1 \wedge E_2 = 1 \wedge E_3 = 1 \wedge E_4 = 1], \tag{3.1}$$

where the events E_i have the following meaning: E_1 is the indicator event that $|N(v_i) \cap P_r| \leq (\frac{n}{k} - 1)p + (\frac{n}{k} - 1)\zeta$, E_2 is the indicator event that $|N(v_i) \cap (P_s \setminus \{v_j\})| \geq (\frac{n}{k} - 1)p + (\frac{n}{k} - 1)\zeta + 1$, E_3 is the indicator event that $|N(v_j) \cap P_s| \leq (\frac{n}{k} - 1)p + (\frac{n}{k} - 1)\zeta$, and E_4 is the indicator event that $|N(v_j) \cap (P_r \setminus \{v_i\})| \geq (\frac{n}{k} - 1)p + (\frac{n}{k} - 1)\zeta + 1$; the value of ζ is fixed below.
Since the (non)edge joining vertices v_i and v_j is excluded from all events, all these four events are independent, and by symmetry,

$$\Pr[E_1 = 1 \wedge E_2 = 1 \wedge E_3 = 1 \wedge E_4 = 1] = (\Pr[E_1 = 1])^2 (\Pr[E_2 = 1])^2. \tag{3.2}$$

Now, to compute $\Pr[E_1 = 1]$, choose $\zeta = 0.1p$ (note that 0.1 is also a constant that might have to be changed for a large p). Clearly, all the edges between v_i and other vertices inside P_r are independent Bernoulli trials with parameter p. Therefore

$$E[|N(v_i) \cap P_r|] = (\frac{n}{k} - 1)p,$$

and since the total variance is $(\frac{n}{k} - 1)p(1 - p)$, for $k \in o(n)$, by Chebyshev's inequality (Lemma 1), $\Pr[E_1 = 1]$ tends to 1 for n sufficiently large.

To compute $\Pr[E_2 = 1]$, note first

$$\begin{aligned} & \Pr\left[|N(v_i) \cap (P_s \setminus \{v_j\})| \geq (\tfrac{n}{k} - 1)p + \zeta(\tfrac{n}{k} - 1) + 1\right] \\ \geq\ & \Pr\left[(\tfrac{n}{k} - 1)(p + \zeta) + 1 \leq |N(v_i) \cap (P_s \setminus \{v_j\})| \leq (1 - \zeta)(\tfrac{n}{k} - 1)\right] \\ =\ & \sum_{i=(\frac{n}{k}-1)p+\zeta(\frac{n}{k}-1)+1}^{(1-\zeta)(\frac{n}{k}-1)} \binom{n/k-1}{i} q^i (1-q)^{n/k-1-i}, \end{aligned}$$

where the upper bound of the summation index in the second line is used for technical reasons to apply Theorem 7 for each term of the sum. Indeed, set $n := \frac{n}{k} - 1$, and $h_i := i - (\frac{n}{k} - 1)q$, for $i = (\frac{n}{k} - 1)p + \zeta(\frac{n}{k} - 1) + 1, \ldots, (1 - \zeta)(\frac{n}{k} - 1)$. Choosing $\zeta = 0.1p$ as before and recalling that $p = 0.5$ and $q = 0.25$, we have $h_i \leq (\frac{n}{k} - 1)0.7$, for $i = (\frac{n}{k} - 1)p + \zeta(\frac{n}{k} - 1) + 1, \ldots, (1 - \zeta)(\frac{n}{k} - 1)$. Now for each term, by Theorem 7 (with q playing the role of p in the statement of the theorem) and the upper bound on h_i we have for some constants $c, c' > 0$

$$\begin{aligned} \binom{n/k-1}{i} q^i (1-q)^{n/k-1-i} &> c\sqrt{\frac{k}{n}} e^{-\frac{0.7^2 n}{(3/8)k} - \frac{0.7^3 n}{(9/8)k} - \frac{0.7^4 n}{(3/64)k} - O(1)} \\ &\geq c'\sqrt{\frac{k}{n}} e^{-\frac{37877n}{5625k}}. \end{aligned}$$

Since there are $\Theta\left(\frac{n}{k}\right)$ terms that are summed up, we get

$$\Pr[E_2 = 1] \geq \Omega\left(\sqrt{\frac{n}{k}} e^{-\frac{37877n}{5625k}}\right). \tag{3.3}$$

Plugging in (3.3) into (3.2) and then (3.2) into (3.1) and noting that $\Pr[E_1 = 1]$ tends to 1 we obtain

$$\Pr[C_{ij} = 1] \geq \Omega\left(\left(\sqrt{\frac{n}{k}} e^{-\frac{37877n}{5625k}}\right)^2\right).$$

Using our assumption that $k \geq 7n/\log n$, the term $\Omega\left(\left(\sqrt{\frac{n}{k}} e^{-\frac{37877n}{5625k}}\right)^2\right)$ is at least $\Omega\left(n^{-75754/39375}\right)$. Now, defining $C := \sum_{i<j} C_{ij}$ we get by linearity of expectation

$$E[C] = \sum_{i<j} E[C_{ij}] = \binom{k}{2} (\frac{n}{k})^2 \Pr[C_{ij} = 1] \sim \frac{n^2}{2} \Pr[C_{ij} = 1].$$

Again, by the previously established bounds and the assumption on k, we get for n sufficiently large that

$$E[C] \geq \Omega\left(n^{2-\frac{75754}{39375}}\right) = \Omega\left(n^{2996/39375}\right).$$

To show that $C > 0$ a.a.s. we will bound the variance of C, i.e., we apply the second moment method. We compute $E[C^2]$:

$$E[C^2] = \sum_{i<j} \sum_{l<m} \Pr[C_{ij} = 1 \wedge C_{lm} = 1] \tag{3.4}$$

Now, if all four vertices v_i, v_j, v_l and v_m are different and additionally all four vertices belong to different partition classes, then

$$\Pr[C_{ij} = 1 \wedge C_{lm} = 1] = \Pr[C_{ij} = 1]\Pr[C_{lm} = 1].$$

We now split the terms contributing to $E[C^2]$ on the right hand side of (3.4) and first consider those 4-tuples of vertices which all belong to different partition classes. For these we have

$$\begin{aligned}
&\sum_{i<j} \sum_{l<m} \Pr[C_{ij} = 1 \wedge C_{lm} = 1] \\
&= \binom{k}{2}\binom{k-2}{2}(\frac{n}{k})^4 \Pr[C_{ij} = 1]\Pr[C_{lm} = 1] \\
&= (E[C])^2(1 + o(1)).
\end{aligned}$$

If all four vertices are different but they belong to at most three different partition classes,

$$\begin{aligned}
&\sum_{i<j} \sum_{l<m} \Pr[C_{ij} = 1 \wedge C_{lm} = 1] \\
&= \Theta(n^4/k)\Pr[C_{ij} = 1 \wedge C_{lm} = 1] \\
&= \Theta(n^4/k)\left(\Pr[C_{ij} = 1]\,(1 - o(1))\right)^2,
\end{aligned}$$

where the latter probability estimate follows from the fact that the two events are "almost" independent in the sense that there is at most a constant number of edges which is counted twice and there are in total

$\omega(1)$ edges going between a vertex and a partition class (since $k \in o(n)$ we have that $\frac{n}{k} \in \omega(1)$). Moreover, since $k \in \omega(1)$, we get

$$\Theta(n^4/k)\left(\Pr[C_{ij} = 1]\,(1 - o(1))\right)^2 \quad = \quad o\left(E[C]\right)^2.$$

If only three of the vertices v_i, v_j, v_l and v_m are different and they are in three different partition classes, then

$$\begin{aligned}
&\sum_{i<j}\sum_{l<m} \Pr[C_{ij} = 1 \wedge C_{lm} = 1] \\
&\quad = \quad \Theta(n^3)\Pr[C_{ij} = 1 \wedge C_{lm} = 1] \\
&\quad \leq \quad (E[C])^{1.5} = o\left((E[C])^2\right),
\end{aligned}$$

where the last inequality follows from the fact that in this case for the probability $\Pr[C_{ij} = 1 \wedge C_{lm} = 1]$ at least three vertices have to have more neighbours in another partition than in its own whereas for $\Pr[C_{ij} = 1]$ only two vertices are involved. Also, by the same argument of "almost" independence (now applied when just three vertices are involved) as before, if only three vertices are different and they are in two different partition classes,

$$\begin{aligned}
&\sum_{i<j}\sum_{l<m} \Pr[C_{ij} = 1 \wedge C_{lm} = 1] \\
&\quad = \quad \Theta(n^3/k)\Pr[C_{ij} = 1 \wedge C_{lm} = 1] \\
&\quad \leq \quad (E[C])^{1.5} = o\left((E[C])^2\right).
\end{aligned}$$

Finally, if there are only two different vertices, then

$$\begin{aligned}
&\sum_{i<j}\sum_{l<m} \Pr[C_{ij} = 1 \wedge C_{lm} = 1] \\
&\quad = \quad \binom{k}{2}\left(\frac{n}{k}\right)^2 \Pr[C_{ij} = 1] \\
&\quad = \quad E[C] = o\left((E[C])^2\right).
\end{aligned}$$

Combining everything and plugging in all terms into (3.4) we get

$$E[C^2] = (E[C])^2 + o\left((E[C])^2\right)$$

and thus $\mathrm{var}[C] = o\left((E[C])^2\right)$. By Chebyshev's inequality (Lemma 1), for $k \geq 7n/\log n$, $C > 0$ a.a.s. Since $C > 0$ is a sufficient condition for the event that a planted k-partition is not optimal, we have proven Theorem 8. □

3.7 Concluding remarks

We presented and analyzed three spectral partitioning algorithms. The analysis of all three algorithms provided non-trivial guarantees for a range of parameters where such guarantees were not known before. As we have presented it, all three algorithms and their analysis are restricted to the case when all classes have exactly the same size. It is an interesting question whether classes of different sizes can be handled in the same way. For the second algorithm it would also be interesting to see whether the simple technique of sampling and iteration that we used here is applicable to boost the performance of other partitioning and clustering algorithms, for example the ones described in [McS01, ST02]. On the other hand, we have shown that for fixed p and q and $k \geq cn/\log n$ for some constant $c = c(p, q)$, the planted k-partition is a.a.s. not a minimal k-partition anymore. Thus, for $k \geq cn/\log n$, no algorithm is able to recover a planted k-partition a.a.s. The natural open question is to design and analyze polynomial or moderately growing superpolynomial algorithms that could work for a bigger value of k and/or to show that even for a smaller value of k, the planted k-partition is a.a.s. not minimal anymore.

Chapter 4

Collaborative ranking

4.1 Introduction

Preference elicitation is daily practice in market research. Its goal is to assess a person's preferences concerning a set of products or a distribution of preferences over a population. The products in the set are usually substitute goods or services, i.e., products that serve the same purpose and can replace each other.

In applications like market share prediction estimating the distribution of the population's preferences is enough whereas in other applications like recommendation systems, see for example [DKR02, KRRT98, RV97], one needs to have a good estimate of an individual's preferences. Preferences can be captured by a value function that assigns to every product a value. Every value function induces a ranking of the products. The higher the value of a product, the higher is the product's position in the ranking. A ranking in general contains less information than a value function. But the direct assessment of a person's value function is difficult and often leads to unreliable results. Ranking products is an easier task, especially if the ranking is obtained from pairwise comparisons (or more general choice tasks), which are popular in market research because they often simulate real buying situations. In many applications a ranking of the products is enough. Sometimes it is even enough to determine the highest ranking product. But often one is also interested in the ordering of the in-between products. For example the

highest ranking product for a group of persons might be costly to produce. If it turns out that there is a product which is much cheaper to produce but is still high up in the ranking of most of the persons, then it might be reasonable to produce the latter product.

Here we study the problem of eliciting product preferences from respondents based on pairwise comparison questionnaires. Eliciting a respondent's ranking would require the respondent to perform $\Theta(n \log n)$ comparisons, where n is the number of products. This might be too much if n exceeds a certain value. If the number of comparisons the respondents have to perform exceeds their "tolerance threshold" they either cancel the whole interview or stop to do the comparisons carefully. The tolerance threshold does not scale with n but is usually a small constant. Therefore we want to ask only a constant number of questions but increase the number of respondents for larger product sets. According to their answers we assign the respondents to consumer types. In the end we compute the rankings for the consumer types from the sparse input.

Related work. Our approach is similar in spirit to work of Drineas, Kerenidis and Raghavan [DKR02] who studied recommendation systems. In their work a complete utility function has to be elicited for a small number of persons. For the remaining persons only a few utility values have to be assessed to give a good product recommendation. In our work, however, all respondents are asked only a few questions. The problem of how to assess utility values directly is not addressed in the paper of Drineas et al. The algorithm used in [DKR02] is similar to our algorithm in the sense that it also uses spectral methods. It is different however since the matrix which serves as a basis for classification of the users is different from ours. Moreover, the classification of the users follows a different procedure, and also the products are not ranked at all.

Another line of work related to ours is in the area of collaborative filtering (see for example [GNOT92, KS04, KS03]), where the main goal is the preference prediction from historical data whereas our goal is to elicit preference data. Among the three papers mentionned [KS04] is the only one which also uses spectral methods. The methods applied by [KS04] are however rather different from ours (their algorithm is based on the 1-norm of some columns of a certain correlation matrix, and these columns are chosen to maximize a certain form of indepen-

dence between them).

Other papers that also fall into the category of spectral methods for data analysis include [McS01,AFKS01,AFKM01,DKF+04,CDG+99,PRTV98]. However, a straightforward spectral approach to our problem is very difficult to analyze since it builds on a random matrix whose entries are not pairwise independent. In order to avoid these difficulties we turn to a more sophisticated procedure for which we can provide (up to constants) as good guarantees as can be expected.

4.2 Ranking algorithm

Given a set of products, we want persons (respondents) to rank the products by pairwise comparisons. Note that other choice tasks, e.g., a one out of three choice task, can be interpreted as multiple paired comparisons. Often, for example in a web based scenario, we can ask each respondent to compare only a small number of products since after a certain number of questions, which is independent of the number of products, respondents tend to stop answering the questions carefully. Our approach is to aggregate the answers obtained from the respondents in order to obtain a few rankings that represent the population of respondents very well. That is, we assume that the population can be segmented into a small number of (consumer) types, i.e., persons that belong to the same consumer type have similar preferences and thus rank the products similarly whereas the rankings of two persons that belong to different types differ substantially. We call the expected ranking (mean ranking) of a user type to be the *typical* ranking of that user type. At first we describe how we want to elicit preferences.

Elicitation procedure. Let X be a set of n products. We want to infer a ranking of X for consumers who have to answer l different paired comparison questions (one out of two choice tasks) chosen independently at random, i.e., for every respondent we choose a subset of size l of product pairs in $\binom{X}{2}$ independently at random. We refer to the consumers who have to perform the choice tasks as respondents. Let m be the number of respondents. Note that here l is a constant independent of n whereas m is dependent on n (m being the only non-constant parameter, n is also considered as a constant).

Next we give an outline of our ranking algorithm. We refer to any $L \subseteq \binom{X}{2}$ of l product pairs shortly as l-tuple. In the following it turns out to be convenient to consider each product pair as an ordered pair (x, y) such that we can refer to x as the first product and y as the second product of this pair. Our algorithm has four phases. In the first phase we choose a random l-tuple L and then segment all respondents that did all comparisons corresponding to the pairs in L into types of similar preference. In the second phase we use the segmentation of this subset of respondents to compute typical partial rankings of the products covered by the l-tuple L for each segment. In the third phase we extend the partial rankings to complete rankings of all products for all the consumer types that we determined in the second phase. Finally in the fourth and last phase, we also segment all the respondents into their respective consumer types that have not been segmented before.

(1) Segmenting the respondents for an l-tuple. Given a randomly chosen l-tuple L the algorithm SEGMENTRESPONDENTS segments all respondents that did all comparisons corresponding to the pairs in L into types of similar preference. The algorithm has two parameters:

(1) A parameter $0 < \alpha < 1$ that imposes a lower bound of αm on the size of the smallest consumer type that it can identify. Respondents from smaller types will be scattered among other types.

(2) A symmetric $(m_L + l) \times (m_L + l)$-matrix $B = (b_{ij})$ that contains the data collected from the m_L respondents, where m_L is the number of respondents that did all comparisons corresponding to the pairs in L. The column and row indices $1, \ldots, m_L$ of B are indexed by the corresponding respondents and the column and row indices $m_L + 1, \ldots, m_L + l$ are indexed by the l ordered product pairs in L. For $i \in \{1, \ldots, m_L\}$, $j \in \{m_L + 1, \ldots, m_L + l\}$, we set $b_{ji} := b_{ij} := -1$, if respondent i prefers in the $(j - m_L)$'th product comparison the second product over the first one, and $b_{ji} := b_{ij} := 1$, if he prefers the first product over the second. All other entries b_{ij} are set to 0.

SEGMENTRESPONDENTS(B, α)
1 $k :=$ number of eigenvalues of B that are larger than some threshold depending on m, n, and l.

2 $P_B :=$ projector onto the eigenspace corresponding to the k most positive and the k most negative eigenvalues of B.
3 $P'_B :=$ restriction of P_B onto its first m_L columns and m_L rows.
4 **for** $r := 1$ **to** m_L **do**
5 **for** $s := 1$ **to** m_L **do**
6 $(c_r)_s := \begin{cases} 1 & : \ (P'_B)_{rs} \geq 0.49\binom{n}{2}/m \\ 0 & : \ \text{otherwise} \end{cases}$
7 **end for**
8 **if** $0.99\alpha m \leq |c_r|^2\binom{n}{2}$ **do**
9 mark r.
10 **end if**
11 **end for**
12 $I := \{1, \ldots, m_L\}$, $\mathcal{C} := \emptyset$
13 **while** $\{j \in I : j \text{ marked}\} \neq \emptyset$ **do**
14 unmark arbitrarily chosen $i \in \{j \in I : j \text{ marked}\}$
15 $C := \left\{ j \in I : j \text{ marked}, \ \frac{\langle c_i, c_j \rangle}{|c_i||c_j|} \geq 0.97 \right\}$
16 $C' := \left\{ j \in I : j \text{ marked}, \ \frac{\langle c_i, c_j \rangle}{|c_i||c_j|} \geq 0.8 \right\}$
17 **if** $|C|\binom{n}{2} \geq 0.9\alpha m$ **and** $|C'|\binom{n}{2} \leq |C|\binom{n}{2} + 0.02\alpha m$ **do**
18 $I := I \setminus C$
19 $\mathcal{C} := \mathcal{C} \cup \{C\}$
20 **end if**
21 **end while**
22 **return** $\mathcal{C}$

The threshold in line 1 of the algorithm SEGMENTRESPONDENTS is used to estimate the number of different consumer types using a carefully chosen threshold that only depends on the known quantities m, n and l. In line 3 the projector is restricted to its first m_L rows and first m_L columns, since this allows us to identify the columns of P'_B with respondents. From each column i of P'_B we compute a vector $c_i \in \{0,1\}^{m_L}$ whose j'th entry is 1 if and only if the corresponding entry b_{ij} in the matrix B is not less than $0.49\binom{n}{2}/m$, see line 6. We also check whether c_i does not contain a too small number of 1 entries. The intuition is that c_i is close to a characteristic vector of a *typical* consumer type. This idea is exploited in the **while**-loop enclosed by lines 13 and 21, where two respondents are grouped together if their corresponding vectors c_i and c_j make a small angle, see lines 15 and 16. The use of two sets

C and C' is there to avoid taking a vector c_i that is too far from any characteristic vector of a typical type. All vectors which are never put into any set C will be discarded.

(2) Computing partial rankings for the segments. For each consumer type computed by the algorithm SEGMENTRESPONDENTS we determine a typical partial ranking of the products covered by the l-tuple L simply by majority vote. For every ordered product pair $(x, y) \in L$ we say that a type prefers x over y if more than half of the respondents of this type has stated that they prefer x over y, otherwise we say the consumer type prefers y over x.

(3) Extending the partial rankings. For a consumer type to extend the partial ranking of the products covered by L to all products in X, we proceed as follows: we replace an arbitrary element $j \in L$ by an element in $X \setminus \bigcup_{Y \in L} Y$. Let L' be the resulting set of pairs. We run the algorithm SEGMENTRESPONDENTS on L' to segment all the respondents that did all comparisons corresponding to the pairs in L'. The segments of respondents computed from L and from L' will then be merged and the replacement process is repeated until the typical rankings for all $\binom{n}{2}$ product pairs are determined.

(4) Segmenting all respondents. Finally, all not yet classified respondents, i.e., those respondents corresponding to an l-tuple not used to determine the typical rankings get also classified. Assume that such a respondent did pairwise product comparisons for pairs in the l-tuple L^*. Such a respondent is classified to be of that consumer type whose ranking restricted to L^* best matches the answers he provided (ties broken arbitrarily).

Let us summarize all parameters of our ranking algorithm in the following table.

n	number of products
m	number of respondents (dependent on n)
l	number of comparisons performed by each respondent (independent of n)
α	parameter in $(0,1)$ that poses a lower bound of αm on the size of the smallest consumer type that can be identified

4.3 Statistical model

The ranking algorithm that we presented in the last section is formulated independently of a model of population and respondents. But in order to theoretically analyze any procedure that computes typical rankings from the input data a model of the population and a model of the respondents is necessary. As in Chapter 3, providing such a model gives rise to a *reconstruction problem*, namely, given data obeying the model the task is to reconstruct the model parameters.

Population model. We assume that the population can be partitioned into k types. Let $\alpha_i \in (0,1)$ be the fraction of the i'th type in the whole population. For each type there is a ranking σ_i, $i = 1,\dots,k$, i.e., a permutation, of the n products in X. It will be convenient to encode a ranking of X as a vector u with $\binom{n}{2}$ components in $\{\pm 1\}$, one component for each product pair (x,y). In the vector the entry at position (x,y) is 1 if x is preferred over y and -1 otherwise. We will refer to the vector u also simply as ranking. As a measure of separation of two permutations we use the Hamming distance which is the number of inverted pairs, i.e., the number of pairs (x,y), $x,y \in X$, where $x \prec y$ in the one permutation and $y \prec x$ in the other permutation. Here $x \prec y$ means that y is ranked higher than x. Obviously the maximum separation of two permutations is $\binom{n}{2}$.

Respondent model. We assume that the set of respondents faithfully represents the population, i.e., α_i is also (roughly) the fraction of respondents of type i among all respondents. Given a respondent let σ be the ranking that corresponds to the type of this respondent. For any comparison of products x and y, with $x \prec y$ according to σ, we

assume that the respondent states his preference of y over x with probability $p > 1/2$. In our model each comparison is a random experiment, independently from all other comparisons, with success probability p, where success means that a respondent answers according to his type. Note that this allows the respondents' answers to violate transitivity, i.e., we still ask comparisons whose outcome could have been derived already from transitivity. From a practical perspective this seems meaningful since stated preferences are often not transitive and the amount of "non-transitivity" is interesting extra information that could be exploited otherwise.

Let us summarize all parameters of our model in the following table.

k	number of types
p	probability for a respondent not to deviate from its type when performing a comparison
$\delta(u_i, u_j)$	separation of typical rankings (vectors) u_i and u_j
α_i	fraction of type i-respondents among all respondents
m_i	$\alpha_i m$, i.e., number of respondents of the i'th type

This model allows to analyze procedures that compute typical rankings from the input data. It leads to the ranking reconstruction problem.

Ranking reconstruction problem. Given the data obtained by our elicitation procedure (see previous section) from a population that follows the model described above, the ranking reconstruction problem asks to reconstruct the number k of consumer types, their corresponding typical rankings and to associate every (many) respondent(s) with his (their) correct type(s).

4.4 Analysis of the algorithm

The reconstruction problem becomes harder if the typical rankings are not well separated. To make this more precise we introduce the following notions of well-separation.

Well-separation. For $0 < \epsilon < 1$ we say that the typical consumer types are ϵ-*well-separated* (or just *well-separated* if ϵ is understood) if for any two different consumer type rankings u_i and u_j, we have

$$(1-\epsilon)\binom{n}{2} \leq 2\,\delta(u_i, u_j) \leq (1+\epsilon)\binom{n}{2}.$$

Given a ranking (vector) u and an l-tuple L, we denote the projection of u onto L by $\pi_L(u)$, i.e., $\pi_L(u)$ is a vector in $\{\pm 1\}^l$. We say that an l-tuple L is ϵ-*well-separating* (or just *well-separating*), if for any two different consumer types with associated rankings u_i and u_j we have

$$|\langle \pi_L(u_i), \pi_L(u_j)\rangle| \leq 3\epsilon l.$$

In this case, we also say that consumer types i and j are ϵ-well-separated (or just well-separated) by L.

Even for well-separated typical rankings it will not always be possible to solve the ranking reconstruction problem, especially if the number of respondents m is too small compared to the number of products n. In the following we show that with high probability (in fact our statements below hold with probability $1-ce^{-100}$ or the like, but c as well as 100 were chosen arbitrarily and can be improved to any arbitrary constant) our ranking algorithm solves the ranking reconstruction problem approximately if:

(1) All parameters besides m are considered constant and the number m of respondents is large enough as some function of n.

(2) The number of comparisons l depends on α, p and ϵ, i.e., it is larger when α (fraction of smallest type) becomes smaller, or when the non-error probability p gets closer to $1/2$, or when ϵ (well-separation of the typical types) becomes smaller. Note however that l does not depend on n, and l is assumed to be much smaller than $\binom{n}{2}$.

We start by showing that a randomly chosen l-tuple is well-separating with high probability.

Lemma 24. *Suppose that the typical consumer type rankings are ϵ-well-separated and suppose that $l = l(p, \alpha, \epsilon) \ll \binom{n}{2}$ is a large constant but independent of n. Then with probability at least $1 - e^{-100}$, a randomly chosen l-tuple is ϵ-well-separating.*

Proof. We fix some order on the $\binom{k}{2}$ consumer type pairs. Let

$$\{u_1, v_1\}, \ldots, \left\{u_{\binom{k}{2}}, v_{\binom{k}{2}}\right\}$$

be the corresponding sequence of consumer type ranking pairs. We will first bound the probability that for $\{u_1, v_1\}$ we have

$$|\langle \pi_L(u_1), \pi_L(v_1)\rangle| > 3\epsilon l$$

for a randomly chosen l-tuple L. For $r \in L$ define the indicator variable C_1^r to be 1 if the consumer type rankings u_1 and v_1 agree in their preference on product pair r, and 0 otherwise. Set $C_1 = \sum_{r\in L} C_1^r$. Similarly, define D_1^r to be 1 if the consumer type rankings u_1 and v_1 disagree in their preference on r, and 0 otherwise. Set $D_1 = \sum_{r\in L} D_1^r$. Note that $C_1 = l - D_1$.

We may assume that the product pairs in L were chosen sequentially. The i'th product pair is chosen uniformly at random from all $\binom{n}{2}-(i-1)$ remaining product pairs. Hence, the probability to choose a product pair on which u_1 and v_1 agree depends on the previously chosen comparisons. However, in any step, the probability to choose an element on which their preference agrees is at least

$$p_\uparrow := \frac{(1-\epsilon)\binom{n}{2} - 2(l-1)}{2\binom{n}{2}},$$

since by our assumption on $\delta(u_1, v_1)$ in any step there are at least $\big((1-\epsilon)\binom{n}{2} - 2(l-1)\big)/2$ product pairs remaining on which u_1 and v_1 agree and there are at most $\binom{n}{2}$ product pairs to be chosen from.

We now couple the random variable C_1 with an auxiliary random variable $\hat{C}$ which is the sum of l independent 0/1-variables $\hat{C}^1, \ldots, \hat{C}^l$ with $\Pr[\hat{C}^i = 1] = p_\uparrow$ for $i = 1, \ldots, l$, i.e.,

$$\hat{C} = \sum_{i=1}^{l} \hat{C}^i.$$

By linearity of expectation we have $E[\hat{C}] = p_\uparrow l$ and by Chernoff bounds (Corollary 1),

$$\begin{aligned}
\Pr[C_1 \le (1-\gamma)p_\uparrow l] &\le \Pr[\hat{C} \le (1-\gamma)p_\uparrow l] \\
&= \Pr[\hat{C} \le (1-\gamma)E[\hat{C}]] \\
&\le e^{-\gamma^2 E[\hat{C}]/4} = e^{-\gamma^2 p_\uparrow l/4}.
\end{aligned}$$

Hence with probability at least $1 - e^{-\gamma^2 p_\uparrow l/4}$ we get that $C_1 > (1 - \gamma)p_\uparrow l$. By the same reasoning, the probability that u_1 and v_1 agree in preference on a randomly chosen product pair is in every step at most $p_\downarrow := \frac{(1+\epsilon)\binom{n}{2}}{2\binom{n}{2}-2(l-1)}$. Again, by coupling, we can see that with probability at least $1 - e^{-\gamma^2 p_\downarrow l/4}$ we get that $C_1 < (1 + \gamma)lp_\downarrow$. Combining both bounds, we see that with probability at least $1 - 2e^{-\gamma^2 \min\{p_\uparrow, p_\downarrow\}l/4}$ we have

$$(1 - \gamma)lp_\uparrow < C_1 < (1 + \gamma)lp_\downarrow.$$

Consequently we have with probability at least $1 - 2e^{-\gamma^2 \min\{p_\uparrow, p_\downarrow\}l/4}$

$$(1 - p_\downarrow(1 + \gamma))l < D_1 < (1 - p_\uparrow(1 - \gamma))l.$$

Since by our assumption, $l \ll \binom{n}{2}$, we can choose $\gamma = \gamma(\epsilon) > 0$ small enough such that we have

$$\begin{aligned} & |C_1 - D_1| \\ \leq\ & \max\{(1 + \gamma)lp_\downarrow - (1 - p_\downarrow(1 + \gamma))l, (1 - p_\uparrow(1 - \gamma))l - (1 - \gamma)lp_\uparrow\} \\ =\ & \max\{(2(1 + \gamma)p_\downarrow - 1)l, (1 - 2(1 - \gamma)p_\uparrow)l\} \\ \leq\ & 3\epsilon l \end{aligned}$$

with probability at least $1 - 2e^{-\gamma^2 \min\{p_\uparrow, p_\downarrow\}l/4}$. Hence with at least the same probability we have

$$|\langle \pi_L(u_1), \pi_L(v_1)\rangle| \leq 3\epsilon l.$$

We can interpret $(1 - 2e^{-\gamma^2 \min\{p_\uparrow, p_\downarrow\}l/4})$ as a lower bound on the fraction of all l-tuples on which u_1 and v_1 are well-separated. Consequently,

$$\beta = 2e^{-\gamma^2 \min\{p_\uparrow, p_\downarrow\}l/4}$$

is an upper bound on the fraction of l-tuples on which u_1 and v_1 (or any fixed pair) are not well-separated.

Next we determine the probability that all pairs of different consumer types are well-separated by L. Unfortunately the events for different pairs of consumer types are not independent. Thus for a fixed pair of consumer types we have to condition the probability under the event that all previous pairs are well-separated by L. Let us first consider the

second pair $\{u_2, v_2\}$. In the worst case all the l-tuples on which u_2 and v_2 are not well-separated are well-separating for $\{u_1, v_1\}$. Let ρ denote the exact fraction of l-tuples that are well-separating for $\{u_1, v_1\}$, i.e.,

$$\Pr\big[|\langle u_1, v_1\rangle| \leq 3\epsilon l\big] = \rho.$$

Then the fraction of l-tuples that are well-separating for both $\{u_1, v_1\}$ and $\{u_2, v_2\}$ is at least $(1 - 2\beta)$, and from the definition of conditional probabilities we get

$$\Pr\big[|\langle u_2, v_2\rangle| \leq 3\epsilon l \big| |\langle u_1, v_1\rangle| \leq 3\epsilon l\big] \geq (1 - 2\beta)/\rho.$$

Hence we get

$$\Pr\big[|\langle u_1, v_1\rangle| \leq 3\epsilon l\big] \Pr\big[|\langle u_2, v_2\rangle| \leq 3\epsilon l \big| |\langle u_1, v_1\rangle| \leq 3\epsilon l\big] \geq 1 - 2\beta.$$

Iterating this argument we get that

$$\Pr\big[L \text{ is well-separating}\big] \geq 1 - \binom{k}{2}\beta.$$

Plugging in our assumption $l = l(\alpha, p, \epsilon) \geq ck^2$, we get by choosing c sufficiently large, that with probability at least $1 - 2\binom{k}{2}e^{-c'k^2} \geq 1 - e^{-100}$ a randomly chosen l-tuple L is well-separating. □

Remark 7. *The upper bound on $\delta(u, v)$ is primarily needed due to technical reasons of the analysis. We think that in practice a lower bound should suffice.*

For some l-tuple L let m_L^i denote the number of consumers of type $i \in \{1, \ldots, k\}$ that compared exactly the l product pairs in L. We show next that all m_L^i are reasonably large. Note that we assume that $m_i \geq \alpha m$ for all consumer types i. Denote also by m_L the total number of consumers that compared exactly the l product pairs in L.

Lemma 25. *Given $\xi > 0$. For any l-tuple L and any consumer type i it holds*

$$(1 - \xi)m_i \leq \binom{n}{2}{l} m_L^i \leq (1 + \xi)m_i$$

with probability at least $1 - e^{-cm}$ for some $c = c(\xi) > 0$.

Proof. Since we assign l-tuples to respondents uniformly at random we get that the expected number of consumers of fixed type i assigned to a fixed l-tuple L is

$$m^i / \binom{\binom{n}{2}}{l}.$$

Using Chernoff bounds, we get that

$$(1-\xi)m^i \leq \binom{\binom{n}{2}}{l} m_L^i \leq (1+\xi)m^i \tag{4.1}$$

with probability at least $1 - 2e^{-c(\xi)m}$ for some $c(\xi) > 0$. Taking a union bound over all possible l-tuples and over all k consumer types we get that (4.1) holds for all l-tuples L and all consumer types i with probability at least

$$1 - k\binom{\binom{n}{2}}{l} e^{-c'(\xi)\alpha m} = 1 - e^{-c(\xi)m}$$

for some $c'(\xi), c(\xi) > 0$, if we use our lower bound αm on any of the m^i. □

In order to estimate the number of consumer types k in line 1 of the algorithm SEGMENTRESPONDENTS we want to compute the largest eigenvalues in absolute value of B (see Section 4.2 for the definition of B). We want to exploit the block structure of B, which can be written as

$$B = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix},$$

where A is an $m_L \times l$ matrix whose rows are indexed by respondents and whose columns are indexed by product pairs. We are not going to compute the eigenvalues of B directly but use a perturbation argument to estimate them. Therefore we compare B with $\hat{B}$, which we define as the matrix of expected values for the entries in B, i.e., $\hat{b}_{ij} = E[b_{ij}]$. In particular that means that for $i \in \{1, \ldots, m_L\}$ and $j \in \{m_L+1, \ldots, m_L+l\}$ we have $\hat{b}_{ji} = \hat{b}_{ij} = 2p - 1$, if respondent i prefers in the $(j - m_L)$'th product comparison the first product over the second with probability p, $\hat{b}_{ji} = \hat{b}_{ij} = 1 - 2p$, if he prefers the second product over the first with probability p, and $\hat{b}_{ij} = 0$ otherwise. Note that the structure of $\hat{B}$ is similar to the structure of B, namely,

$$\hat{B} = \begin{pmatrix} 0 & \hat{A} \\ \hat{A}^T & 0 \end{pmatrix}.$$

Observation 2. *The matrix* B *has a spectrum symmetric to* 0.

Proof. Let $\lambda \neq 0$ be an eigenvalue of B and $(v, u) \in \mathbb{R}^{m_L+l}$ be an corresponding eigenvector, where $v \in \mathbb{R}^{m_L}$ and $u \in \mathbb{R}^l$. Using the structure of B we get

$$Au = \lambda v \text{ and } A^T v = \lambda u.$$

For the vector $(v, -u)$ we get

$$B(v, -u) = (-Au, A^T v) = -\lambda(v, -u),$$

which shows that also $-\lambda$ is an eigenvalue of B. □

Now we can prove that B and $\hat{B}$ are close.

Lemma 26.

$$\|B - \hat{B}\|_2 \leq 4\sqrt{m_L + l} < 5\sqrt{m_L}$$

with probability at least $1 - e^{-cm}$ *for some* $c > 0$.

Proof. We have for the variance of $b_{ij} - \hat{b}_{ij}$ for any index pair ij:

$$\begin{aligned} \nu ar[b_{ij} - \hat{b}_{ij}] &= E[(b_{ij} - \hat{b}_{ij})^2] - E[b_{ij} - \hat{b}_{ij}]^2 \\ &= E[b_{ij}^2] - E[\hat{b}_{ij}^2] - 0 \\ &= 1 - (2p - 1)^2 \; = \; 4p - 4p^2 \; \leq \; 1. \end{aligned}$$

Using Chernoff bounds we have $m_L = \Theta(m)$ with probability $1 - e^{-c'm}$ for some $c' > 0$. Note that here we consider l and n as constant. Hence by Corollary 2,

$$\|B - \hat{B}\|_2 \leq 4\sqrt{m_L + l} < 5\sqrt{m_L}$$

with probability at least $1 - 2e^{-c''(m_L+l)} = 1 - e^{-cm}$ for some $c'', c > 0$. □

The eigenvalues of $\hat{B}$ can be computed from their squared values, i.e., from the eigenvalues of $\hat{B}^2$. The matrix $\hat{B}^2$ has the form

$$\hat{B}^2 = \begin{pmatrix} \hat{A}\hat{A}^T & 0 \\ 0 & \hat{A}^T\hat{A} \end{pmatrix},$$

Thus the eigenvalues of $\hat{B}^2$ can be computed as the eigenvalues of the $m_L \times m_L$ matrix $\hat{A}\hat{A}^T$ and the $l \times l$ matrix $\hat{A}^T\hat{A}$, respectively. Furthermore, $\hat{A}\hat{A}^T$ and $\hat{A}^T\hat{A}$ have the same non-zero eigenvalues. The largest k eigenvalues of $\hat{A}\hat{A}^T$ are estimated in the following lemma. For the sake of simplicity in this lemma and the corollary immediately afterwards we assume that the l-tuple L is well-separating; only in Theorem 9 later the probability of this event will be included.

Lemma 27. *For* $i = 1, \ldots, k$ *it holds that*

$$\lambda_i(\hat{A}\hat{A}^T)\binom{n}{2} \geq 9lm(2p-1)^2\alpha/10,$$

with probability at least $1-e^{-cm}$ *for some* $c > 0$*, if* $\epsilon \leq (2p-1)^2\alpha^3/90000$.

Proof. We decompose the $m_L \times m_L$ matrix $\hat{A}\hat{A}^T$ as follows

$$\hat{A}\hat{A}^T = C + D + E,$$

which are defined as follows: $c_{rs} = (\hat{A}\hat{A}^T)_{rs}$ if respondents r and s belong to the same consumer type and 0 otherwise, $d_{rs} = (\hat{A}\hat{A}^T)_{rs}$ if r and s belong to different types and $(\hat{A}\hat{A}^T)_{rs} \geq 0$ and 0 otherwise, and finally $e_{rs} = (\hat{A}\hat{A}^T)_{rs}$ if r and s belong to different consumer types and $(\hat{A}\hat{A}^T)_{rs} < 0$, and 0 otherwise. Now, C is a rank k matrix whose k nonzero eigenvalues are $l(2p-1)^2m_L^i$, $i = 1, \ldots, k$. Thus by Lemma 25 and by the lower bound on the size of any consumer type these eigenvalues are with probability at least $1 - e^{-cm}$, for some $c > 0$, larger than

$$l(2p-1)^2(1-\xi)m_i/\binom{n}{2} \geq l(2p-1)^2(1-\xi)\alpha m/\binom{n}{2},$$

for some $\xi > 0$.

Since all entries in D and E have the same sign, their largest eigenvalue in absolute value is at most the maximum of row or column sums of the absolute values of their entries. Since L is assumed to be well-separating the absolute value of every entry in D and E, respectively, is at most $3\epsilon l$. Thus, the largest eigenvalue in absolute value of D and E, respectively, is at most $3\epsilon l m_L$. We will use a weaker upper bound of $4\epsilon l m_L$ here, since the lemma is also applied in the analysis of the phase *Extending the partial rankings* of our algorithm to an l-tuple

which might be "almost" well-separating, but not quite. By applying Theorem 2 twice we get for $i = 1, \ldots, k$,

$$\begin{aligned}\lambda_i(\hat{A}\hat{A}^T) &\geq \lambda_i(C) + \lambda_{m_L}(D) + \lambda_{m_L}(E) \\ &\geq l(2p-1)^2(1-\xi)\alpha m / \binom{\binom{n}{2}}{l} - 8\epsilon l m_L\end{aligned}$$

with probability at least $1 - e^{-cm}$. Since by Chernoff bounds, with probability at least $1 - e^{-c'm}$, $m_L \leq (1+\xi)m/\binom{\binom{n}{2}}{l}$, we have

$$\lambda_i(\hat{A}\hat{A}^T) \geq \left(lm((2p-1)^2(1-\xi)\alpha - 8(1+\xi)\epsilon)\right) / \binom{\binom{n}{2}}{l}$$

with probability at least $1 - e^{-cm}$ for some $c > 0$. Plugging in our assumption

$$\epsilon \leq (2p-1)^2 0.01^2 \alpha^3/9 < 0.01(2p-1)^2\alpha$$

we get for sufficiently small ξ that for $i = 1, \ldots, k$,

$$\lambda_i(\hat{A}\hat{A}^T)\binom{\binom{n}{2}}{l} \geq lm(2p-1)^2\alpha 0.9,$$

with probability at least $1 - e^{-cm}$. □

Corollary 5. *The matrix $\hat{B}$ has rank $2k$ with probability at least $1-e^{-cm}$.*

Proof. Since $\hat{A}\hat{A}^T$ and $\hat{A}^T\hat{A}$ have the same non-zero eigenvalues we already get with the probability that Lemma 27 holds that $\hat{B}^2$ has $2k$ non-zero eigenvalues, i.e., rank at least $2k$. Note that $\hat{B}$ and $\hat{B}^2$ have the same rank. Furthermore $\hat{B}$ also has rank at most $2k$ since among the rows of $\hat{A}$ there are at most k different ones, i.e., the rank of $\hat{A}$ and the rank of $\hat{A}^T$ are at most k which implies that $\hat{B}$ has rank at most $2k$. In combination we get that $\hat{B}$ has rank $2k$ with probability at least $1 - e^{-cm}$. □

Observe now that if λ or $-\lambda$ is a non-zero eigenvalue of $\hat{B}$, then λ^2 is an eigenvalue of $\hat{B}^2$. Hence the absolute value of any of the $2k$ non-zero eigenvalues of $\hat{B}$ is at least $\sqrt{lm(2p-1)^2\alpha 0.9/\binom{\binom{n}{2}}{l}}$ with probability at least $1 - e^{-cm}$. Now we can provide a good value for the threshold

that we use in line 1 of the algorithm SEGMENTRESPONDENTS to estimate the number of consumer types k (the constants of this theorem are not needed elsewhere). This shows that with high probability we can reconstruct the number of different consumer types correctly.

Theorem 9. *Let* L *be a randomly chosen* l*-tuple for* $l = l(p, \alpha, \epsilon)$ *sufficiently large but independent of* n*. With probability at least* $1 - 2e^{-100}$*,* B *has exactly* k *positive eigenvalues and exactly* k *negative eigenvalues whose absolute value is larger than*

$$10\sqrt{m/\binom{n}{2}}.$$

Proof. First assume that L is well-separating. To estimate the eigenvalues of B, we apply Lemma 27 and Theorem 2 together with the previously established bounds on $\|B - \hat{B}\|_2$. For $i = 1, \ldots, k$ we have

$$\begin{aligned}\lambda_i(B) &\geq \lambda_i(\hat{B}) - \lambda_{m_L+l}(B - \hat{B}) \\ &\geq \sqrt{lm(2p-1)^2\alpha 0.9/\binom{n}{2}} - 5\sqrt{m_L}\end{aligned}$$

with probability at least $1 - e^{-cm}$ for some $c > 0$. Applying Chernoff bounds to obtain an upper bound on m_L once again, we get that

$$\lambda_i(B) \geq \sqrt{\frac{m}{\binom{n}{2}}}\left(\sqrt{l(2p-1)^2\alpha 0.9} - 5\sqrt{1+\xi}\right)$$

for $i = 1, \ldots, k$ with probability at least $1 - e^{-cm}$. By the same argument, the absolute value of any of the k most negative eigenvalues of B is with probability at least $1 - e^{-cm}$ at least that value. All other eigenvalues $\lambda_i(B)$, $i = k+1, \ldots, m_L + l - k$ are 0 in $\hat{B}$, and thus once again by Theorem 2, we get

$$|\lambda_i(B)| \leq 5\sqrt{m_L} \leq 5\sqrt{(1+\xi)m/\binom{n}{2}}$$

with probability at least $1 - e^{-cm}$, $i = k+1, \ldots, m_L + l - k$. Combining everything, by choosing $l = l(p, \alpha, \epsilon)$ sufficiently large and ξ sufficiently small we get that with probability at least $1 - e^{-cm}$ that there are exactly

k positive eigenvalues and k negative eigenvalues whose absolute value is larger than

$$10\sqrt{m/\binom{\binom{n}{2}}{l}}.$$

Finally, the probability that the chosen l-tuple is not well-separating is by Lemma 24 at most e^{-100}. The statement of the corollary now follows from a simple union bound, since $1 - 2e^{-100} < 1 - e^{-100} - e^{-cm}$ for $c > 0$ and m large enough. □

In the following let k denote the number of eigenvalues of B which are larger than $10\sqrt{m/\binom{\binom{n}{2}}{l}}$. We will show next that we do not just know bounds for the eigenvalues of B that hold with high probability, but we can also say something about the structure of the projectors onto the corresponding eigenspaces. Later we will use these projectors to cluster the respondents. Denote in the following by $P_A^{(k)}$ the projector onto the space spanned by the eigenvectors corresponding to the k largest eigenvalues of the symmetric matrix A.

Observation 3.

$$P_{\hat{B}}^{(k)} + \left(I - P_{\hat{B}}^{(m_L+l-k)}\right) = P_{\hat{B}^2}^{(2k)}.$$

Proof. Choose k orthonormal eigenvectors of $P_{\hat{B}}^{(k)}$ and k orthonormal eigenvectors of $\left(I - P_{\hat{B}}^{(m_L+l-k)}\right)$. These vectors are also eigenvectors of $\hat{B}^2$, corresponding to the $2k$ largest squared eigenvalues of $\hat{B}$. □

Lemma 28. *Suppose that $l = l(p, \alpha, \epsilon)$ is a sufficiently large constant such that $|\lambda_k(\hat{B})| \geq (37000/\alpha^2)\sqrt{m/\binom{\binom{n}{2}}{l}}$. Then we have with probability at least $1 - 2e^{-100}$*

$$\left\| P_B^{(k)} - P_{\hat{B}}^{(k)} \right\|_2 < \alpha^2/3000$$

and

$$\left\| P_B^{(m_L+l-k)} - P_{\hat{B}}^{(m_L+l-k)} \right\|_2 < \alpha^2/3000.$$

Proof. We know from the proof of Theorem 9 and by Lemma 26, that with probability at least $1 - 2e^{-100}$,

$$\delta_k(\hat{B}) = \left|\lambda_k(\hat{B})\right| \geq \sqrt{m/\binom{n}{2}}\left(\sqrt{l(2p-1)^2\alpha 0.9} - 5\sqrt{1+\xi}\right)$$

and

$$\|B - \hat{B}\|_2 \leq 5\sqrt{(1+\xi)m/\binom{n}{2}} < 6\sqrt{m/\binom{n}{2}}.$$

By our assumption we have $|\lambda_k(\hat{B})| \geq (37000/\alpha^2)\sqrt{m/\binom{n}{2}}$, and thus by Theorem 3, with probability at least $1 - 2e^{-100}$,

$$\begin{aligned}\left\|P_B^{(k)} - P_{\hat{B}}^{(k)}\right\|_2 &\leq \frac{2\|B-\hat{B}\|_2}{|\lambda_k(\hat{B})| - 2\|B-\hat{B}\|_2} \\ &\leq \frac{12}{37000/\alpha^2 - 12} \\ &< \alpha^2/3000.\end{aligned}$$

For the space spanned by the k most negative eigenvalues we proceed similarly to bound the norm of

$$(I - P_B^{(m_L+l-k)}) - (I - P_{\hat{B}}^{(m_L+l-k)}) = P_{\hat{B}}^{(m_L+l-k)} - P_B^{(m_L+l-k)}.$$

Again by Theorem 9, with probability at least $1 - 2e^{-100}$, it holds $\delta_{m_L+l-k}(\hat{B}) = |\lambda_{m_L+l-k+1}(\hat{B})| = |\lambda_k(\hat{B})|$. Using our assumption on l (such that $|\lambda_k(\hat{B})| \geq (37000/\alpha^2)\sqrt{m/\binom{n}{l}}$ holds) we get

$$\begin{aligned}\|P_{\hat{B}}^{(m_L+l-k)} - P_B^{(m_L+l-k)}\|_2 &\leq \frac{2\|B-\hat{B}\|_2}{|\lambda_k(\hat{B})| - 2\|B-\hat{B}\|_2} \\ &\leq \frac{12}{37000/\alpha^2 - 12} \\ &< \alpha^2/3000.\end{aligned}$$

with probability at least $1 - 2e^{-100}$. □

In order to cluster the respondents we want to work with $m_L \times m_L$ matrices and identify each column of such a matrix with a respondent. We are especially interested in the $m_L \times m_L$ matrix $\hat{A}\hat{A}^T$ and its decomposition $\hat{A}\hat{A}^T = C + D + E$ as it was used in the proof of Lemma 27.

Lemma 29. *Suppose that* $\epsilon \leq (2p-1)^2\alpha^3/90000$. *Then with probability at least* $1 - 2e^{-100}$ *it holds*

$$\left\| P^{(k)}_{\hat{A}\hat{A}^T} - P^{(k)}_C \right\|_2 < 3\alpha^2/10000.$$

Proof. As in the proof of Lemma 27 (again using a slightly weaker upper bound for $\|D\|_2$ and $\|E\|_2$, since the lemma is needed in the phase *Extending the partial rankings* of the algorithm again) we have

$$\begin{aligned} \|D + E\|_2 &\leq \|D\|_2 + \|E\|_2 \\ &\leq 8\epsilon l m_L \\ &< 9\epsilon l m / \binom{\binom{n}{2}}{l} \\ &\leq (2p-1)^2 0.01^2 \alpha^3 l m / \binom{\binom{n}{2}}{l} \end{aligned}$$

with probability at least $1 - 2e^{-100}$ (combining here and in the next sentence the probability that Lemma 27 holds as in Theorem 9 with the probability that an l-tuple is well-separating). By the same lemma, with probability at least $1 - 2e^{-100}$, the k non-zero eigenvalues of C are all at least

$$0.9l(2p-1)^2\alpha \frac{m}{\binom{\binom{n}{2}}{l}}.$$

Thus, by Theorem 3, with probability at least $1 - 2e^{-100}$,

$$\begin{aligned} \left\| P^{(k)}_{\hat{A}\hat{A}^T} - P^{(k)}_C \right\|_2 &\leq \frac{2\|\hat{A}\hat{A}^T - C\|_2}{|\lambda_k(C)| - 2\|\hat{A}\hat{A}^T - C\|_2} \\ &\leq \frac{2\alpha^3 0.01^2}{0.9\alpha - 2\alpha^3 0.01^2} \\ &< 3\alpha^2/10000. \end{aligned}$$

□

Denote by P'_B the restriction of $P_B^{(k)} + (I - P_B^{(m_L+l-k)})$ onto its first m_L rows and first m_L columns, i.e., P'_B is also an $m_L \times m_L$ matrix.

Theorem 10. *With probability at least* $1 - 2e^{-100}$*, we have*

$$\|P'_B - P_C^{(k)}\|_2 < \alpha^2/1000.$$

Proof. To simplify the notation, let

$$P_{\hat{B}} = P_{\hat{B}}^{(k)} + \left(I - P_{\hat{B}}^{(m_L+l-k)}\right) \quad \text{and} \quad P_B = P_B^{(k)} + \left(I - P_B^{(m_L+l-k)}\right).$$

Note that by Observation 3 it holds that $P_{\hat{B}} = P_{\hat{B}^2}^{(2k)}$. Since by Corollary 5 $\hat{B}^2$ has rank $2k$ with probability at least $1 - e^{-cm}$ the latter projector inherits the block structure from $\hat{B}^2$ with that probability. Consider the left upper $m_L \times m_L$ block matrix in the projector of $P_{\hat{B}^2}^{(2k)}$ and denote it by $P'_{\hat{B}}$. By construction we have that $P'_{\hat{B}} = P_{\hat{A}\hat{A}^T}^{(k)}$. We now claim that

$$\left\| P'_B - P_{\hat{A}\hat{A}^T}^{(k)} \right\|_2 \leq \|P_B - P_{\hat{B}}\|_2 .$$

Suppose for contradiction that this claim is not true. Choose a unit vector v maximizing $\left\| P'_B - P_{\hat{A}\hat{A}^T}^{(k)} \right\|_2$ and fill it up with additional 0 entries in the last l columns to make it an $m_L + l$-dimensional vector. Now, the first m_L coordinates of $(P_B - P_{\hat{B}})v$ are by construction equal to those of $(P'_B - P_{\hat{A}\hat{A}^T}^{(k)})v$ and the following l coordinates contribute a non-negative value to the 2-norm of the resulting vector. Hence, $\|P_B - P_{\hat{B}}\|_2$ is at least as large as $\|P'_B - P_{\hat{A}\hat{A}^T}^{(k)}\|_2$.

Putting everything together and using Lemmas 28 and 29, we get that

with probability at least $1 - 2e^{-100}$,

$$\begin{aligned}
\left\|P'_B - P_C^{(k)}\right\|_2 &= \left\|P'_B - P^{(k)}_{\hat{A}\hat{A}^T} + P^{(k)}_{\hat{A}\hat{A}^T} - P_C^{(k)}\right\|_2 \\
&\leq \left\|P'_B - P^{(k)}_{\hat{A}\hat{A}^T}\right\|_2 + \left\|P^{(k)}_{\hat{A}\hat{A}^T} - P_C^{(k)}\right\|_2 \\
&\leq \|P_B - P_{\hat{B}}\|_2 + \left\|P^{(k)}_{\hat{A}\hat{A}^T} - P_C^{(k)}\right\|_2 \\
&= \left\|P_B^{(k)} + (I - P_B^{(m_L+l-k)}) - P_{\hat{B}}^{(k)} - (I - P_{\hat{B}}^{(m_L+l-k)})\right\|_2 + \\
&\quad \left\|P^{(k)}_{\hat{A}\hat{A}^T} - P_C^{(k)}\right\|_2 \\
&\leq \left\|P_B^{(k)} - P_{\hat{B}}^{(k)}\right\|_2 + \left\|(I - P_B^{(m_L+l-k)}) - (I - P_{\hat{B}}^{(m_L+l-k)})\right\|_2 + \\
&\quad \left\|P^{(k)}_{\hat{A}\hat{A}^T} - P_C^{(k)}\right\|_2 \\
&< \alpha^2/3000 + \alpha^2/3000 + 3\alpha^2/10000 \\
&< \alpha^2/1000.
\end{aligned}$$

□

Later on it will be more convenient to work with the Frobenius norm $\|\cdot\|_F$ instead of the L_2 matrix norm. We get the following for the Frobenius norm.

Corollary 6. *We have with probability at least* $1 - 2e^{-100}$

$$\left\|P'_B - P_C^{(k)}\right\|_F^2 < \alpha^2/210000.$$

Proof. Since the rank of $P'_B + P_C^{(k)}$ is at most $3k$ and since $k \leq 1/\alpha$, we have by Theorem 10 with probability at least $1 - 2e^{-100}$ that

$$\left\|P'_B - P_C^{(k)}\right\|_F^2 \leq 3k \left\|P'_B - P_C^{(k)}\right\|_2^2 < \frac{3\alpha^3}{1000^2} < \alpha^2/210000.$$

□

Corollary 6 will be useful for the analysis of our algorithm, since we can characterize $P_C^{(k)}$ explicitly. By the block diagonal structure of the

matrix C we have $(P_C)^{(k)}_{rs} = 1/m^i_L$, if both respondents r and s belong to type i (and compare all the product pairs in the l-tuple L), and $(P_C)^{(k)}_{rs} = 0$ otherwise.

Theorem 11. *With probability at least $1 - 3e^{-100}$, for a randomly chosen l-tuple L, for every consumer type i, the algorithm* SEGMENTRESPONDENTS *misclassifies at most 3% of the m^i_L respondents.*

Proof. We first assume that the event described in Corollary 6 holds and we will consider the corresponding probability only at the end of each the two cases of the proof. We want to use the bound in Corollary 6 on the squared Frobenius norm to bound the number of entries in P'_B that become large though the corresponding entry in $P^{(k)}_C$ is zero. A column in the projector $P^{(k)}_C$ whose column index corresponds to a respondent of type i has $m_L - m^i_L$ zero entries and

$$m^i_L \geq (1-\xi)\alpha m / \binom{\binom{n}{2}}{l} > 0.99\alpha m / \binom{\binom{n}{2}}{l}$$

(with probability at least $1 - e^{-cm}$ for some $c > 0$) entries of value $1/m^i_L$. Let x_r be the number of entries in the r'th column of P'_B that are either at least $0.49\binom{\binom{n}{2}}{l}/m$ although the corresponding entry in $P^{(k)}_C$ is zero or that are less than $0.49\binom{\binom{n}{2}}{l}/m$ although the corresponding entry in $P^{(k)}_C$ is $1/m^i_L \geq 0.99\binom{\binom{n}{2}}{l}/m$. Let $x = \sum_{r=1}^{m_L} x_r$ be the total number of such entries. In any case, the contribution of an entry which is counted in x_r to the squared Frobenius norm of $P'_B - P^{(k)}_C$ is at least $\left(0.49\binom{\binom{n}{2}}{l}/m\right)^2$. Using the bound $\|P'_B - P^{(k)}_C\|^2_F < \alpha^2/210000$ from Corollary 6, we can bound x from above as

$$x\left(0.49\binom{\binom{n}{2}}{l}/m\right)^2 < \alpha^2/210000$$

or equivalently

$$x < \frac{m^2\alpha^2}{50421\binom{\binom{n}{2}}{l}^2} < \frac{m^2\alpha^2}{50000\binom{\binom{n}{2}}{l}^2}.$$

That is, there are at most

$$\frac{\alpha^2 m^2}{50000\binom{\binom{n}{2}}{l}^2}$$

entries which are either zero in $P_C^{(k)}$ and at least $\frac{0.49}{m}\binom{n}{2}{l}$ in P'_B or at least $\frac{0.99}{m}\binom{n}{2}{l}$ in $P_C^{(k)}$ and less than $\frac{0.49}{m}\binom{n}{2}{l}$ in P'_B. Hence, in at most $0.01\alpha m/\binom{n}{2}{l}$ of the columns of P'_B at least one of the following events can happen:

(1) $0.01\alpha m/\binom{n}{2}{l}$ entries being zero in $P_C^{(k)}$ become at least $\frac{0.49}{m}\binom{n}{2}{l}$ in P'_B.

(2) $0.01\alpha m/\binom{n}{2}{l}$ entries that are at least $\frac{0.99}{m}\binom{n}{2}{l}$ in $P_C^{(k)}$ become less than $\frac{0.49}{m}\binom{n}{2}{l}$ in P'_B.

Thus observe that only for these columns the condition

$$|c_r|^2 \geq 0.99\alpha m/\binom{\binom{n}{2}}{l}$$

in line 8 of the algorithm might not be satisfied – the constants 0.01 as well as the constants in some of the inequalities above were chosen to be not strict, so that we do not have to change the constants anymore to account for the fact that m_L might be smaller or larger than $m/\binom{n}{2}{l}$ and that the size of the smallest consumer type might be slightly smaller than $\alpha m/\binom{n}{2}{l}$ (we will only in the end include the probabilities of these events). Call the elements $i \in \{1, \ldots, m_L\}$ corresponding to the columns where one of the two events mentioned above occurs to be *bad*. Note that a *bad* element i might also satisfy the condition in line 8 of the algorithm. To prove the correctness of the algorithm we now make a case analysis depending on the fact whether the element i chosen in line 14 of the algorithm is *bad* or not.

Case that i is not *bad*: if i is not *bad*, then for all other elements j corresponding to the same type r which are also not *bad* we have

$$\langle c_i, c_j \rangle \geq (1 - 0.02)m_L^r = 0.98m_L^r.$$

Since both i and j are not *bad*, we have that both $|c_i|$ and $|c_j|$ are smaller than $\sqrt{(1 + 0.01)m_L^r}$. Hence,

$$\frac{\langle c_i, c_j \rangle}{|c_i||c_j|} \geq \frac{0.98}{1.01} > 0.97.$$

Thus all these elements are put together into C and also into C′ in lines 15 and 16, respectively, of the algorithm. On the other hand, for all j of a different type s which are not *bad* we have

$$\langle c_i, c_j \rangle \leq 0.02\alpha m / \binom{\binom{n}{2}}{l}.$$

Since both i and j are not *bad*, we have that

$$|c_i| \geq \sqrt{(1-0.01)m_L^r} \quad \text{and} \quad |c_j| \geq \sqrt{(1-0.01)m_L^s}.$$

Hence,

$$\frac{\langle c_i, c_j \rangle}{|c_i||c_j|} \leq \frac{0.02\alpha m}{0.99\sqrt{m_L^r m_L^s}\binom{\binom{n}{2}}{l}} \leq \frac{0.02}{0.99} < 0.03,$$

since $m_L^r, m_L^s \geq \alpha m / \binom{\binom{n}{2}}{l}$. Thus, an element corresponding to a different type that is not *bad* is neither put into C nor C′. Since there are at most $0.01\alpha m / \binom{\binom{n}{2}}{l}$ many *bad* elements which might be put into C′ but not into C we have

$$|C'| \leq |C| + 0.02\alpha m / \binom{\binom{n}{2}}{l}.$$

The fact that there are only $0.01\alpha m / \binom{\binom{n}{2}}{l}$ many *bad* elements also implies that we have $|C| \geq 0.9\alpha m / \binom{\binom{n}{2}}{l}$. Since everything was conditioned under the event that Corollary 6 holds and all Chernoff bounds in this case hold with probability at least $1 - e^{-cm}$ for some $c > 0$, by taking a union bound, we have shown that in this case at least a 99%-fraction of the respondents of type r is segmented correctly with probability at least $1 - 2e^{-100} - e^{-cm} > 1 - 3e^{-100}$.

Case that i is *bad*: if i is *bad* but still satisfies the condition in line 8 of the algorithm, a new type is reconstructed in line 19 of the algorithm only if

$$|C| \geq 0.9\alpha m / \binom{\binom{n}{2}}{l} \text{ and } |C'| \leq |C| + 0.02\alpha m / \binom{\binom{n}{2}}{l}.$$

We first show that among all elements, which are not *bad*, only those that correspond to respondents of one type can be put into C. Let j be

an element that is not *bad*, whose corresponding respondent is of type r and that is put into C. Then we must have

$$\frac{\langle c_i, c_j \rangle}{|c_i||c_j|} \geq 0.97.$$

Now, if k is an element that is not *bad* and whose corresponding respondent is of type s with $s \neq r$, then we have by the same argument as in the first case that

$$\frac{\langle c_j, c_k \rangle}{|c_j||c_k|} \leq 0.03.$$

Hence we get for the angle $\angle c_i c_j$ between c_i and c_j that $\angle c_i c_j < \pi/10$ and for the angle $\angle c_j c_k$ between c_j and c_k that $\angle c_j c_k > 9\pi/20$. Note that since all $c_i, c_{i'}$ are vectors in $\{0,1\}^{m_L}$, we have $0 \leq \frac{\langle c_i, c_{i'} \rangle}{|c_i||c_{i'}|} \leq 1$ and hence we only have to consider angles $\in [0, \pi/2)$. Thus, by the triangle inequality for angles,

$$\angle c_i c_k \geq \angle c_j c_k - \angle c_i c_j > \frac{7\pi}{20} > \frac{\pi}{3},$$

and hence

$$\cos(\angle c_i c_k) = \frac{\langle c_i, c_k \rangle}{|c_i||c_k|} < 1/2.$$

Thus c_k does not satisfy the condition in line 15 of the algorithm and the element k will not be put into C.

Next we show that if some elements that all correspond to the same type r and that are all not *bad*, are put into C then either a 97%-fraction of all not *bad* elements corresponding to the respondents of type r is put into C or none of them. To prove this, we make use of the set C'. If there was only the condition that $|C| \geq 0.9\alpha m/\binom{n}{2}$ it could happen that the respondents of type r would be split, because a large fraction of the characteristic vectors c_j corresponding to the respondents would still satisfy the condition $\frac{\langle c_i, c_j \rangle}{|c_i||c_j|} \geq 0.97$ and the rest would not satisfy it anymore. For example, this value could be just around this threshold for all characteristic vectors corresponding to respondents of one type. We now show that if we have a second threshold test $\frac{\langle c_i, c_j \rangle}{|c_i||c_j|} \geq 0.8$, and

if we require

$$|C'| \leq |C| + 0.02\alpha m/\binom{\binom{n}{2}}{l} \text{ and } |C| \geq 0.9\alpha m/\binom{\binom{n}{2}}{l},$$

then at least a 97%-fraction of the respondents of type r which is not *bad* is segmented correctly. To see this, let j be an element that is not bad which corresponds to a respondent of type r and is put into C and let j' be another element corresponding to respondent of the same type r, which is not *bad* either. Since j is put into C we must have

$$\frac{\langle c_i, c_j \rangle}{|c_i||c_j|} > 0.97.$$

Also, as shown before,

$$\frac{\langle c_j, c_{j'} \rangle}{|c_j||c_{j'}|} > 0.97.$$

Thus, since $\angle c_i c_j < \pi/10$ and $\angle c_j c_{j'} < \pi/10$, again by the triangle inequality for angles, $\angle c_i c_{j'} < \pi/5$ and hence

$$\cos \angle c_i c_{j'} = \frac{\langle c_i, c_{j'} \rangle}{|c_i||c_{j'}|} > 0.8.$$

Thus, if there exists some element j corresponding to a respondent of type r, which is not *bad* and satisfies the condition $\frac{\langle c_i, c_j \rangle}{|c_i||c_j|} \geq 0.97$, then all other elements j' that correspond to respondents of type r and are not *bad* either satisfy at least the condition $\frac{\langle c_i, c_j \rangle}{|c_i||c_j|} \geq 0.8$. Now, if there are more than $0.02\alpha m/\binom{\binom{n}{2}}{l}$ many elements j', which satisfy only the second condition, then $|C'| > |C| + 0.02\alpha m/\binom{\binom{n}{2}}{l}$ and no type is reconstructed in line 19 of the algorithm in this iteration of the **while**-loop. Otherwise, all but at most $0.02\alpha m/\binom{\binom{n}{2}}{l}$ elements which are not *bad* satisfy both conditions, and since there are only $0.01\alpha m/\binom{\binom{n}{2}}{l}$ many *bad* elements corresponding to each type, at least a 0.97-fraction of the respondents of any type is segmented correctly. Since there are in total only $0.01\alpha m/\binom{\binom{n}{2}}{l}$ many *bad* elements, in some iteration an element which is not *bad* is chosen and a 97%-fraction of the respondents of any type will be segmented correctly at some point. Since the probability that Corollary 6 fails is at most $2e^{-100}$, and the probability that other

undesired events happen throughout the proof is at most e^{-cm} for some $c > 0$, by a union bound, with probability at least

$$1 - 2e^{-100} - e^{-cm} > 1 - 3e^{-100}$$

the theorem holds. □

The theorem tells us that with high probability for all consumer types at most a 3%-fraction of the respondents gets misclassified. Conditioned under this event, we can now easily, by majority vote, extract the rankings of the k consumer types: for each reconstructed consumer type and for each (ordered) product pair $(x, y) \in L$ we say that the consumer type prefers x over y if more than half of the respondents of this type have stated that they prefer x over y, otherwise we say the consumer type prefers y over x.

Lemma 30. *With probability at least $1-4e^{-100}$, for every consumer type i and its typical ranking u_i, $\pi_L(u_i)$ is reconstructed perfectly.*

Proof. In the following we condition everything under the event that for each consumer type at most 3% of its respondents get classified wrongly by the algorithm SEGMENTRESPONDENTS, by Theorem 11 this happens with probability at least $1 - 3e^{-100}$. For an arbitrary element $j \in L$ and an arbitrary consumer type i suppose without loss of generality that consumer type i prefers in the product comparison j the first element over the second with probability $p > 1/2$. Consider the (unique) reconstructed type $\hat{i}$ with corresponding size $m_L^{\hat{i}}$ whose elements belong to at least 97% to type i. For each respondent v belonging to $\hat{i}$, define the indicator variable u_v^j to be 1 if v prefers in product comparison j the first element over the second and 0 otherwise. Set $u_j := \sum_{v \in \hat{i}} u_v^j$. Then

$$E[u_j] \geq 0.97pm_L^{\hat{i}} + 0.03(1-p)m_L^{\hat{i}} = m_L^{\hat{i}}(0.94p + 0.03) > m_L^{\hat{i}}(0.5 + 0.9\gamma),$$

where the last inequality follows since $p > 1/2$, and we can thus find an arbitrarily small constant $\gamma > 0$ such that $p = 1/2 + \gamma$. Since all respondents answer their product comparisons independently, by Lemma 25 and once again by Chernoff bounds, for some small $\xi = \xi(\gamma) > 0$ such that $(1 - \xi)(0.5 + 0.9\gamma) > 1/2$, we get

$$\begin{aligned} \Pr\left[(1-\xi)(0.5+0.9\gamma)m_L^{\hat{i}} \leq u_j\right] &\geq \Pr\left[(1-\xi)E[u_j] \leq u_j\right] \\ &\geq 1 - e^{-\xi^2 E[u_j]/4} \\ &\geq 1 - e^{-\xi c' m}, \end{aligned}$$

for some $c' = c'(n, l, \alpha, \epsilon) > 0$. This also implies that, with probability at least $1 - e^{-\xi c' m}$, the number of respondents belonging to $m_L^{\hat{i}}$ preferring in product comparison j the second element over the first is at most

$$m_L^{\hat{i}} (1 - (1 - \xi)(0.5 + 0.9\gamma)) m_L^{\hat{i}} = (0.5 - \gamma') m_L^{\hat{i}}$$

for some constant $\gamma' = \gamma'(\gamma, \xi) > 0$. Thus, with probability at least $1 - e^{-\xi c' m}$, the number of respondents preferring the first element over the second in product comparison j minus the number of respondents preferring the second element over the first in this product comparison is at least

$$m_L^{\hat{i}} ((1 - \xi)(0.5 + 0.9\gamma) - (0.5 - \gamma')) = \gamma_0 m_L^{\hat{i}} = \gamma_0 c m$$

for some $\gamma_0 = \gamma_0(\gamma, \xi, \gamma') > 0$ and some $c > 0$. Thus, $\gamma_0 c m > 0$ and the majority vote will give the right ranking with probability at least $1 - e^{-\xi c' m}$. Analogous calculations give the same bounds if consumer type i prefers in the product comparison j the second element over the first with probability p. Taking a union bound over all l elements and all k consumer types and taking into account the probability that Theorem 11 holds, we get that with probability at least

$$1 - 3e^{-100} - lke^{-cm} > 1 - 4e^{-100},$$

for any consumer type i with typical ranking u_i, $\pi_L(u_i)$ is reconstructed perfectly. □

Next we use that if L is well-separating and we exchange one pair in L with a pair from $X \setminus \bigcup_{Y \in L} Y$ to get an l-tuple L', then also L' is "almost" well-separating and basically everything we proved for L remains valid for L'. That is, with high probability for both L and L', respectively, exactly the same number k of typical consumer types will be computed whose rankings all agree on all the $l - 1$ product comparisons in $L \cap L'$. Thus the segments computed for L and L' can be easily merged.

Theorem 12. *Suppose that $l = l(\alpha, p, \epsilon)$ is a sufficiently large constant (but independent of n). Then, with probability at least $1 - 5e^{-100}$, all consumer type rankings can be reconstructed perfectly.*

Proof. By Theorem 11 and Lemma 30 we know that with probability at least $1 - 4e^{-100}$ a randomly chosen l-tuple L is well-separating, at

most a 3%-fraction of each consumer type answering exactly the l product comparisons in L is misclassified and for any consumer type i with corresponding consumer type ranking u_i, $\pi_L(u_i)$ is reconstructed perfectly. Observe that since the algorithm at any step replaces only one element in L by another element to obtain L', L' will still be "almost" well-separating. Indeed, if an l-tuple L is well-separating, then recall from Lemma 27 (with the notation of D and E borrowed from there) that for this l-tuple we get $|D_{ij}| \leq 3\epsilon l$ for any two different consumer types i and j (the same holds for E). Since L and L' differ in only one element, for the same matrix D applied to L' we have $|D_{ij}| \leq 3\epsilon l + 2$ for any two different consumer types i and j, since the absolute value can change by at most 2. A short glance at the proof of Lemma 27, however, shows that in this proof we require only that $|D_{ij}| \leq 4\epsilon l$ (and also $|E_{ij}| \leq 4\epsilon l$), and by choosing $l = l(\alpha, p, \epsilon)$ bigger than $2/\epsilon$, we can guarantee that Lemma 27 also holds when applied with L'. The proofs of Lemma 29, Theorem 11 and Lemma 30 then also hold with probability at least $1 - e^{-cm}$ for some $c > 0$ (conditioned under the fact that L was well-separating). This implies that with probability at least $1 - e^{-cm}$ for any L' such that $|L' \setminus L| = |L \setminus L'| = 1$ and any consumer type i with typical ranking u_i, $i = 1, \ldots, k$, the $l - 1$ common product comparisons of $\pi_L(u_i)$ and $\pi_{L'}(u_i)$ agree perfectly, and since they were well-separated in L, they can easily be merged (in the obvious way). Taking a union bound over all k consumer types and all at most $\binom{n}{2}$ iterations of the algorithm, with probability $1 - k\binom{n}{2}e^{-cm} = 1 - e^{-c'm}$ for some $c, c' > 0$ all typical consumer type rankings get reconstructed perfectly. Combining the probabilities, we see that with probability at least

$$1 - 4e^{-100} - e^{-c'm} > 1 - 5e^{-100}$$

all consumer type rankings get reconstructed perfectly. □

Finally, we show that also most of the respondents get classified correctly.

Theorem 13. *Suppose that $l = l(p, \alpha, \epsilon)$ is a sufficiently large constant (but independent of n). Then with probability at least $1 - 6e^{-100}$, for each consumer type i, $i = 1, \ldots, k$, at most a e^{-97}-fraction of the respondents of that type is misclassified.*

Proof. We condition our analysis under the event that all consumer type rankings are reconstructed perfectly, that is that Theorem 12 holds, and

include the probability of this event only at the end. For each not yet reconstructed respondent r of consumer type i answering a well-separating l-tuple L of product comparisons, define $\delta(r, u_i)_L$ to be the Hamming distance between the $\{\pm 1\}^l$ vector of answers provided by r and the projection of the typical ranking u_i of consumer type i onto L, i.e., the number of inverted pairs on L. We have

$$E[\delta(r, u_i)_L] = (1-p)l,$$

and since all l product comparisons are performed independently, by Chernoff bounds, for $\xi > 0$, where ξ is chosen small enough such that $(1+\xi)E[\delta(r, u_i)_L] < (1-\xi)\left(\frac{l-3\epsilon l}{2}p + (1-p)\frac{l+3\epsilon l}{2}\right)$ holds, we get

$$\Pr\left[\delta(r, u_i)_L \geq (1+\xi)E[\delta(r, u_i)_L]\right] \leq e^{-\xi^2(1-p)l/4}.$$

For any other typical consumer type ranking u_j, $j \neq i$, denote by $\delta(i,j)_L$ the Hamming distance between $\pi_L(u_i)$ and $\pi_L(u_j)$. Since L is well-separating, $\delta(i,j)_L \geq \frac{l-3\epsilon l}{2}$, and since $p > 1/2$ we get

$$E[\delta(r, u_j)_L] = p\delta(i,j)_L + (1-p)(l - \delta(i,j)_L) \geq p\frac{l-3\epsilon l}{2} + (1-p)\frac{l+3\epsilon l}{2}.$$

Also, by Chernoff bounds, for this $\xi > 0$,

$$\begin{aligned}
&\Pr\left[\delta(r, u_j)_L \leq (1-\xi)\left(p\frac{l-3\epsilon l}{2} + (1-p)\frac{l+3\epsilon l}{2}\right)\right] \\
&\quad\leq \Pr\left[\delta(r, u_j)_L \leq (1-\xi)E[\delta(r, u_j)_L]\right] \\
&\quad\leq e^{-\xi^2 E[\delta(r,u_j)_L]/4} \\
&\quad\leq e^{-\xi^2\left(\frac{l-3\epsilon l}{2}p + (1-p)\frac{l+3\epsilon l}{2}\right)/4}.
\end{aligned}$$

Thus, by taking a union bound over all consumer types $j \neq i$, by our choice of ξ, for sufficiently large l, with probability at least

$$1 - e^{-\xi^2(1-p)l/4} - (k-1)e^{-\xi^2\left(\frac{l-3\epsilon l}{2}p + (1-p)\frac{l+3\epsilon l}{2}\right)/4} > 1 - e^{-100}$$

a randomly chosen respondent answering l product comparisons in L is correctly classified. Thus, for each consumer type i, the expected number of correctly classified respondents who do the comparisons corresponding to L is at least

$$(1-e^{-100})|m_L^i| \geq (1-e^{-100})(1-\xi')m^i \Big/ \binom{\binom{n}{2}}{l}$$

by Lemma 25 with probability at least $1 - e^{-cm}$. Since at least a $(1 - e^{-99})$-fraction of l-tuples is well-separating and not yet used in one of the at most $\binom{n}{2}$ previous iterations of the algorithm SEGMENTRESPONDENTS, the expected number of correctly classified respondents of type i is at least

$$(1 - e^{-99})(1 - e^{-100})(1 - \xi')m^i > (1 - e^{-98})m^i$$

with probability at least $1 - e^{-cm}$ (again by Lemma 25). Furthermore, since for those respondents who were not used in the $\binom{n}{2}$ previous runs of the algorithm, the classification of each respondent is independent from any other respondent, by another application of Chernoff bounds together with a union bound over all $\Theta(m)$ respondents of each consumer type and all k consumer types, with probability at least $1 - e^{-cm}$ for some $c = c(\xi) > 0$ at least

$$(1 - \xi'')(1 - e^{-98})m^i > (1 - e^{-97})m^i$$

respondents of each consumer type get correctly classified. Since everything was conditioned under the fact that the typical consumer type rankings get reconstructed perfectly, by combining the probabilities, we get that with probability at least

$$1 - 5e^{-100} - e^{-cm} > 1 - 6e^{-100}$$

at most a e^{-97}-fraction of each consumer type gets misclassified. □

4.5 Concluding remarks

We have studied the problem to elicit product preferences of a population, where the preferences are represented as a ranking of the products. During elicitation we asked many respondents to perform a few pairwise comparisons. We provided an algorithm to process the elicited data and introduced models of population and respondents and analyzed our algorithm in their context. The following theorem summarizes the analysis of our collaborative ranking algorithm.

Theorem 14. *For any $\xi > 0$ and any $\gamma > 0$, we can choose $l = l(p, \alpha, \epsilon)$ to be a sufficiently large constant (but independent of n) such that for our model of population and respondents and a sufficiently large number of respondents we can with probability $1 - \xi$, correctly*

(1) infer the number k *of different consumer types, and*

(2) segment a $(1-\gamma)$*-fraction of the respondents into correct types, and*

(3) reconstruct the k *typical consumer rankings.*

Note that in the actual proof, i.e., in the proofs of Theorems 9, 12 and 13, we chose the arbitrary values $\xi = 1 - 6e^{-100}$ and $\gamma = e^{-97}$ to make the notation of the statements not overly complicated. The constants are arbitrary, but in our model with high probability there will always be a small but constant fraction of respondents of each type which gets misclassified, and hence we cannot get $\gamma = o(1)$. However, it seems possible, that with an additional amount of work the probability $1-\epsilon$ could be replaced by something that holds with probability $1-o(1)$.

Eliciting typical rankings and segmenting a population by asking each respondent only a few questions seems appealing for web based marketing research though there remain many issues to deal with, e.g., spam in the form of malicious respondents or measuring more detailed information on an interval scale.

Bibliography

[AFKM01] D. Achlioptas, A. Fiat, A. R. Karlin, and F. McSherry. Web search via hub synthesis. *Proc. 42nd IEEE Symposium on Foundations of Computer Science*, pages 500–509, 2001.

[AFKS01] Y. Azar, A. Fiat, A. R. Karlin, and J. Saia. Spectral analysis of data. *Proc. 33th Symposium on Theory of Computing*, pages 619–626, 2001.

[AKS98] N. Alon, M. Krivelevich, and B. Sudakov. Finding a large hidden clique in a random graph. *Random Structures and Algorithms*, 13:457–466, 1998.

[AKV02] N. Alon, M. Krivelevich, and V. H. Vu. On the concentration of eigenvalues of random symmetric matrices. *Israel Journal of Mathematics*, 131:259–267, 2002.

[AS00] N. Alon and J. Spencer. *The Probabilistic Method*. Wiley, 2nd Ed., 2000.

[Bol01] B. Bollobás. *Random Graphs*. Cambridge University Press, New York, 2nd Ed., 2001.

[Bop87] R. B. Boppana. Eigenvalues and graph bisection: An average-case analysis. *Proceedings of 28th IEEE Symposium on Foundations on Computer Science*, pages 280–285, 1987.

[BS04] B. Bollobás and A.D. Scott. Max cut for random graphs with a planted partition. *Combinatorics, Probability and Computing*, 13:451–474, 2004.

[CDG+99] S. Chakrabarti, B. Dom, D. Gibson, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Topic distillation and spectral filtering. *Artif. Intell. Rev.*, 13 (5-6):409–435, 1999.

[CK01] A. Condon and R. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

[CO06] A. Coja-Oghlan. A spectral heuristic for bisecting random graphs. *Random Structures and Algorithms*, 29:351–398, 2006.

[DDL+90] S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41 (6):391–407, 1990.

[Dey04] T.K. Dey. Curve and surface reconstruction. *Handbook of Discrete and Computational Geometry*, 2004.

[DKF+04] P. Drineas, R. Kannan, A. Frieze, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56:9–33, 2004.

[DKR02] P. Drineas, I. Kerenidis, and P. Raghavan. Competitive recommendation systems. *Proceedings of 32nd IEEE Symposium on Theory of Computing*, pages 82–90, 2002.

[FK81] Z. Füredi and J. Komlós. The eigenvalues of random symmetric matrices. *Combinatorica I*, 3:233–241, 1981.

[GJS76] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified *NP*-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[GM05a] J. Giesen and D. Mitsche. Boosting spectral partitioning by sampling and iteration. *Proc. 16th Annual International Symposium on Algorithms and Computation (ISAAC)*, pages 473–482, 2005.

[GM05b] J. Giesen and D. Mitsche. Bounding the misclassification error in spectral partitioning in the planted partition model. *Proc. 31st International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 409–420, 2005.

[GM05c] J. Giesen and D. Mitsche. Reconstructing many partitions using spectral techniques. *15th International Symposium on Fundamentals of Computation Theory (FCT)*, pages 433–444, 2005.

[GMS07] J. Giesen, D. Mitsche, and E. Schuberth. Collaborative ranking: An aggregation algorithm for individuals' preference estimation. *Proceedings of 3rd International Conference on Algorithmic Aspects in Information and Management*, 2007.

[GNOT92] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35:51–60, 1992.

[Gut05] M. H. Gutknecht. *L*ineare *A*lgebra. *Lecture Notes ETH Zurich*, 2005.

[Kle99] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46 (5):604–632, 1999.

[KRRT98] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Recommendation systems: A probabilistic analysis. *Proc. 39th IEEE Symposium on Foundations of Computer Science*, pages 664–673, 1998.

[KS03] J. Kleinberg and M. Sandler. Convergent algorithms for collaborative filtering. *Proc. ACM Conference on Electronic Commerce*, 2003.

[KS04] J. Kleinberg and M. Sandler. Using mixture models for collaborative filtering. *Proc. 36th Symp. Theory of Computing*, pages 569–578, 2004.

[McS01] F. McSherry. Spectral partitioning of random graphs. *Proceedings of 42nd IEEE Symposium on Foundations of Computer Science*, pages 529–537, 2001.

[MV] M. Meila and D. Verma. A comparison of spectral clustering algorithms. *UW CSE Technical report 03-05-01*.

[PB98] L. Page and S. Brin. Pagerank, an eigenvector based ranking approach for hypertext. *21st Annual ACM/SIGIR International Conference on Research and Development in Information Retrieval*, 1998.

[PRTV98] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. *Symposium on Principles of Database Systems*, pages 159–168, 1998.

[RV97] P. Resnick and H.R. Varian. CACM special issue on recommender systems. *Communications of the ACM*, 40(3), 1997.

[SS90] G. Stewart and J. Sun. *Matrix perturbation theory*. Academic Press, Boston, 1990.

[ST96] D. Spielman and S.-H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Proceedings of 37th IEEE Symposium on Foundations on Computer Science*, pages 96–105, 1996.

[ST02] R. Shamir and D. Tsur. Improved algorithms for the random cluster graph model. *Proceedings 7th Scandinavian Workshop on Algorithm Theory*, pages 230–259, 2002.

[Str88] G. Strang. *Linear Algebra and its Applications*. Harcourt Brace, San Diego, 3rd Ed., 1988.

[Vu05] V. H. Vu. Spectral norm of random matrices. *Proceedings of 37th ACM Symposium on Theory of Computing*, pages 423–430, 2005.

[Wel02] E. Welzl. Basic examples of probabilistic analysis. *Manuscript ETH Zurich*, 2002.