Consistency of Spectral Algorithms for Hypergraphs under Planted Partition Model

A THESIS

SUBMITTED FOR THE DEGREE OF

Doctor of Philosophy

IN THE

Faculty of Engineering

BY

Debarghya Ghoshdastidar

Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

November, 2016

Declaration of Originality

I, **Debarghya Ghoshdastidar**, with SR No. **04-04-00-10-12-12-1-09711** hereby declare that the material presented in the thesis titled

Consistency of Spectral Algorithms for Hypergraphs under Planted Partition Model

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2012–2016**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the thesis.

Advisor Name: Advisor Signature

The true adventurer goes forth aimless and uncalculating to meet and greet unknown fate. A fine example was the Prodigal Son when he started back home.

O. Henry, *The Four Million*

Acknowledgements

I feel blessed to have been able to reach this stage of my journey. There are memories I will always cherish, and there are times that I would like to forget. But, I will certainly remember this trip as a remarkable adventure.

Before we start a journey, we are always advised to fill our backpacks with all the necessities. Yet, how often do we find ourselves in need for new resources to cross the hurdles that come our way. I am fortunate to have worked amongst people, who have always provided me with all the tools whenever I needed them. I am indebted to my advisor, Prof. Ambedkar Dukkipati, who has always gladly shared his knowledge and experience. I also thank all professors at the Indian Institute of Science who have taught me new lessons every day. In particular, I feel that the teachings of Prof. Shalabh Bhatnagar, Prof. Venu Madhav Govindu, Prof. Kaushal Verma, Prof. Manjunath Krishnapur, Prof. A. K. Nandakumaran, Prof. P. S. Sastry, Prof. Chandra R. Murthy, and Prof. Shivani Agarwal have and will always get reflected in my works.

It is hard to carry on unless one has the means to support himself through the journey. I sincerely thank Google for providing financial supports through the Google Ph.D. Fellowship. I also thank Mr. Ashwani Sharma at Google India, the staff at the Academic and Finance units of I.I.Sc., and the staff in the Department of Computer Science & Automation, who have all extended their help at different stages of my journey.

As we meander around the barriers on the road, we pick up the trails of travelers who passed before us. I am grateful to all the researchers worldwide, whose works have taught me, inspired me and motivated me to find answers to the numerous questions that have sprung up during my research. In particular, many thanks to the authors who have made their implementations available. Repositories for scientific data and research papers, including UCI machine learning repository and arxiv.org among others, have been a blessing for all of us. The conferences and workshops I have attended in the past few years have always provided me with new research directions, and I thank the organizers of these events.

Everyone knows that we learn from our mistakes, but few can learn to realize their mistakes. Theoretical research is a great example of this paradox. Fortunately, my works have always

been examined by experienced researchers, who have identified my mistakes and have guided me in the correct direction. I am grateful to the examiners of this thesis, who have taken the time to go through the entire thesis and pointed out all errors. I would also like to thank my advisor for his careful reading and comments in all our papers as well as this thesis. I also thank all the anonymous reviewers of our papers, and the committee of my comprehensive examination.

How often do we need to hear that applause or get the pat on the back to move on with our quest. I would like to express my sincere gratitude to Dr. Mayur Datar and Prof. Ulrike von Luxburg for the appreciation they have showered upon my work. I also thank my collaborators, group members and friends for their cheerful comments. The company of friends like Abhranil, Saswata and Suraj has been delightful.

This journey would have been impossible without my beloved sister, Debostuti, and my dearest friend and companion, Oindrila, who have walked this road along with me. They smiled in my joy, and wept for my sorrow, nursed my wounds each day, and gave me strength for the morrow. I am also privileged to have such wonderful parents and a fabulous sister, Debarati, whom I could always look up to for inspiration. My brothers-in-law, Pallab and Jacob, have always been more than brothers to me. I am also grateful to Oindrila's parents, and her sister, Anwesha, for their support.

And now it's time to move on – continue on this road. I do wonder where it is going to take me, and what am I going to find at the next bend. But, I do believe that it will be as splendid as my journey so far.

Abstract

Hypergraph partitioning lies at the heart of a number of problems in machine learning as well as other engineering disciplines. While partitioning uniform hypergraphs is often required in computer vision problems that involve multi-way similarities, non-uniform hypergraph partitioning has applications in database systems, circuit design etc. As in the case of graphs, it is known that for given objective and balance constraints, the problem of optimally partitioning a hypergraph is NP-hard. Yet, over the last two decades, several efficient heuristics have been studied in the literature and their empirical success is widely appreciated. In contrast to the extensive studies related to graph partitioning, the theoretical guarantees of hypergraph partitioning approaches have not received much attention in the literature. The purpose of this thesis is to establish the statistical error bounds for certain spectral algorithms for partitioning uniform as well as non-uniform hypergraphs.

The mathematical framework considered in this thesis is the following. Let $\mathcal{V}$ be a set of n vertices, and $\psi : \mathcal{V} \to \{1, \ldots, k\}$ be a (hidden) partition of $\mathcal{V}$ into k classes. A random hypergraph $(\mathcal{V}, \mathcal{E})$ is generated according to a planted partition model, *i.e*, subsets of $\mathcal{V}$ are independently added to the edge set $\mathcal{E}$ with probabilities depending on the class memberships of the participating vertices. Let ψ' be the partition of $\mathcal{V}$ obtained from a certain algorithm acting on a random realization of the hypergraph. We provide an upper bound on the number of disagreements between ψ and ψ'. To be precise, we show that under certain conditions, the asymptotic error is $o(n)$ with probability $(1 - o(1))$. In the existing literature, such error rates are only known in the case of graphs (Rohe et al., Ann. Statist., 2011; Lei & Rinaldo, Ann. Statist., 2015), where the planted model coincides with the popular stochastic block model. Our results are based on matrix concentration inequalities and perturbation bounds, and the derived bounds can be used to comment on the consistency of spectral hypergraph partitioning algorithms.

It is quite common in the literature to resort to a spectral approach when the quantity of interest is a matrix, for instance, the adjacency or Laplacian matrix for graph partitioning. This is certainly not true for hypergraph partitioning as the adjacency relations cannot be encoded

into a symmetric matrix as in the case of graphs. However, if one restricts the problem to m-uniform hypergraphs for some $m \geq 2$, then a symmetric tensor of order m can be used to express the multi-way interactions or adjacencies. Thus, the use of tensor spectral algorithms, based on the spectral theory of symmetric tensors, is a natural choice in this scenario. We observe that a wide variety of uniform hypergraph partitioning methods studied in the literature can be related to any one of two principle approaches: (1) solving a tensor trace maximization problem, or (2) use of the higher order singular value decomposition of tensors. We derive statistical error bounds to show that both these approaches lead to consistent partitioning algorithms.

Our results also hold when the hypergraph under consideration allows weighted edges, a situation that is commonly encountered in computer vision applications such as motion segmentation, image registration etc. In spite of the theoretical guarantees, a tensor spectral approach is not preferable in this setting due to the time and space complexity of computing the weighted adjacency tensor. Keeping this practical scenario in mind, we prove that consistency can still be achieved by incorporating certain tensor sampling strategies. In particular, we show that if the edges are sampled according to certain distribution, then consistent partitioning can be achieved with only few sampled edges. Experiments on benchmark problems demonstrate that such sampled tensor spectral algorithms are indeed useful in practice.

While vision tasks mostly involve uniform hypergraphs, in database and electronics applications, one often finds non-uniform hypergraphs with edges of varying sizes. These hypergraphs cannot be expressed in terms of adjacency matrices or tensors, and hence, use of a spectral approach is tricky in this context. The partitioning problems gets more challenging due to the fact that, in practice, these hypergraphs are quite sparse, and hence, provide less information about the partition. We consider spectral algorithms for partitioning clique and star expansions of hypergraphs, and study their consistency under a sparse planted partition model.

The results of hypergraph partitioning can be further extended to address the well-known hypergraph vertex coloring problem, where the objective is to color the vertices such that no edge is monochromatic. The hardness of this problem is well established. In fact, even when a hypergraph is bipartite or 2-colorable, it is NP-hard to find a proper 2-coloring for it. We propose a spectral coloring algorithm, and show that if the non-monochromatic subsets of vertices are independently added to the edge set with certain probabilities, then with probability $(1 - o(1))$, our algorithm succeeds in coloring bipartite hypergraphs with only two colors.

To the best our knowledge, these are the first known results related to consistency of partitioning general hypergraphs.

Publications based on this Thesis

1. Ghoshdastidar, D. and A. Dukkipati (2016). Consistency of spectral hypergraph partitioning under planted partition model. *The Annals of Statistics* (in press), arXiv:1505.01582.

2. Ghoshdastidar, D., A. P. Adsul, and A. Dukkipati (2016). Learning with Jensen-Tsallis kernels. *IEEE Transactions on Neural Networks and Learning Systems* (in press).

3. Ghoshdastidar, D. and A. Dukkipati (2016). Uniform hypergraph partitioning: Provable tensor methods and sampling techniques. *Manuscript submitted*, arXiv:1602.06516.

4. Ghoshdastidar, D. and A. Dukkipati (2015). Coloring random non-uniform bipartite hypergraphs. *Manuscript submitted*, arXiv:1507.00763.

5. Ghoshdastidar, D. and A. Dukkipati (2015). A provable generalized tensor spectral method for uniform hypergraph partitioning. In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 400–409.

6. Ghoshdastidar, D. and A. Dukkipati (2015). Spectral clustering using multilinear SVD: Analysis, approximations and applications. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2610–2616.

7. Ghoshdastidar, D. and A. Dukkipati (2014). Consistency of spectral partitioning of uniform hypergraphs under planted partition model. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 397–405.

8. Ghoshdastidar, D., A. Dukkipati, A. P. Adsul, and A. S. Vijayan (2014). Spectral clustering with Jensen-type kernels and their multi-point extensions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1472–1477.

List of notations and abbreviations

Standard quantities and functions.

Notation	Description
$\mathbb{1}\{\cdot\}$	Indicator function, returns one if argument is true, else zero.
$\ln(\cdot)$	Natural logarithm.
$\ell_{0-1}(\cdot,\cdot)$	0-1 loss function that is indicator of two labels being identical.
$\mathbb{N}$	Set of natural numbers, $\{1, 2, \ldots\}$.
$\mathbb{R}$	Set of real numbers.
$\mathbb{R}^{n_1 \times n_2 \times \ldots \times n_m}$	Space of all real tensors of order m of dimension $n_1 \times n_2 \times \ldots \times n_m$. Special cases include space of all $n_1 \times n_2$ matrices (for $m = 2$), and space of all n_1-dimensional vectors (for $m = 1$).
$\mathcal{V}_1^c$	Complement of any set $\mathcal{V}_1$.
$\lvert \cdot \rvert$	Cardinality of a set.
I	Identity matrix, dimension can be understood from context.
$\mathrm{Trace}(\cdot)$	Trace or sum of diagonal entries of a matrix or tensor
$\det(\cdot)$	Determinant of a matrix.
$\lVert \cdot \rVert_2$	Euclidean norm for vector and the spectral norm for matrix
$\lVert \cdot \rVert_F$	Frobenius norm of matrix or tensor. Square root of sum of squares of all entries.
$\mathsf{E}[\cdot]$	Expectation with respect to a specified distribution. For most of the thesis, we consider distribution of the planted model, except in Chapter 6. See Remark 6.3 for the latter case.
$\mathsf{Var}(\cdot)$	Variance with respect to a specified distribution. Above remark holds in this case also.
$\mathsf{P}(\cdot)$	Probability of an event with respect to a specified distribution, typically that of planted model (except in Chapter 6).

$O(\cdot)$	For two sequences $(a_n)_{n\in\mathbb{N}}$, $(b_n)_{n\in\mathbb{N}}$, we say $a_n = O(b_n)$ if there is a constant $C > 0$ such that $a_n \leq Cb_n$ for all n.
$\Omega(\cdot)$	$a_n = \Omega(b_n)$ if there is a constant $C > 0$ such that $a_n \geq Cb_n$ for all large n.
$\Theta(\cdot)$	$a_n = \Theta(b_n)$ if both $a_n = O(b_n)$ and $a_n = \Omega(b_n)$.
$o(\cdot)$	$a_n = o(b_n)$ if $\frac{a_n}{b_n} \to 0$ as $n \to \infty$.

Terminology specific to matrices and tensors.

Notation	Description
$\mathbf{A}$, $\mathbf{B}$ etc.	Bold faces used only for tensors
$\widetilde{\mathbf{A}}$	Flattened matrix of a tensor (only exception to bold face rule). See (2.4) for definition.
$\times_k$	Mode-k product of a tensor with a matrix. See Definition 2.4.
$\otimes$	Outer product of two or more vectors, resulting in rank-one matrices or tensors. Used in Definition 2.3.
$A_{i\cdot}$	i^{th} row of a matrix A.
$A_{\cdot i}$	i^{th} column of a matrix A.
$\lambda_{\min}(A)$	Smallest eigenvalue of A.
$\lambda_k(A)$	k^{th} largest or smallest eigenvalue of A (depends on context). In proof of Lemma 3.11, $\lambda_k(\cdot)$ refers to k^{th} largest singular value.
$\mathsf{E}[A]$ or $\mathsf{E}[\mathbf{A}]$	Entry-wise expectation of matrix or tensor.
$\mathsf{Var}(A)$	Variance of a matrix defined as $\mathsf{Var}(A) = \mathsf{E}\left[(A - \mathsf{E}[A])^2\right]$.
$\mathcal{M}_{n\times r}(k)$	Set of all matrices in $\mathbb{R}^{n\times r}$ with at most k distinct rows.
$\eta_k(A)$	If $A \in \mathbb{R}^{n\times r}$, $\eta_k(A) = \min_{S\in\mathcal{M}_{n\times r}(k)} \|A - S\|_F$. See Section 2.5.3 for use.

Graph and hypergraph terminology.

Notation	Description
$\mathcal{V}$	Set of vertices. Typically we use $i, j, i_1, i_2 s$ etc. to refer to the vertices.
$\mathcal{E}$	Set of undirected edges. Typically an edge is denoted by e.
$(\mathcal{V}, \mathcal{E})$	Unweighted undirected graph or hypergraph.
$(\mathcal{V}, \mathcal{E}, w)$	Weighted graph or hypergraph with weight function w. In this case, one may assume $\mathcal{E}$ to be collection of all subsets of $\mathcal{V}$.
$w(e)$ or w_e	Weight of an edge e in a weighted graph or hypergraph.
$\deg(i)$	Degree of a vertex $i \in \mathcal{V}$.

A	Adjacency matrix or weighted adjacency matrix of a graph. We also use the same notation to refer to weighted adjacency matrix of graph resulting from hypergraph reduction.
$\mathbf{A}$	Adjacency tensor of uniform hypergraph.
H	Incidence matrix of hypergraph.
Δ	Edge cardinality matrix, diagonal with $\Delta_{ee} = \|e\|$.
D	Degree matrix, diagonal with $D_{ii} = \deg(i)$, or $D_{ii} = \sum_j A_{ij}$ (depends on context).
L	Normalized graph (or hypergraph) Laplacian, $L = I - D^{-1/2}AD^{-1/2}$.
L_{un}	Unnormalized graph Laplacian, $L_{un} = D - A$.
$\mathrm{Vol}(\mathcal{V}_1)$	Volume of a group of vertices $\mathcal{V}_1 \subset \mathcal{V}$, which is the sum of degrees of all vertices $\mathcal{V}_1$.
$\partial\mathcal{V}_1$	Boundary of a group of vertices. Denotes all edges that have non-empty intersection with both $\mathcal{V}_1$ and $\mathcal{V}_1^c$.
$\mathrm{Cut}(\mathcal{V}_1)$	Total weight of edges in $\partial\mathcal{V}_1$. Two different definitions given in Section 2.2 and Remark 5.3.
$\mathrm{Assoc}(\mathcal{V}_1)$	Total weight of edges that that reside within $\mathcal{V}_1$. Two different definitions given in Sections 2.2 and 4.1.
R-Cut	Ratio cut of a partition of a graph. See (2.12).
R-Assoc	Ratio associativity of a partition of a graph. See (2.15).
N-Cut	Normalized cut of a partition of a graph. See (2.13).
N-Assoc	Normalized associativity of a partition of a graph. See (2.14).
NH-Cut	Normalized cut of a partition of a hypergraph. See (5.4).
NH-Assoc	Normalized associativity of a partition of a hypergraph. See (4.1).
$(\mathcal{V}, \widetilde{\mathcal{E}})$	An ideal uniform hypergraph, which is an union of completely connected components.

Quantities used in planted partition model and analysis[1].

Notation	**Description**
n	Number of vertices in planted graph or hypergraph. Exception in Chapter 7, where hypergraph has $2n$ vertices.
m	Order of uniform hypergraphs studied in Chapters 3, 4 and 6.

[1] Other quantities, not mentioned here, have also been introduced and used in different sections.

k	Denotes the size of partition planted in graph or hypergraph. These classes are denoted by $\mathcal{V}_1, \ldots, \mathcal{V}_k$, and their sizes are given by $n_1, \ldots, n_k$.
ψ or $\psi_1, \ldots, \psi_n$	True (or planted) labels of vertices of hypergraph.
ψ' or $\psi'_1, \ldots, \psi'_n$	Labels of vertices estimated by partitioning algorithm.
$\mathrm{Error}(\psi, \psi')$	Clustering error, or number of disagreements between ψ and ψ' upto possible permutation of labels. See (2.20). We suffix this quantity with algorithm name to refer to the error incurred by a specific algorithm.
Z	Membership or cluster assignment matrix, $Z_{i\psi_i} = 1$, and zero otherwise.
α_m	Parameter controlling sparsity edges of size m in a random hypergraph. See Sections 3.3 and 5.1. In Section 2.3, we α_2 is simply written as α.
$\mathbf{B}^{(m)}$	Tensor specifying probabilities of edges of size m among different classes in a planted hypergraph. See Sections 3.3 and 5.1. In Section 2.3, we $\mathbf{B}^{(2)}$ is written as B.
$\mathcal{A}$	Population adjacency matrix, $\mathcal{A} = \mathsf{E}[A]$, where A is (weighted) adjacency matrix of reduced hypergraph.
$\mathcal{D}$	$\mathcal{D} = \mathsf{E}[D]$, expectation with respect to probability measure of planted model.
$\mathcal{L}$	Population version of normalized Laplacian, $\mathcal{L} = I - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$.
$\mathcal{D}_{\min}$ and d	$\mathcal{D}_{\min} = \min_{i \in \mathcal{V}} \mathcal{D}_{ii}$, and $d = \min_{i \in \mathcal{V}} \mathsf{E}[\deg(i)]$. They coincide in the case of algorithms presented in Chapter 5.
$\mathcal{A}_{\min}$	$\mathcal{A}_{\min} = \min\{\mathcal{A}_{ij} : \mathcal{A}_{ij} > 0\}$.
δ	A quantifier for identifiability of partition in planted model. Technically, δ is a lower bound on eigen gap between k^{th} and $(k+1)^{th}$ eigenvalues of $\mathcal{L}$ or $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$.
X, $\overline{X}$ and $\mathcal{X}$	X is the matrix of orthonormal eigenvectors computed in the presented spectral methods. $\overline{X}$ is similar to X, but each row normalized to have unit norm. $\mathcal{X}$ is population version of X.
γ	Approximation factor of approximate k-means algorithms. See Section 2.5.3.
$\widehat{A}$	Unbiased estimator of A when edge sampling is used.
$\widehat{D}$	Unbiased estimator of D when edge sampling is used.

Contents

'Begin at the beginning,' the King said, very gravely,
'and go on till you come to the end: then stop.'

Lewis Carroll, *Alice's Adventures in Wonderland*

Chapter 1

Introduction

The study of networks plays a key role in analyzing relational data. For nearly a century, sociologists have relied on network analysis to understand various aspects of social, political and economic systems. Recent advances in biological sciences as well as the "rise of social networking" have further escalated the importance of network analysis by providing a plethora of applications that spread across various fields of science and engineering. The world wide web (Berners-Lee and Fischett, 2000) and the neural network (Lettvin et al., 1959) merely demonstrate the significance of networks in the progress of both science and society. While the applications are plentiful, the resulting problems are quite challenging. The theoretical and computational challenges of network analysis have intrigued researchers over several decades. In fact, the design of efficient algorithms for network analysis is still one of major research areas in several disciplines including statistics, communication, machine learning, game theory among others.

A crucial problem in the field of network sciences has been clustering or community detection. On one hand, it is the main tool for sociologists to extract the properties of the network, whereas on the other hand, clustering plays a major role in problems such as balancing server loads of websites, or determining functional relationships in biological networks. Due to the ubiquity of networks involving pairwise interactions, for instance friendship networks or communication networks, the bulk of the literature invariably assume that the network can be viewed as a graph. Subsequently, the network clustering problem boils down to a *graph partitioning* problem that has been extensively studied in mathematics and computer science.

Informally, the objective of the graph partitioning problem is to divide a graph into smaller components with 'specific' properties. For example, one often desires components of comparable sizes with low connectivity between them. In spite of the hardness of this *balanced* graph partitioning problem (Garey and Johnson, 1979), a number of efficient methods have been

developed in the literature. These algorithms use a wide variety of approaches including spectral techniques (Fiedler, 1973; Ng et al., 2002), modularity and likelihood based methods (Girvan and Newman, 2002; Bickel and Chen, 2009), convex optimization (Arora et al., 2004; Chen et al., 2014), belief propagation (Decelle et al., 2011) among others.

These methods are still viewed as heuristic solutions for the partitioning problem, but their empirical success have been widely appreciated in several engineering applications such as community detection in social or biological networks (Wasserman, 1994; Guimera and Amaral, 2005), electronic circuit design (Kernighan and Lin, 1970), data analysis (Ng et al., 2002), image processing and computer vision (Shi and Malik, 2000) among others. It is not an exaggeration to say that, in recent years, balanced partitioning has been one of the most prominent graph problems from a practical point of view. However, there are also other graph partitioning problems that have been of significant interest to practitioners. For instance, *graph coloring* has often been used to solve several scheduling and resource allocation problems (Chaitin, 1982), while the problem of finding *cliques* in a graph has a long history in network analysis (Luce and Perry, 1949).

In spite of the vast applicability of network modeling using graphs, there exist more complex networks, where pairwise interactions cannot accurately model the system of interest. A common example is folksonomy, where individuals annotate on-line resources, such as images or research papers. Such problems appear to have a tri-partite structure in form of "user–resource–annotation", and is naturally represented as a 3-uniform hypergraph (Ghoshal et al., 2009), where each edge connects exactly three vertices. The necessity of uniform hypergraphs has also been observed in data analysis (Gibson et al., 2000; Agarwal et al., 2005), in particular when one deals with problems involving complex similarity measures defined over multiple data instances. Moreover, certain networks encountered in databases (Boley, 1977), VLSI circuit design (Karypis and Kumar, 2000), bioinformatics (Michoel and Nachtergaele, 2012) as well as other domains involve group interactions, and can only be modeled as non-uniform hypergraphs. A classic example is a market transaction database, where each transaction naturally corresponds to a connection among the involved commodities (Guha et al., 2000).

Generalization of graph problems to the case of hypergraphs is not a new area of research. In fact, study of hypergraphs can be dated back to the early 20^{th} century; the notions of Property B (Bernstein, 1908) and matroids (Whitney, 1935), as well as the result of Sperner (Sperner, 1928), which were originally stated in terms of set systems, can be equivalently presented in the language of hypergraphs. However, Berge (1984) dates the beginning of extensive studies on hypergraphs to the 1960s, which saw a number of remarkable results that could generalize, and at the same time simplify, results of graph theory. In the words of Berge (1984):

> *"It was noticed that this generalisation often led to simplification; moreover, one single statement, sometimes remarkably simple, could unify several theorems on graphs."*

However, the results studied in that era mostly had a mathematical flavor, and could hardly establish the real necessity of hypergraph modeling. The first instance of the use of hypergraphs in engineering appeared in the works of Boley (1977) and Schweikert and Kernighan (1979), where it was observed that hypergraphs are more appropriate than graphs for modeling databases and electrical logic circuits, respectively. The latter work has been seriously pursued in the VLSI community and has resulted in hMETIS (Karypis and Kumar, 2000), which is the most popular hypergraph partitioning algorithm as well as the most common large scale circuit partitioning tool till date. In the spirit of Berge's statement, one may say that the use of hypergraphs simplifies the problem description, and hence, leads to development of natural and accurate solution strategies.

When one discusses about *hypergraph partitioning*, which is in fact the subject of this thesis, one can observe that there is a rapid increase in the study of this problem in various research communities. Earlier, one would have invariably associated the hypergraph partitioning problem to circuit design (Schweikert and Kernighan, 1979; Karypis and Kumar, 2000). But since the "turn of the century", this problem has certainly gained significance in terms of practical applicability. The versatility of hypergraph modeling has made hypergraph partitioning a natural solution to several applications in parallel computing (Catalyurek and Aykanat, 1999), database systems (Gibson et al., 2000), computer vision (Agarwal et al., 2005), machine learning (Zhou et al., 2007) and biology (Michoel and Nachtergaele, 2012). Related problems such as hypergraph coloring have also found use in computer architecture (Capitanio et al., 1995) and communication systems (Wu et al., 2015).

A special class of hypergraphs, called m-uniform hypergraphs, has also gained significance in both theory and practice, and particularly in the context of partitioning. Such hypergraphs have edges of a fixed cardinality (say, m), and can be seen as an immediate generalization of graphs, which are essentially 2-uniform hypergraphs. The breadth of graph partitioning methods has expanded to such an extent that one often poses standard data clustering problem in terms of partitioning a *similarity graph*, whose vertices correspond to data instances and edges provide information about pairwise similarities among vertices. Uniform hypergraph partitioning broadens the horizon for this data clustering strategy, and has been of particular interest in computer vision, where the problem requires computation of multi-way similarities among data points. For instance, in subspace clustering (Agarwal et al., 2005; Chen and Lerman, 2009) and geometric grouping (Govindu, 2005; Arias-Castro et al., 2011), one often

constructs a weighted uniform hypergraph, where the weight of each edge is computed from the error of fitting a particular geometric model through a set of data points. Even from a theoretical stand point, uniform hypergraphs have recently gathered more attention than their non-uniform counterparts. The reason for this is the alternative characterization of uniform hypergraphs in terms of its adjacency tensor. Note that unlike graphs, which can be neatly represented in terms of adjacency matrices, general hypergraphs do not comply with a simple representation. However, in the case of m-uniform hypergraphs, the adjacencies can be expressed by means of a symmetric tensor of order m. This allows one to use results from tensor theory (Qi, 2005) to comment on the algebraic connectivity (Hu and Qi, 2012) and chromatic number (Cooper and Dutle, 2012) of uniform hypergraphs.

It is needless to mention that the computational hardness of the partitioning problem does not get any easier when one shifts the focus from graphs to hypergraphs (Khot, 2002; Khot and Saket, 2014). Hence, heuristics approaches are "the order of the day". The hypergraph partitioning heuristics studied in the literature exhibit great diversity, perhaps even more than the variety in graph partitioning techniques. The methods range from combinatorial move based (Schweikert and Kernighan, 1979) or multi-level approaches (Karypis and Kumar, 2000) to hypergraph reduction techniques (Hadley, 1995; Chen and Frieze, 1996) and spectral algorithms (Rodríguez, 2002; Zhou et al., 2007), and even extend to tensor decomposition (Govindu, 2005) and other optimization methods (Rota Bulo and Pelillo, 2013).

Graph and hypergraph partitioning algorithms have been successfully used to solve several practical problems, and comparative studies among various approaches are scattered around the literature. But, a clear picture can only be achieved if these experimental findings also follow from a theoretical comparison of the algorithms. This poses a crucial question:

> **Question 1.** What is an appropriate theoretical unit for measuring 'goodness' of a network partitioning algorithm?

To this end, one may note that partitioning essentially refers to a clustering of the vertices of the network, and so, a theoretical framework for analyzing clustering algorithms also suits the purpose. However, unlike supervised learning problems such as classification or regression, a clustering problem or even other unsupervised tasks do not naturally involve concepts like *true labels* or *empirical risk* (Vapnik, 1998). This makes it quite tricky to formalize a theoretical study of unsupervised learning algorithms.

A natural approach arises from the observation that clustering algorithms typically optimize a certain objective. While methods like k-means algorithm (Lloyd, 1982; Ostrovsky et al., 2012) aim for a distance minimization objective, likelihood based approaches, like expectation

maximization (Dempster et al., 1977), provide solutions that are local maxima for the likelihood function. Similar objectives exist even in the case of graph partitioning. For instance, graph theorists often pose a partitioning problem as that of finding a *balanced min cut* of graph. Alternate, and more popular, objectives include minimization of *s-t cut*, *ratio cut* or *normalized cut* (von Luxburg, 2007), or maximization of *normalized associativity* (Shi and Malik, 2000) among others. In this context, one may ask whether an algorithm indeed achieves the global optimum of the objective. To this end, classical results in spectral graph theory and in particular isoperimetric inequalities show that methods based on spectral properties of the graph Laplacian can provide reasonably good solutions (Chung, 1997). Similar observations have been extended to hypergraphs as well (Friedman and Wigderson, 1995; Bolla, 1993; Zhou et al., 2007). Recent works show that better solutions can be achieved using alternate optimization techniques that solve tighter relaxation of the cut problems (Bühler and Hein, 2009; Rangapuram et al., 2014).

Though the above results provide some insights into the appropriateness of certain partitioning algorithms, they are unable to provide any quantitative information about the accuracy of different methods. A partial answer to this problem can be found in (Ng et al., 2002; Kannan et al., 2004; Peng et al., 2015), where it is shown that under certain conditions, one can provide quantitative guarantees on the 'goodness'[1] of the solution obtained from certain spectral algorithms. It is not too far-fetched to think that this quantitative theoretical guarantees have been the important factor that has led to the immense popularity of the algorithm proposed by Ng et al. (2002), which is presently well known as *normalized spectral clustering* or simply *spectral clustering*. While these works provide worst case guarantees on graph partitioning, the analysis need not hold in more general, and practical, situations. For instance, the results in (Ng et al., 2002) only when the input graph closely resembles an 'ideal graph' with k disjoint cliques, which is an unlikely situation in practice. Moreover, unlike a learning theory setting, this result does not consider a statistical framework, where the data (or rather, the graph in the present context) is obtained from a generative model.

This is an appropriate stage to recall the notion of *consistency* often used in statistical learning theory (Vapnik, 1998). Here, one assumes that the training samples are obtained from a particular probability measure. An algorithm is said to be consistent if the expected misclassification error for the algorithm converges in probability to the optimal (Bayes) error as the number of training samples increases.

A major step towards a statistical study of graph partitioning was first considered in (von

[1] To preserve the simplicity of the text, we avoid explicit definitions for this notion. At this stage, we only mention that such goodness measures are algorithm specific, and more importantly, they are not related to any standard notion of accuracy used in practice. Further details can be found in (Ng et al., 2002; Kannan et al., 2004), but in the present work, we use the clustering error to measure performance of an algorithm.

Luxburg et al., 2008; Shi et al., 2009), where the authors extended the notion of consistency to the case of spectral clustering and its variants. Limiting the study to the case of *similarity graphs* constructed from data instances, von Luxburg et al. (2008) established that if the data is randomly generated from a specified probability measure, then the solution obtained from normalized spectral clustering eventually converges to the optimal clustering. These results are quite general provided that one restricts the study to similarity graphs. Though this class of graphs has widespread applications in data analysis, their use is limited in the context real world networks. Furthermore, the above mentioned studies do not provide room for characterizing the algorithmic performance in terms of the clustering error, and cannot shed any light on the behavior of an approach in the finite sample case. The last observation is in contrast with the classical notion of consistency that can be easily translated to provide high probability error bounds for supervised learning algorithms.

It turns out that the gaps in the previous theoretical frameworks can be filled up by studying the performance on partitioning algorithms on random graphs. Random graphs originated in the works of Erdös and Rényi (1959) and Gilbert (1959), and was initially used to study connectivity properties of graphs. Informally, a random graph on n vertices is constructed by adding edges independently to edge set with a pre-defined probability[1]. Since its inception, the study of random graphs, and even random uniform hypergraphs, has been an exciting field of research for theoreticians from various backgrounds, including mathematics, physics and computer science. It has been proved over and over again that a rigorous analysis of random graphs and hypergraphs provides insights into interesting physical phenomena such as percolation and phase transition, that arise in the context of networks (Erdös and Rényi, 1959; Aizenman and Barsky, 1987), satisfiability problems (Achlioptas and Coja-Oghlan, 2008; Panagiotou and Coja-Oghlan, 2012) and numerous other situations.

Classical random graphs are not very useful if one is interested in 'testing' the goodness of partitioning algorithms. The simple reason for this is that a random graph need not exhibit an appropriate partition of the vertices. Hence, one needs to look for so-called non-classical random graph models, where the edges are not identically distributed. The particular model that is of considerable significance in the present context is the *stochastic block model* or the *planted partition model*. This model was originally proposed by sociologists to model the presence of communities in random networks (Holland et al., 1983). Later, McSherry (2001) used the same model to study the asymptotic properties of a spectral partitioning algorithm. In the planted model, one considers a random graph on n vertices with a well defined k-way partition. Let

[1] This model is usually referred to as a binomial random graph since the vertex degrees follow binomial distribution. There also exists alternatives construction, which we do not discuss here.

$\psi : \{1, \ldots, n\} \to \{1, \ldots, k\}$ denote the true labeling function for the n vertices. The edges are randomly added with probabilities depending on the class labels of the participating vertices. Goodness of graph (or hypergraph) partitioning algorithm is measured through a formal version of Question 1.

> **Question 2.** Let ψ' be the partition obtained from an algorithm, then what is the number of disagreements between ψ and ψ'?

One typically asks for a high probability bound on the above error in terms of n. Such error bounds have been established for a variety of partitioning algorithms including spectral algorithms (McSherry, 2001; Rohe et al., 2011; Krzakala et al., 2013), modularity and likelihood based methods (Bickel and Chen, 2009; Choi et al., 2012), convex optimization (Amini and Levina, 2014; Chen et al., 2014), belief propagation (Mossel et al., 2013a) among others. Chen et al. (2014) compare the theoretical guarantees for many of these approaches.

While spectral methods were among the first studied algorithms under the block model (McSherry, 2001), the popular variant of spectral clustering was studied only in recent times (Rohe et al., 2011; Lei and Rinaldo, 2015). It is now known that for a planted graph with $\Omega(\ln n)$ minimum vertex degree, spectral clustering has an error rate that is sub-linear in n. This property of achieving $o(n)$ error is commonly referred to as the *weak consistency* of spectral clustering. This is not the best known error rate as exact recovery of the partition is known to be possible using other approaches (Amini and Levina, 2014). Recent results (Vu, 2014; Lei and Zhu, 2014) show that an additional refinement process can improve the partitioning of spectral clustering to exactly recover the partition, which leads to *strongly consistent* algorithms. One can easily relate such definitions for consistency with the finite sample error bounds studied for supervised learning algorithms.

What is this thesis about?

In contrast to the extensive studies on graph partitioning, very little is known about the guarantees of hypergraph partitioning algorithms, and only few attempts have been made in the literature to study the problem in some special cases of uniform hypergraphs partitioning. The first known attempt was made by Chen and Frieze (1996), who analyzed a spectral algorithm for coloring 3-uniform hypergraphs. More recently, Chen and Lerman (2009) extended the perturbation analysis of Ng et al. (2002) to analyze a tensor decomposition based subspace clustering approach. Arias-Castro et al. (2011) considered a different line of study, and presented a probabilistic model for generating random data from a union of geometric structures.

The analysis of their tensor based approach in a way generalizes the consistency result of von Luxburg et al. (2008).

While the above results shed some light on the performance of spectral methods used in multi-way similarity based clustering or hypergraph coloring, the general question of "goodness of hypergraph partitioning algorithms" is still open. The main aim of this thesis is to fill this gap by providing performance guarantees and consistency results for uniform and non-uniform hypergraph partitioning algorithms. As discussed earlier, a natural way of treating this problem is in terms of Question 2, where a random hypergraph is generated from a planted model. To the best of our knowledge, the only attempt to address this question was done in the special case of coloring 3-uniform hypergraphs (Chen and Frieze, 1996). We provide an answer in the full generality of the planted hypergraph model, and also address practical aspects of hypergraph partitioning.

1.1 Revisiting the essentials

We rewind to the beginning of this chapter, and clarify the essential components in simple mathematical terms and with examples.

1.1.1 Network: Graph or hypergraph?

The term 'network' has various definitions based on the context – it may mean a communication network, the social connections formed in a networking website, molecular interactions in a cell or various other physical or conceptual phenomena. But, essentially any network can be thought of as a collection $\mathcal{V}$ of entities, where subsets of entities are related or connected in some sense. Let us denote the collection of all such connections by $\mathcal{E}$. Note that each $e \in \mathcal{E}$ is a subset of $\mathcal{V}$.

A simple, and popular, example is a friendship network. Here, the entities are people, and a two-sided connection occurs whenever two people are friends on a social networking site, say Facebook. In this case, every connection involves exactly two entities. Such a network can be represented as a graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of vertices, and $\mathcal{E}$ is the set of edges, each edge connecting one vertex to another.

On the other hand, consider the network of Google groups. Here, the entities are various email accounts, and several accounts form a group. One may model this network in the above way, where $\mathcal{V}$ is the collection of accounts, and each $e \in \mathcal{E}$ denotes a group. It would be hard to imagine that every Google group consists of exactly two members, and hence, in practice, such a network cannot be modeled as a graph. However, one can easily represent it as a

hypergraph $(\mathcal{V}, \mathcal{E})$, where the edges $e \in \mathcal{E}$ need not be of size two. Such edges are often termed as 'hyperedges'. However, we prefer to ignore this distinction, and hence, we use the term 'edge' to refer to both edges in graphs and hypergraphs.

Finally, consider a situation where every group has exactly m members, for example $m = 11$ if each group is a football or cricket team. Then the corresponding hypergraph is said to be *m-uniform*. Thus, the use of a graph or a hypergraph model depends on the network and the application at hand. One can also think of further generalizations, where every group has a number or weight associated, such as number of posts in a Google group. This network is appropriately modeled by a *weighted hypergraph* $(\mathcal{V}, \mathcal{E}, w)$, where every edge $e \in \mathcal{E}$ has an associated weight $w(e)$. Here, $w : \mathcal{E} \to \mathbb{R}$ is some predefined scalar function.

1.1.2 From community detection to data clustering

Typically, a *community* denotes a subset of vertices in a network that have a high number of edges amongst themselves. Community detection stands for the task of detecting multiple communities in a network. However, one often uses this term to refer to the problem of dividing a network into several communities, which is also known as network partitioning. Formally, the task is to partition $\mathcal{V}$ into k disjoint sets, $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_k$, to meet certain specifications.

Several applications of community detection have been mentioned early in this chapter. As an example, consider the friendship network of Facebook that is obviously too large to be stored on a single server. While dividing the load into multiple servers, one would prefer to minimize server communications. Thus, it is advantageous to allocate each community to a single server. Observe that if the objective involves distribution of comparable load among the servers, while reducing server communication, then the problem is similar to a balanced cut problem.

One may argue that community detection is equivalent to clustering the vertices of a network based on 'edge information'. The scope of network partitioning expands greatly if one exploits the above point of view, and constructs networks based on similarities of data instances. A classic example is the ϵ-neighborhood graph, where every data point is seen as a vertex of the graph, and is connected to other points within an ϵ distance from itself. Subsequently, a partition of the similarity network provides a clustering of the given data. Hence, a spectral partitioning method is often coined as *spectral clustering*.

More generally, given a collection of data points $x_1, \ldots, x_n$ and a symmetric pairwise similarity measure $f(\cdot, \cdot)$, one can define a similarity graph with weighted edges such that $w(\{x_i, x_j\}) = f(x_i, x_j)$. This idea also extends to m-uniform hypergraphs when f measures similarity among m data points. For instance, consider the situation where $x_1, \ldots, x_n$ belong

to a union of k intersecting lines (one-dimensional subspaces). In this case, pairwise similarities are inappropriate and one needs to check collinearity of three or more points. This leads to a m-uniform hypergraph with $m \geq 3$, and edges correspond to collinearity of m points. More general applications are discussed later in Chapter 2.

1.1.3 What is random in 'random networks'?

Random graphs and hypergraphs are repeatedly mentioned throughout the thesis. So it is necessary to segregate the deterministic and random components of these structures. Let $(\mathcal{V}, \mathcal{E})$ denote a random graph or hypergraph with $|\mathcal{V}| = n$. We will assume the set of vertices to be deterministic, and will denote this set as $\mathcal{V} = \{1, 2, \ldots, n\}$. Often, one studies the behavior of these networks in the asymptotic case as $n \to \infty$. However, the analysis is typically carried out after fixing n. Furthermore, in a planted partition model, we assume that every vertex in $\mathcal{V}$ has a deterministic class label.

The randomness of these networks is associated with the presence of edges, *i.e*, for any subset $e \subset \mathcal{V}$, there is a probability associated with the event $\{e \in \mathcal{E}\}$. Hence, the collection $\mathcal{E}$ is random. We also consider random weighted hypergraphs $(\mathcal{V}, \mathcal{E}, w)$. Here, we assume that $\mathcal{E}$ is the collection of all subsets of $\mathcal{V}$, and for every $e \in \mathcal{E}$, $w(e)$ is a random variable. One may note that, in a weighted hypergraph, the absence of an edge e is equivalent to setting $w(e) = 0$. As is standard with random graphs, the model considered in the thesis assumes the events $\{e \in \mathcal{E}\}$ to be mutually independent for all $e \subset \mathcal{V}$. Similarly, the random variables $\{w(e) : e \in \mathcal{E}\}$ are assumed to be mutually independent.

The Erdös-Rényi model is particular example of random networks, where all edges (or edge weights) are independent and identically distributed. We consider models for random networks in which all the edges do not follow the same law.

1.1.4 The spectral connection

It is quite surprising to see that spectral theory often provides answers about several combinatorial problems related to graphs and hypergraphs. The primary spectral connection of graph $(\mathcal{V}, \mathcal{E})$ is through its adjacency matrix $A \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{V}|}$, where $A_{ij} = 1$ if and only if $\{i, j\} \in \mathcal{E}$. The definition also extends to weighted graphs $(\mathcal{V}, \mathcal{E}, w)$, where one defines a weighted adjacency matrix, or affinity matrix, $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ with $A_{ij} = w(\{i, j\})$. It turns out that the eigenvalues and eigenvectors of A, or related matrices, provide insights into several properties of the corresponding graph, such as connectivity, colorability etc. (Spielman, 2011; Chung, 1997).

The spectral theory of graphs has also been extended to m-uniform hypergraphs, where the adjacencies (or edge weights) can be represented by a m-way tensor $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|\times\ldots\times|\mathcal{V}|}$. For a set $e = \{i_1, \ldots, i_m\} \subset \mathcal{V}$ in an unweighted m-uniform hypergraph, $\mathbf{A}_{i_1\ldots i_m} = 1$ if $e \in \mathcal{E}$, and 0 otherwise. On the other hand $\mathbf{A}_{i_1\ldots i_m} = w(e)$ in weighted m-uniform hypergraphs. This representation enables one to comment on several uniform hypergraph problems by studying the spectral properties of the adjacency tensor (Qi, 2005).

We later discuss how graph or hypergraph partitioning can be relaxed into a spectral problem, and present algorithms that provides a good partition by simply exploiting spectral decompositions of adjacency matrices and tensors, as well as related quantities.

1.2 Summary of contributions

The main purpose of this thesis is a statistical treatment of spectral algorithms used for hypergraph partitioning. This is achieved in the form of a two-fold contribution: proposing a model for random planted hypergraphs, and analysis of spectral algorithms under this model. We provide some extended results related to consistency of partitioning sampled hypergraphs, and also for the hypergraph coloring problem.

1.2.1 Planted partition model

We present a model for random hypergraphs that naturally extends the stochastic block model. The requirements for such a model are: (i) the presence of a planted partition of the vertices, and (ii) independent edges with label dependent probabilities. The challenge lies in formulating a model that is appropriate for analysis of partitioning algorithms, and at the same time, conforms with characteristics of real-world hypergraphs.

Extensions of random graph models (Erdös and Rényi, 1959) to uniform hypergraphs have been often considered in theoretical computer science (Achlioptas and Coja-Oghlan, 2008; Feldman et al., 2015). In a version of this model, a m-uniform hypergraph is generated by independently adding m-sized subsets of vertices to the edge set with a fixed probability. The proposed planted model generalizes the above construction by allowing label dependent probabilities for edges. The novelty of our model lies in viewing the uniform hypergraph in terms of its adjacency tensor, which in turn leads to a simple, yet general, specification of the model parameters. The model also allows the possibility of sparse hypergraphs, or the presence of weighted edges.

Non-uniform hypergraph generalizations of the Erdös-Rényi model have received less attention in the literature (Schmidt-Pruzan and Shamir, 1985; Darling and Norris, 2005; Stasi

et al., 2014). The key observation here is that one can consider a non-uniform hypergraph as a collection of m-uniform hypergraphs for varying m. Based on this, we present a planted hypergraph model that constructs a non-uniform hypergraph by independently generating a sequence of uniform hypergraphs, each with a different size of edges.

1.2.2 Consistency of spectral methods

We complement the proposed model with an immediate application in the study of spectral partitioning algorithms, and in particular provide an answer to Question 2.

We analyze the approach studied by Govindu (2005), which elegantly extends spectral clustering to uniform hypergraphs by means of tensor decompositions. We observe that under certain conditions, the algorithm is weakly consistent. Moreover, in particular cases, the algorithm also exhibits strong consistency. In a search for algorithms with smaller error rates than the above method, we formulate the hypergraph partitioning problem from the "first principles" defined in the graph case. We observe that in the case of uniform hypergraphs, the problem of finding a partition that maximizes normalized associativity is equivalent to a tensor trace maximization problem. Surprisingly, this formulation generalizes a wide class of techniques used in the machine learning, often termed as higher order learning algorithms. We show that a spectral relaxation of the tensor trace maximization problem results in a weakly consistent algorithm that has improved theoretical, as well as empirical, performance as compared to the algorithm in (Govindu, 2005).

We next focus on non-uniform hypergraphs, and consider a popular spectral algorithm that solves the normalized cut minimization problem for hypergraphs (Zhou et al., 2007). Analysis of this method under a planted non-uniform hypergraph model proves its consistency properties under certain sufficient conditions. At this stage, we also scrutinize some necessary conditions for identifiability of the partition by a spectral algorithm, and observe that the algorithm of Zhou et al. (2007) is capable of identifying the partition in reasonable circumstances. We also briefly discuss an extension of the tensor trace maximization approach that solves the normalized associativity maximization problem for non-uniform hypergraphs. The theoretical properties of this approach can be argued to be similar to the previous method. The study of both these techniques nearly accounts for all hypergraph reduction techniques via clique or star expansions (Agarwal et al., 2006).

Our analysis of the spectral methods rely on few important tricks: (i) a suitable characterization of the adjacencies in a random hypergraph, and the associated matrices and tensors; (ii) the use of matrix concentration inequalities (Tropp, 2012) that were previously used for

studying spectral properties of sparse random graphs; (iii) matrix perturbation analysis (Davis and Kahan, 1970; Stewart and Sun, 1990); and (iv) a rigorous study of the the distance based clustering involved in a spectral partitioning algorithm. We spend few more words on this component. Typically, spectral partitioning algorithms involve a post-processing stage of distance based clustering. Though the k-means algorithm (Lloyd, 1982) or its approximate variants (Kumar et al., 2004; Ostrovsky et al., 2012) are the popular choice in practice, such algorithms are not always guaranteed to provide good clustering. Gao et al. (2015) discusses the implication of this drawback on the consistency results for spectral clustering under the block model (Lei and Rinaldo, 2015). On the other hand, we establish that under certain conditions, the approximate k-means algorithm of Ostrovsky et al. (2012) provides a good clustering with high probability, thereby addressing a long standing question in the block model literature.

1.2.3 Edge sampling in planted hypergraphs

The time complexity of graph or hypergraph partitioning is typically linear in the number of edges. This limits the practical use of hypergraph partitioning algorithms, particularly when the constructed hypergraph has a large number of weighted edges, a situation commonly encountered in computer vision problems. Hence, one often studies efficient methods that compute weights for only a small number of edges (Chen and Lerman, 2009; Duchenne et al., 2011). To this end, the following question has considerable significance.

> **Question 3.** Consider a weighted hypergraph $(\mathcal{V}, \mathcal{E}, w)$. What is the smallest N such that consistent partitioning of the hypergraph can be achieved by using the weights of only $N \ll |\mathcal{E}|$ sampled edges?

We answer the above question by studying the statistical behavior of a sampled variant of the tensor trace maximization approach. We show that if an appropriate sampling strategy is used, then computing the weights of few sampled edges suffices to achieve a consistent partitioning. In particular, frequent sampling edges with larger weights is observed to be provably better than uniform edge sampling. We further propose a sampled algorithm that achieves state of the art performance in benchmark subspace clustering problems.

1.2.4 Spectral hypergraph coloring

The final contribution in this thesis is a spectral algorithm for 2-coloring the vertices of a non-uniform hypergraph. This problem is computationally hard, and no polynomial time algorithm is known till date that can color a bipartite non-uniform hypergraph with a constant number of

colors. We present a spectral coloring algorithm and show that if a random bipartite hypergraph is generated from a planted partition model, then our algorithm succeeds, with high probability, in properly coloring the hypergraph with exactly two colors. The significance of this study in the context of the thesis is due to the following fact. While the other algorithms studied here are weakly consistent in general, our coloring algorithm incorporates an additional refinement procedure that helps to achieve strong consistency.

1.3 Organization of the thesis

We briefly summarize the results that are presented in the subsequent chapters of this thesis.

Chapter 2. Here, we discuss in detail some of the background materials that are essential for subsequent technical developments of the thesis.

Chapter 3. This is the first contributing chapter of the thesis. We begin with a brief description of higher order singular value decomposition (HOSVD) of tensors that forms the basis of the uniform hypergraph partitioning algorithm proposed by Govindu (2005). We also list this algorithm, which we refer to as HOSVD.

We discuss a perturbation type analysis for the algorithm, and comments on the drawbacks. Subsequently, we present the planted partition model for uniform hypergraphs, and derive a consistency result for the HOSVD algorithm.

Chapter 4. We continue our quest for uniform hypergraph partitioning algorithm. We approach the partitioning problem from a graph theoretic aspect. To be precise, we propose to partition a hypergraph such that the normalized associativity of the partition is maximized. This objective lies at the heart of spectral clustering.

It is observed that in case of a uniform hypergraph, the normalized associativity maximization problem can be reformulated as a tensor trace maximization (TTM) problem. The interesting features of TTM include its connection with tensor eigenvalue problem, and tensor diagonalization problem. But more interestingly, we show that a wide variety of higher order learning algorithms solves different relaxations of TTM.

We present an algorithm, which we call as TTM, that solves a spectral relaxation of the above problem. Consistency of this algorithm is proved under the planted uniform hypergraph model, and as a consequence, it follows that TTM has better theoretical properties than HOSVD. Numerical comparison of different hypergraph partitioning algorithms are also provided.

Chapter 5. In this chapter, we consider the problem of non-uniform hypergraph partitioning. We begin by extending the random model to the case of non-uniform hypergraphs. We

present two partitioning algorithms: (i) a spectral method based on normalized hypergraph cut minimization (Zhou et al., 2007), and (ii) extension of the TTM algorithm to non-uniform hypergraphs that solves a the normalized associativity maximization problem. We refer to these two techniques as NH-Cut and NH-Assoc, respectively.

Consistency of NH-Cut is proved under the planted partition model, and arguments are given to show that NH-Assoc has similar consistency properties. As corollaries, we also study the performance of NH-Cut in special cases of the planted partition model, and in particular, compare our results with the existing results in the case of graphs.

Chapter 6. Here, we focus on efficient hypergraph partitioning, and provide a rigorous answer to Question 3 in the case of the TTM algorithm. Consistency of TTM is studied when one has access to few sampled edges of the hypergraph. Based on our result, we justify the success of sampling techniques that are popular in practice.

We also propose a sampled variant TTM, which we refer to as tensor trace maximization with iterative sampling or simply Tetris. The empirical efficacy of Tetris is demonstrated in subspace clustering problems, and benchmark motion segmentation data sets.

Chapter 7. This chapter deals with the hypergraph coloring problem. We present a special case of the planted hypergraph model that generates a 2-colorable non-uniform hypergraph with equal color classes. A spectral coloring algorithm, called COLOR, is proposed, and it is shown that under the planted model, this algorithm succeeds with high probability in coloring a random hypergraph with only two colors.

Chapter 8. We include the numerical studies in this chapter. The purpose of our experiments include both validation of the theoretical findings of this thesis, and demonstration of the empirical performance of spectral partitioning algorithm in benchmark problems.

Chapter 9. This is the concluding chapter, where we discuss our final thoughts on the results presented in the thesis. We also provide an elaborate account of possible future research directions related to hypergraph partitioning under a planted partition model.

But just in case some of you may be tempted to skip this particular section and go on to juicier things, let me assure you that there will be juice in plenty dripping from these pages. I wouldn't have it otherwise.

Roald Dahl, *My Uncle Oswald*

Chapter 2

Preliminaries and Background

In this chapter, we briefly review some topics that need explanation before plunging into the technical details and results in the thesis. Section 2.1 gives an overview of spectral theory that has evolved from matrices to tensors. Sections 2.2 and 2.3 provides a quick recap of the graph partitioning problem, including spectral approach, and the stochastic block model for analyzing graph partitioning methods, respectively. We present a review of the hypergraph partitioning literature in Section 2.4, and conclude this chapter with a list of standard results in Section 2.5 that are used in the subsequent chapters. One may refer to the front matter for the list of notations and abbreviations used in this thesis.

2.1 Spectral theory: From matrices to tensors

Before beginning our journey on network partitioning, we present an overview of spectral theory of both matrices and tensors. This discussion is crucial since the partitioning algorithms that are studied in this thesis have a spectral flavor.

2.1.1 Spectral decomposition of matrices

Spectral theory, or more precisely matrix spectral theory, has often transcended the boundaries of linear algebra, and has played important roles in various branches of science and engineering. The core relation in matrix spectral theory is the following. Let $A \in \mathbb{R}^{n\times n}$ be a matrix, and let $\lambda \in \mathbb{R}$ and $u \in \mathbb{R}^n$ satisfy

$$Au = \lambda u. \tag{2.1}$$

Then λ is called an eigenvalue of A, and u is an eigenvector of A corresponding to λ. A consequence of this relation is more useful in our context, which states that if the matrix A is symmetric then

$$A = \sum_{i=1}^{n} \lambda_i u_i u_i^T = U \Lambda U^T \ , \tag{2.2}$$

where (λ_i, u_i) are eigen pairs of A with $u_1, \ldots, u_n$ being an orthonormal set of eigenvectors. In matrix notation, this can be written as $A = U\Lambda U^T$, where Λ is a diagonal matrix of eigenvalues, and U is the orthonormal eigenvector matrix. A similar decomposition exists for asymmetric matrices as well. If $A \in \mathbb{R}^{n \times \ell}$ with $n \leq \ell$, then one can express A as

$$A = \sum_{i=1}^{n} \sigma_i u_i v_i^T = U \Sigma V^T \ , \tag{2.3}$$

where $\sigma_1, \ldots, \sigma_n \in [0, \infty)$ are called singular values of A, and correspond to the entries of the principal diagonal of $\Sigma \in \mathbb{R}^{n \times \ell}$. The matrices $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{\ell \times \ell}$ are orthonormal with columns $u_1, \ldots, u_n$ and $v_1, \ldots, v_\ell$, respectively. These are called the left and right singular vectors of A.

One may refer to (Horn and Johnson, 2013) for further material on this topic. We take a different course and discuss how the above results can be generalized from two-dimensional arrays (matrices) to m-dimensional arrays for $m > 2$ (tensors). We conclude this discussion by introducing a terminology that will be followed throughout this thesis.

Definition 2.1. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. For $k \leq n$, we say $\lambda_1, \ldots, \lambda_k$ are the k *dominant* eigenvalues of A. The corresponding vectors $u_1, \ldots, u_k$ in (2.2) are called the k dominant orthonormal eigenvectors.

If A is positive semi-definite, *i.e*, $\lambda_n \geq 0$, then the eigenvalues $\lambda_{n-k+1}, \ldots, \lambda_n$ are called the k *leading* eigenvalues of A, and $u_{n-k+1}, \ldots, u_n$ are the k leading orthonormal eigenvectors.

In the asymmetric case (2.3), the notion of k dominant singular vectors will be used to refer to the columns of U and V that correspond to the k largest singular values of the matrix.

2.1.2 Tensors and basic operations

Let $\mathbf{A} \in \mathbb{R}^{n \times n \times \ldots \times n}$ be a tensor of order m. Note that it suffices our purpose to discuss only about tensors whose size is same along each dimension. In addition, we will often assume the tensor to be symmetric.

Definition 2.2. A tensor $\mathbf{A} \in \mathbb{R}^{n \times n \times \ldots \times n}$ of order m is said to be symmetric[1] if for any $i_1, \ldots, i_m \in \{1, \ldots, n\}$ and a permutation map σ on $\{1, \ldots, m\}$,

$$\mathbf{A}_{i_1 \ldots i_m} = \mathbf{A}_{i_{\sigma(1)} \ldots i_{\sigma(m)}} .$$

Before elaborating on tensor decompositions, it will be useful to describe few tensor notations and operations. Similar to the notion of trace of a matrix, a tensor trace denotes the sum of the diagonal entries, *i.e*, $\text{Trace}(\mathbf{A}) = \sum_{i=1}^{n} \mathbf{A}_{ii \ldots i}$. A matrix representation of tensor often comes in handy. One such representation involves defining a matrix $\widetilde{\mathbf{A}} \in \mathbb{R}^{n \times n^{m-1}}$ such that

$$\widetilde{\mathbf{A}}_{ij} = \mathbf{A}_{ii_2 \ldots i_m} \qquad \text{when } j = 1 + \sum_{\ell=2}^{m} (i_\ell - 1) n^{\ell-2}. \tag{2.4}$$

Here, $\widetilde{\mathbf{A}}$ is known as the flattened matrix, or more technically mode-1 flattened matrix, of $\mathbf{A}$. See illustration in Figure 2.1.

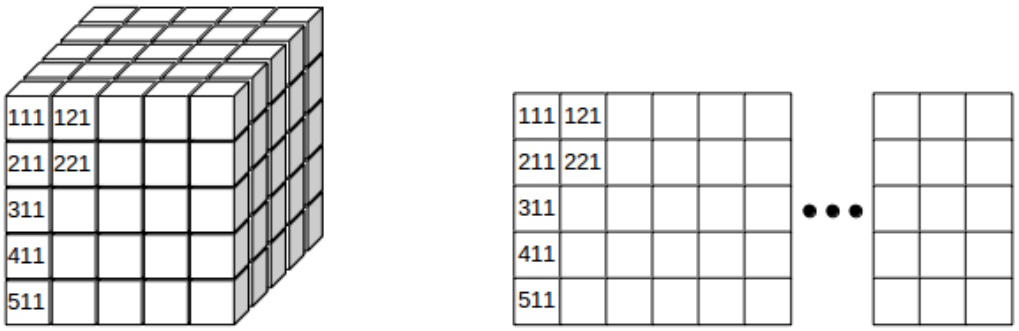

Figure 2.1: A $5 \times 5 \times 5$ tensor (left), and the corresponding 5×25 flattened matrix (right).

A special class of tensor, called a rank-one tensor, often arises in the context of decompositions.

Definition 2.3. A tensor $\mathbf{A} \in \mathbb{R}^{n \times n \times \ldots \times n}$ of order m is said to be of rank one if there exist m vectors $u^{(1)}, \ldots, u^{(m)} \in \mathbb{R}^n$

$$\mathbf{A}_{i_1 i_2 \ldots i_m} = u^{(1)}_{i_1} u^{(2)}_{i_2} \ldots u^{(m)}_{i_m}$$

for all $i_1, \ldots, i_m \in \{1, \ldots, n\}$. We denote such a tensor as $\mathbf{A} = u^{(1)} \otimes u^{(2)} \otimes \ldots u^{(m)}$.

Observe that Definition 2.3 naturally generalizes the notion of rank-one matrices to tensors, which useful in extending the decomposition in (2.3). An useful operation on a tensor is its

[1] In some works, this property is termed as super-symmetry of a tensor.

mode-k multiplication with a matrix, defined as follows.

Definition 2.4. Let $\mathbf{A} \in \mathbb{R}^{n\times n\times\ldots\times n}$ and $U \in \mathbb{R}^{r\times n}$. The mode-$k$ product of $\mathbf{A}$ and U is a m^{th}-order tensor, denoted by $\mathbf{A} \times_k U$, whose size is n along all dimensions except the k^{th} one, for which the the dimension is r. The entries of $\mathbf{A} \times_k U$ are given by

$$(\mathbf{A} \times_k U)_{i_1\ldots i_{k-1}ji_{k+1}\ldots i_m} = \sum_{i_k=1}^{n} \mathbf{A}_{i_1\ldots i_{k-1}i_k i_{k+1}\ldots i_m} U_{ji_k}.$$

for $i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_m \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, r\}$.

We will be mostly interested in repeated multiplications along all dimensions. For instance, if $\mathbf{A} \in \mathbb{R}^{n\times n\times\ldots\times n}$ is mode-k multiplied by $U^{(k)} \in \mathbb{R}^{r\times n}$ for every $k = 1, \ldots, m$, then the resultant m^{th}-order tensor $A \times_1 U^{(1)} \times_2 U^{(2)} \ldots \times_m U^{(m)} \in \mathbb{R}^{r\times r\times\ldots\times r}$, and has entries

$$(A \times_1 U^{(1)} \times_2 U^{(2)} \ldots \times_m U^{(m)})_{j_1\ldots j_m} = \sum_{i_1,\ldots,i_m=1}^{n} \mathbf{A}_{i_1\ldots i_m} U^{(1)}_{j_1 i_1} U^{(2)}_{j_2 i_2} \ldots U^{(m)}_{j_m i_m} \tag{2.5}$$

for $j_1, \ldots, j_m = 1, \ldots, r$. An interesting special case arises for $r = 1$, *i.e*, when a tensor is multiplied by m row vectors and the product is a scalar. Here, one can view a tensor $\mathbf{A}$ as a m-linear functional such that for any m vectors $u_1, \ldots, u_m \in \mathbb{R}^n$, we have

$$(u_1, \ldots, u_m) \mapsto \mathbf{A} \times_1 u_1^T \times_2 u_2^T \ldots \times_m u_m^T . \tag{2.6}$$

One may recall the case for $m = 2$, a matrix A can be thought of a bilinear functional such that $(u, v) \mapsto u^T Av$. This functional is significant in several problems, including Rayleigh's principle that provides an useful characterization of the eigen pairs of A. In the next section, we mention a generalization of this characterization to the case of tensors.

2.1.3 Tensors decompositions and spectral theory

This thesis does not involve all components of the spectral theory of tensors. Still we provide a brief overview of different aspects. A major impact of tensors in the machine learning community is due to the generalizations of the spectral decompositions (2.2) and (2.3) to the case of tensors. Quite interestingly, the two equivalent representations in (2.3) do not lead to the same generalization in the case of tensors of order 3 or more.

The representation in (2.2) states that a symmetric matrix $A \in \mathbb{R}^{n\times n}$ can be expressed as a sum of n rank-one matrices. The CP decomposition extends this result to the case of

tensors, and attempts to represent any tensor $\mathbf{A}$ of order m as sum of rank-one tensors of same order and dimension. CP stands for CANDECOMP/PARAFAC, which in turn denote canonical decomposition (Carroll and Chang, 1970) or parallel factorization (Hitchcock, 1927) of a tensor, respectively. These two decompositions were independently proposed, but result in the same representation. Formally, the CP decomposition is defined as follows.

Definition 2.5. Let $\mathbf{A} \in \mathbb{R}^{n \times \ldots \times n}$ be a tensor of order m. If there exists vectors $u_i^{(j)} \in \mathbb{R}^n$ for each $i = 1, \ldots, r$ and $j = 1, \ldots, m$ such that

$$\mathbf{A} = \sum_{i=1}^{r} u_i^{(1)} \otimes u_i^{(2)} \otimes \ldots u_i^{(m)} \ ,$$

then the above representation is said to be a CP decomposition of $\mathbf{A}$. The minimal $r \in \mathbb{N}$ for which such a representation exists is called the rank of $\mathbf{A}$.

This decomposition, illustrated in Figure 2.2 (top row), finds use in several problems in machine learning (Anandkumar et al., 2014) as well as in other disciplines (Kolda and Bader, 2009), and has received considerable attention in recent years (Anandkumar et al., 2014; Jain and Oh, 2014). However, a major drawback of the CP decomposition of tensors is that, unlike the case of matrices, this decomposition need not exist for tensors of order 3 or more. This limits the use of such a representation in a theoretical framework.

Alternatively, one could extend the second representation in (2.3), which essentially expresses A in terms of a *core* diagonal matrix, Σ, multiplied by orthonormal matrices along its two dimensions: U is multiplied along rows of Σ (left multiplication), while V is multiplied along the columns (right multiplication). An extension of this to the case of tensors is given by the Tucker decomposition (Tucker, 1966), or rather its more formal variant known as higher order singular value decomposition (HOSVD) (De Lathauwer et al., 2000). Unlike CP decomposition, this representation exists for all tensors and can be easily computed from eigendecompositions of certain matrices. A formal description of this decomposition is given below.

Definition 2.6. One can express any tensor $\mathbf{A} \in \mathbb{R}^{n \times \ldots \times n}$ of order m as

$$\mathbf{A} = \mathbf{\Sigma} \times_1 U_1 \times_2 U_2 \times_3 \ldots \times_m U_m \ ,$$

where $U_1, \ldots, U_m \in \mathbb{R}^{n \times n}$ are orthonormal matrices. The m^{th} order tensor $\mathbf{\Sigma} \in \mathbb{R}^{n \times \ldots \times n}$ is the core tensor that satisfies a certain property of all-orthogonality[1].

[1] We do not elaborate on the all-orthogonality property since it is complicated, and has no direct bearing on the discussions in this thesis.

Furthermore, if $\mathbf{A}$ is symmetric, then $U_1 = \ldots = U_m$, and this matrix can be computed as the matrix of the orthonormal left singular vectors of $\widetilde{\mathbf{A}}$ defined in (2.4).

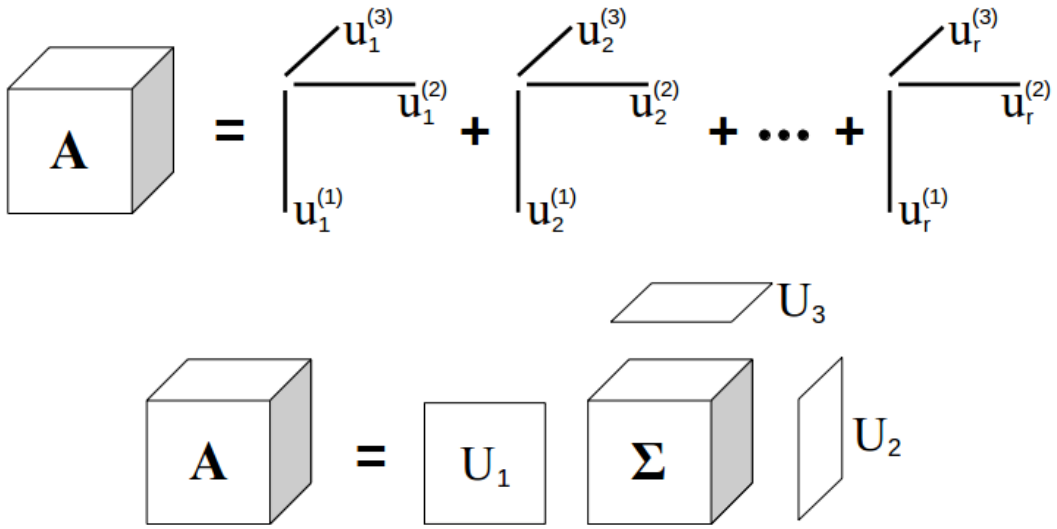

Figure 2.2: (top) CP-decomposition of a rank-r 3^{rd}-order tensor $\mathbf{A}$, and (bottom) higher order singular value decomposition of $\mathbf{A}$. Observe that $u_1^{(i)}, \ldots, u_r^{(i)}$ correspond to columns of U_i only when $\mathbf{\Sigma}$ is diagonal.

Unlike matrix theory, the research on tensor decompositions and the tensor eigenvalue problem have always been quite independent of one another. We briefly mention two variants of the eigenvalue value problem, one of which has a connection to the tensor trace maximization problem presented in Chapter 4.

Consider (2.6), where we relate a tensor to a m-linear functional. One can study the problem maximizing the functional over all possible argument vectors. For instance, consider the matrix case, *i.e*, $m = 2$. Let A be a symmetric matrix, then the optimization problem is stated as

$$\underset{u:\|u\|_2=1}{\text{maximize}}\ u^T A u\ . \tag{2.7}$$

It is well known that the solution to this problem is the largest eigenvalue of A, which is achieved when u is the dominant eigenvector. By imposing additional constraints on u leads to the famous Rayleigh's principle, which states that other eigen pairs are obtained as solution depending on the constraint space of the optimization problem. The moral here is that the eigenvectors of A are the stationary points for the problem (2.7). This forms the basis for the tensor eigenvectors defined by Lim (2005), which we state only for symmetric tensors. For a

symmetric tensor $\mathbf{A}$, one considers the following optimization

$$\underset{u:\|u\|_p=1}{\text{maximize}}\ \mathbf{A} \times_1 u^T \times_2 \ldots \times_m u^T \ , \tag{2.8}$$

and the stationary points for the problem are said to be the ℓ_p-eigenvectors of $\mathbf{A}$, and the corresponding values for the objective function are the ℓ_p-eigenvalues of $\mathbf{A}$. Similarly, the ℓ_2-eigenvectors obtained from (2.8) satisfy the relation

$$\mathbf{A} \times_2 u^T \times_3 \ldots \times_m u^T = \lambda u \ , \tag{2.9}$$

where $\mathbf{A}$ acts as as a multilinear transformation, which is linear in all its $(m-1)$ arguments. The above relation suggests that an ℓ_2-eigenvector, also called the Z-eigenvector, is simply scaled by the corresponding Z-eigenvalue under this transformation. This is known as the Z-eigenvalue problem. A similar study in the ℓ_m case has also been studied, which is popular termed as the H-eigenvalue problem (Qi, 2005).

2.2 Graph partitioning and spectral clustering

The graph partitioning problem has been formalized in several ways, which makes it quite difficult to review all possible approaches towards this problem. We briefly discuss a particular line of study along which several formal definitions of the problem have been proposed. This particular direction based on *cut* or *associativity* of partitions has mostly been popular in the machine learning community (von Luxburg, 2007), and is also related to *graph edge expansion* studied in theoretical computer science (Spielman, 2011). Alternative formal ways of graph partitioning have also been considered, which include *max flow* problem that plays role in optimization theory (Arora et al., 2004), *network modularity* studied in statistical physics (Girvan and Newman, 2002) among others.

2.2.1 Formal definitions of balanced graph partitioning

Given a graph $(\mathcal{V}, \mathcal{E})$, one of the principle approaches of k-way graph partitioning is to remove some edges from $\mathcal{E}$ such that the residual graph has k disconnected components. The objective is to minimize the total number of removed edges. This is usually formalized by means of a cut. For any set of vertices $\mathcal{V}_1 \subset \mathcal{V}$, the boundary of $\mathcal{V}_1$ is defined as

$$\partial\mathcal{V}_1 = \left\{\{i,j\} \in \mathcal{E} : i \in \mathcal{V}_1, j \notin \mathcal{V}_1\right\} , \tag{2.10}$$

and the cut of $\mathcal{V}_1$ is simply given as $\text{Cut}(\mathcal{V}_1) = |\partial\mathcal{V}_1|$, *i.e*, the number of edges that need to be removed to partition the vertices of the graph into $\mathcal{V}_1$ and its complement, $\mathcal{V}_1^c$. In the case of a weighted graph $(\mathcal{V}, \mathcal{E}, w)$, one extends the definition of cut as $\text{Cut}(\mathcal{V}_1) = \sum_{e\in\partial\mathcal{V}_1} w(e)$.

A collection of sets $\mathcal{V}_1, \ldots, \mathcal{V}_k$ is said to be a partition of $\mathcal{V}$ if $\mathcal{V}_\ell \cap \mathcal{V}_r$ for all $\ell \neq r$, and $\mathcal{V} = \bigcup_{\ell=1}^{k} \mathcal{V}_\ell$. We will often refer to these subsets as clusters of vertices. The optimization problem associated with the above k-way graph partitioning problem is to find a partition $\mathcal{V}_1, \ldots, \mathcal{V}_k$ of $\mathcal{V}$ that solves

$$\underset{\mathcal{V}_1,\ldots,\mathcal{V}_k}{\text{minimize}} \ \frac{1}{2}\sum_{i=1}^{k} \text{Cut}(\mathcal{V}_i) \ , \tag{2.11}$$

where the above objective is equal to the total number of edges removed. Observe that (2.11) has a trivial solution, where one may assign all vertices to one of the clusters while the others can be empty. Thus, this formulation, which is also called the *min-cut* problem, does not provide useful partitions.

Several applications of graph partitioning, including VLSI design, often require the sets to be of comparable sizes. This leads to the balanced graph partitioning problem, where one imposes the constraint of $|\mathcal{V}_\ell| \leq \frac{(1+\epsilon)}{k}|\mathcal{V}|$ for all $\ell = 1, \ldots, k$. This ensures that the cluster sizes differ by at most $\epsilon\mathcal{V}$ vertices. Such an explicit constraint is not essential in typical machine learning applications, and hence, one introduces implicit constraints by defining the following objectives for the minimization problem (von Luxburg, 2007):

$$\text{R-Cut}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \frac{1}{2}\sum_{\ell=1}^{k} \frac{\text{Cut}(\mathcal{V}_\ell)}{|\mathcal{V}_\ell|} \ , \tag{2.12}$$

which is known as the *ratio cut* of a partition, or

$$\text{N-Cut}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \frac{1}{2}\sum_{\ell=1}^{k} \frac{\text{Cut}(\mathcal{V}_\ell)}{\text{Vol}(\mathcal{V}_\ell)} \ , \tag{2.13}$$

called the *normalized cut* of a partition. Here, the volume of a set of vertices quantifies the total connectivity of the set. It is defined as the sum of the degrees of the vertices in the set, *i.e*, $\text{Vol}(\mathcal{V}_\ell) = \sum_{i\in\mathcal{V}_\ell} \deg(i)$, where $\deg(i)$ is the total number (or weight) of edges connected to vertex i.

The quantities in (2.12) and (2.13) are closely related to the notions of *edge expansion* and *conductance* of a graph, which are often studied in theoretical computer science (Chung, 1997;

Arora et al., 2004; Peng et al., 2015), where one replaces the summation of the ratios by their maximum.

A problem that is equivalent to (2.13) was suggested by Shi and Malik (2000), and forms the theoretical basis for the spectral clustering algorithm of (Ng et al., 2002). Here, one reformulates the cut minimization as a maximization problem. For any set $\mathcal{V}_1 \subset \mathcal{V}$, define its associativity as $\text{Assoc}(\mathcal{V}_1) = \left|\left\{\{i,j\} \in \mathcal{E} : i,j, \in \mathcal{V}_1\right\}\right|$ for an unweighted graph, or $\text{Assoc}(\mathcal{V}_1) = \sum_{e \in \mathcal{E}: e \subset \mathcal{V}_1} w(e)$ in the case of a weighted graph. Note that the associativity of a set is inversely related to its cut since

$$\frac{\text{Assoc}(\mathcal{V}_1)}{\text{Vol}(\mathcal{V}_1)} + \frac{1}{2}\frac{\text{Cut}(\mathcal{V}_1)}{\text{Vol}(\mathcal{V}_1)} = \frac{1}{2} \, .$$

In view of the above relation, one may restate the problem of minimizing (2.13) as

$$\underset{\mathcal{V}_1,\ldots,\mathcal{V}_k}{\text{maximize}} \ \text{N-Assoc}(\mathcal{V}_1,\ldots,\mathcal{V}_k) = \sum_{\ell=1}^{k} \frac{\text{Assoc}(\mathcal{V}_\ell)}{\text{Vol}(\mathcal{V}_\ell)} \, . \tag{2.14}$$

The optimization problems presented in (2.12)–(2.14) are known to be NP-Hard. Hence, it is common to solves relaxations of these problems. In the next section, we focus on a spectral relaxation that leads to spectral partitioning algorithms. In this respect, we may also introduce a related quantity that is not explicitly defined in the literature. We call this the ratio associativity of a partition, defined as

$$\text{R-Assoc}(\mathcal{V}_1,\ldots,\mathcal{V}_k) = \sum_{\ell=1}^{k} \frac{\text{Assoc}(\mathcal{V}_\ell)}{|\mathcal{V}_\ell|} \, . \tag{2.15}$$

However, spectral relaxation of maximizing ratio associativity leads to a variant of spectral clustering that is often considered in the statistics community (Lei and Rinaldo, 2015).

2.2.2 Spectral relaxation of cut minimization and related problems

As pointed out in Chapter 1, the spectral connection of graph theory arises due to the representation of a graph in terms of its adjacency matrix, A. For a graph $(\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$, the matrix $A \in \mathbb{R}^{n \times n}$ is binary with $A_{ij} = 1$ if and only if $\{i,j\} \in \mathcal{E}$. For weighted graph $(\mathcal{V}, \mathcal{E}, w)$, we define $A_{ij} = w(\{i,j\})$. It turns out that spectral properties of A, or related matrices, provide insights into several properties of the corresponding graph, such as connectivity, colorability etc. (Spielman, 2011). In the context of graph partitioning, it is well known that spectral properties of a graph are closely related to the notions of normalized cut and graph

conductance through the discrete Cheeger inequality (Chung, 1997; Lee et al., 2012). More precisely, the optimal value of the objective function in (2.13) can be bounded in terms of the eigenvalues of the normalized graph Laplacian $L \in \mathbb{R}^{n\times n}$ defined as

$$L = I - D^{-1/2} A D^{-1/2} \,, \tag{2.16}$$

where A is the adjacency matrix of the graph (or weighted adjacency in case of weighted graphs), and $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $D_{ii} = \deg(i)$.

The primary concern of this thesis is to exploit the spectral properties of the adjacency matrix or the graph Laplacian to obtain a partition of $\mathcal{V}$, which is achieved by considering a spectral relaxation of the optimization problem based on the quantities defined in (2.12)-(2.15). A spectral relaxation of minimizing (2.13) is obtained as follows. For any partition $\mathcal{V}_1, \ldots, \mathcal{V}_k$, define a matrix $Y \in \mathbb{R}^{n\times k}$ such that

$$Y_{i\ell} = \sqrt{\frac{\deg(i)}{\mathrm{Vol}(\mathcal{V}_\ell)}} \mathbb{1}\{i \in \mathcal{V}_\ell\} \,. \tag{2.17}$$

It is easy to see that

$$\left(Y^T L Y\right)_{\ell\ell} = \frac{1}{\mathrm{Vol}(\mathcal{V}_\ell)} \left(\sum_{i\in\mathcal{V}_\ell} D_{ii} - \sum_{i,j\in\mathcal{V}_\ell} A_{ij} \right) = \frac{\mathrm{Cut}(\mathcal{V}_\ell)}{\mathrm{Vol}(\mathcal{V}_\ell)} \,,$$

which, in turn, implies that $\text{N-Cut}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \mathrm{Trace}\left(Y^T L Y\right)$. Thus, one can rewrite the normalized cut minimization problem as

$$\underset{\mathcal{V}_1,\ldots,\mathcal{V}_k}{\text{minimize}} \ \mathrm{Trace}\left(Y^T L Y\right) \,, \tag{2.18}$$

where Y has the form specified in (2.17). The above optimization is well known to be NP-hard, but one can observe that the matrix Y also has orthonormal columns, *i.e*, $Y^T Y = I$. This fact motivates one to relax the problem in (2.18) as

$$\underset{Y:\ Y^T Y = I}{\text{minimize}} \ \mathrm{Trace}\left(Y^T L Y\right) \,. \tag{2.19}$$

Standard results in matrix theory show that L is positive semi-definite, and the solution of (2.19) corresponds to the matrix of the leading k orthonormal eigenvectors of L. Hence, there exists a simple solution to the above relaxation of the N-Cut problem, which is referred to as a spectral relaxation due to its spectral connection. In a similar manner, one can relax the problems and

the solutions can be listed as in Table 2.1[1].

Table 2.1: Graph partitioning objectives and spectral relaxations.

Problem	Solution from spectral relaxation
Minimize N-Cut (2.13)	Leading k orthonormal eigenvectors of normalized graph Laplacian $L = I - D^{-1/2}AD^{-1/2}$
Maximize N-Assoc (2.14)	Dominant k orthonormal eigenvectors of normalized adjacency matrix $D^{-1/2}AD^{-1/2}$
Minimize R-Cut (2.12)	Leading k orthonormal eigenvectors of unnormalized graph Laplacian $L_{un} = D - A$
Maximize R-Assoc (2.15)	Dominant k orthonormal eigenvectors of adjacency matrix A

The spectral solutions mentioned in Table 2.1 motivates a spectral graph partitioning approach. Several variants of this method, listed below, are commonly known as the spectral clustering algorithm.

Algorithm Spectral Clustering : Finds a k-way partition of a graph

Input: Adjacency matrix A of a graph on vertex set $\mathcal{V}$ with $|\mathcal{V}| = n$.

Compute the matrix $X \in \mathbb{R}^{n \times k}$ whose columns are one of the following:

- Leading k orthonormal eigenvectors of $L = I - D^{-1/2}AD^{-1/2}$
- Dominant k orthonormal eigenvectors of $D^{-1/2}AD^{-1/2}$
- Leading k orthonormal eigenvectors of $L_{un} = D - A$
- Dominant k orthonormal eigenvectors of A

Normalize rows of X to have unit norm, and denote this matrix as $\overline{X}$.

Cluster the rows of $\overline{X}$ using a distance based clustering algorithm.

Output: Partition of $\mathcal{V}$ that corresponds to the cluster assignment of the rows of $\overline{X}$.

The last step of distance based clustering needs some discussion. If the solution of the spectral relaxation results in a matrix X that conforms with the structure given in (2.17), then one can see after row normalization, $\overline{X}$ corresponds to a binary matrix with exactly one non-zero term in each row. In other words, $\overline{X}$ corresponds to a cluster assignments matrix for the vertex set. Hence, one obtains the desired partition simply by inspecting the rows of $\overline{X}$. In general, X does not follow the nice structure in (2.17), but provided that it is close to such a representation, one can expect a pair of rows of $\overline{X}$ to be nearly identical if and only if the

[1] Please refer to Definition 2.1 for the terminology of leading and dominant eigenvectors.

corresponding vertices lie in the same cluster. Hence, one resorts to clustering the rows of $\overline{X}$ based on pairwise distances.

The popular choice for distance based clustering is the k-means algorithm (Lloyd, 1982), and hence, one typically uses this method in spectral clustering. Even for the extensions of spectral clustering to the case of hypergraphs that are studied in this thesis, we use k-means as the distance based clustering algorithm. However, theoretical guarantees of the classical k-means method is not convincing, and in fact, convergence of the algorithm is not guaranteed. Hence, for theoretical analysis, one relies on approximate versions of k-means (Kumar et al., 2004; Ostrovsky et al., 2012) or alternative clustering schemes (Gao et al., 2015) that have well studied guarantees. In the consistency results of this thesis, we assume that the distance based clustering is performed using the algorithm proposed by Ostrovsky et al. (2012).

2.3 Planted partition in graphs: Stochastic block model

The purpose of this thesis is to provide analytical bounds on the goodness of spectral methods for hypergraph partitioning. So, it is important to explain this notion before proceeding further.

2.3.1 Goodness of partitioning algorithms

Earlier we observed that partitioning the vertices of a graph is essentially a clustering problem. This implies that one may quantify the performance of a network partitioning algorithm using measures for clustering performance evaluation. This is followed in practice, and various metrics such as F1 score, normalized mutual information (NMI), Rand index or its adjusted form (ARI) etc. are used to measure the accuracy of partitioning algorithms. A more natural measure in terms of clustering error is commonly used in the statistics literature for studying performance of partitioning algorithms. The clustering error is measured as the number of disagreements between two sets of labels. Let $\psi : \{1, \ldots, n\} \to \{1, \ldots, k\}$ denote the true labeling function for a given a set of n vertices, *i.e*, ψ_i is the true label of vertex i. Let $\psi'_1, \ldots, \psi'_n$ be the labels obtained from a partitioning algorithm. We define the clustering error as

$$\text{Error}(\psi, \psi') = \min_{\sigma} \sum_{i=1}^{n} \mathbb{1}\{\psi_i \neq \sigma(\psi'_i)\} \ , \tag{2.20}$$

where $\mathbb{1}$ denotes the indicator for disagreements and the minimum is taken over all permutation maps σ on the set $\{1, \ldots, k\}$, *i.e*, all possible permutation of output labels. This allowance for permuting the output labels is crucial since a partitioning algorithm, or any unsupervised

method, does not distinguish among the labels. The quantity in (2.20) will be used later to formally answer Question 2 mentioned in the introduction.

It is quite tricky to theoretically analyze the performance of clustering or partitioning algorithms. Due to the absence of training data, any unsupervised learning approach is unaware of the true labels, and clusters are estimated by solving a variety of objectives that are not even closely related to (2.20). Hence, worst case bounds on (2.20) or related quantites are usually meaningless. One possible study, that plays fair to the algorithm, is to judge an algorithm based on the objective function optimized by the method. Guarantees are provided on the optimality of the solutions, usually under certain assumptions (Kannan et al., 2004).

A statistical treatment of the problem is to assume that the data or the network is generated from a random model, and then one derives bounds on (2.20) (Rohe et al., 2011). Such a study is capable of guaranteeing both the accuracy of the algorithm as well as the appropriateness of the underlying objective function. Both forms of analysis are prevalent in statistics and computer science, and mixture of these can also be found in several results (von Luxburg et al., 2008; Shi et al., 2009). The latter approach for theoretical analysis, which we describe in the next section, allows one to comment on both finite sample as well as asymptotic performance of the algorithm. In the asymptotic case, one can study notions of consistency similar to the case of supervised learning methods. The stochastic block model literature refers to two notions of consistency:

- A partitioning algorithm is *strongly consistent* if $\text{Error}(\psi, \psi') = o(1)$ with high probability, *i.e*, $\mathsf{P}(\text{Error}(\psi, \psi') = o(1)) \to 1$ as $n \to \infty$. This condition implies that for sufficiently large n, $\text{Error}(\psi, \psi') < 1$ and hence, the algorithm accurately clusters all vertices.

- A weaker notion of consistency is also studied in the literature, and is usually satisfied by most of the practical approaches (Rohe et al., 2011; Choi et al., 2012). A partitioning algorithm is *weakly consistent* if $\text{Error}(\psi, \psi') = o(n)$ with high probability.

The condition of weak consistency seems quite strange at the first glance since it allows an algorithm to incur a large error that can grow 'almost' linearly with the number of vertices. However, this notion is still useful from a practical perspective, where we are typically interested in percentage clustering error or $\frac{1}{n}\text{Error}(\psi, \psi')$. For weakly consistent algorithms, this quantity is $o(1)$ and hence, eventually vanishes. Refinement procedures are known in the literature (Vu, 2014; Gao et al., 2015) that can refine the partition obtained from a weakly consistent algorithm to provide a strongly consistent solution. This observation further emphasizes the importance of weakly consistent algorithms.

2.3.2 Stochastic block model and the related literature

Classical random graphs, generated from the Erdös-Rényi model, has identically distributed edges. It is known that if the edge probability is very small, then there are a large number of disconnected components, whereas for a large edge probability, one can observe a giant connected component in the graph (Erdös and Rényi, 1959). It is obvious that such a model cannot account for true partition in a graph.

The study of strongly connected communities in a network has always been of interest in the context of social networks. To model this phenomenon in a random graphs, sociologists considered the stochastic block model (Holland et al., 1983). Here, we describe a particular version of the model, known as the sparse stochastic block model (Lei and Rinaldo, 2015).

Consider a vertex set $\mathcal{V} = \{1, \ldots, n\}$, where the vertices are labeled using k class labels. Let $\psi : \{1, \ldots, n\} \to \{1, \ldots, k\}$ denote the labeling function. Assume that there is a scalar $\alpha \in (0, 1]$ that may vary with n, and a fixed symmetric matrix $B \in [0, 1]^{k \times k}$. The edge set $\mathcal{E}$ is constructed as follows. For every $i, j \in \mathcal{V}$, one considers the event $\big\{\{i, j\} \in \mathcal{E}\big\}$. These events are assumed to be mutually independent, and they occur with probability

$$\mathsf{P}(\{i, j\} \in \mathcal{E}) = \alpha B_{\psi_i \psi_j} \, .$$

Observe that presence of an edge (i, j) is governed by the class labels ψ_i and ψ_j. We do not allow the entries of B to vary with n. Hence, if $\alpha = 1$, every edge occurs with a fixed specified probability. This leads to formation of dense graphs where the expected number of edges has a quadratic dependence on the number of vertices. This behavior is not common in real world graphs, where the number of edges grow almost linearly with the number of vertices. To account for this factor, one set α as a decreasing function of n that controls the sparsity of a graph.

From the point of view of the adjacency matrix, the above model generates a binary symmetric matrix A with $\mathsf{P}(A_{ij} = 1) = \alpha B_{\psi_i \psi_j}$ for all $i \neq j$. It is quite interesting to observe the structure of the matrix $\mathcal{A} = \mathsf{E}[A]$, where the expectation is considered entry wise and over the probability measure of the random graph. One can see that $\mathcal{A}$, commonly called the *population* adjacency matrix, has a *block* structure since it is constant over each block, where ψ_i and ψ_j are fixed. Hence, this model is known as the stochastic block model. The use of this model, in its general form, for the analysis of graph partitioning algorithms was first considered by McSherry (2001), who preferred to call this model a *planted partition* model. This name is due to the fact that a partition is planted or hidden in the random graph, and generalizes previously studied graph problems related to planted cliques and planted colorings. A similar model termed as *k-partite random graphs* was earlier studied by Simonovits and Sós (1991).

The study of spectral clustering under this model is relatively recent (Rohe et al., 2011; Lei and Rinaldo, 2015), and lies amidst a vast literature that study the weak consistency of several algorithms under the stochastic block model. Some of the notable works are mentioned in Chapter 1. Specific instances of the model have been studied for a long time.

Planted k-coloring. In the graph vertex k-coloring problem, one aims to color the vertices of a graph using k colors such that there is no edge that connects two vertices of same color. Formally, the objective is to find a labeling ψ' such that $\psi'_i \neq \psi'_j$ for all $\{i, j\} \in \mathcal{E}$.

A planted coloring model is defined as follows. Given $\mathcal{V}$ and a true label function ψ, the matrix B is defined such that $B_{\psi_i \psi_j} = 0$ if $\psi_i = \psi_j$, otherwise it takes a fixed value $p \in (0, 1)$. Alon and Kahale (1997) considered the case $k = 3$[1] with classes of equal size. It was showed that if $\alpha \geq \frac{C}{n}$ for an absolute constant $C > 0$, then a spectral algorithm correctly colors all the vertices with probability $(1 - o(1))$. A converse statement was recently proved by Chen and Xu (2014), which states that there is a constant $C' > 0$, such that, if $\alpha \leq \frac{C'}{n}$, then no algorithm can provide a proper k-coloring with success probability close to 1.

Planted clique. Here, one considers a clique (fully connected component) planted in a random graph. Technically, the planted model has two classes, where one is of size $s < n$, and represents the planted clique. Thus, $\alpha = 1$ and $B \in [0, 1]^{2\times 2}$ is such that $B_{11} = 1$, and $\frac{1}{2}$ otherwise.

If $s \geq C\sqrt{n}$ for some constant $C > 0$, then a spectral trick can identify the planted clique (Alon et al., 1998). However, there are no known polynomial time algorithm till date that can find planted cliques of size $o(\sqrt{n})$, while a Markov Chain Monte Carlo method can find a clique of size nearly $\ln n$ is super polynomial time (Jerrum, 1992). We refer the reader to the papers by (Feldman et al., 2012; Raghavendra and Schramm, 2015) for developments related to the planted clique problem.

Planted bisection. This special case is the closest relative of the clustering or community detection framework, and hence, it is of considerable interest in statistics and machine learning. The model consists of $k = 2$ classes of equal size, and for given parameters $p, q \in [0, 1]$ such that $(p+q) \leq 1$, the matrix $B \in [0, 1]^{2\times 2}$ is defined as $B_{11} = B_{22} = (p+q)$, and $B_{12} = B_{22} = q$. This implies that edges across cluster boundaries occur with probability αq, while within cluster edges occur with a higher probability $\alpha(p+q)$. Here, the nature of the sparsity factor α mostly governs the difficulty of the problem, and bisection gets harder if the graph is sparser, *i.e*, when α decays rapidly with n.

The best known error bound for spectral clustering (Lei and Rinaldo, 2015) shows that if $\alpha \geq \frac{C \ln n}{n}$, then the algorithm makes $o(n)$ incorrect assignments, *i.e*, the method is weakly

[1] The case $k = 2$ corresponds to bipartite graphs, which can be easily 2-colored by breadth first search.

conistent. Additional refinement procedures (Vu, 2014; Lei and Zhu, 2014; Gao et al., 2015) can be used to exactly recover the partition with probability $(1 - o(1))$. Other methods are also known to exactly recover the partition under similar conditions (Amini and Levina, 2014). Owing to the impossibility results of Chen and Xu (2014), one can not achieve a high success probability for smaller growth rate of α.

However, it was empirically observed that even when $\alpha = \Omega(\frac{1}{n})$, the partition can be identified by a belief propagation algorithm (Decelle et al., 2011). Subsequently, Mossel et al. (2013a) proved that there is a sharp threshold $C_0 > 0$ such that if $\alpha < \frac{C_0}{n}$ then it is impossible to detect the partition with positive probability by any algorithm, whereas the belief propagation schemes succeeds with a constant probability when $\alpha > \frac{C_0}{n}$. Until recently, it was not known whether a spectral method works in such sparse cases. While the question is still open whether the spectral properties of graph adjacency or Laplacian is useful for partitioning in this regime, it has been shown that exact recovery is possible with a positive probability if one considers spectral decomposition of a regularized adjacency matrix (Le et al., 2015) or a non-backtracking matrix (Krzakala et al., 2013).

The above discussions are limited to studies on the standard sparse stochastic block model. Some recent studies have also incorporated practical aspects of networks such as degree heterogeneity (Karrer and Newman, 2011), or overlapping communities (Zhang et al., 2014). Consistency results for spectral clustering and alternative approaches have been studied under such modifications (Lei and Rinaldo, 2015; Zhang et al., 2014).

2.4 A review of hypergraph partitioning

We take this opportunity to mention different applied problems that have been modeled as hypergraph partitioning problems, and the variety of approaches that have proposed to solve the problem. The early research on hypergraph partitioning was restricted to circuit design applications, and a review of different methods used in the VLSI community can be found in (Alpert and Kahng, 1995; Karypis and Kumar, 2000). We only describe the principle behind these classical approaches. Later, Agarwal et al. (2006) reviewed some of the spectral algorithms based on hypergraph reduction. However, since these reviews, considerable research has been done on this problem in machine learning and computer vision. We categorize the algorithms based on the underlying applications, and briefly describe various techniques used for hypergraph partitioning.

2.4.1 Circuit partitioning

Consider a logic circuit that contains several logic gates, called modules, connected amongst themselves. The modules are viewed as vertices, and the connections, called nets, act as edges. Interestingly, a single net is often used to connect several modules, which leads to the hypergraph structure. A balanced partitioning objective arises in the following way. Typically, integrated circuits contain arbitrarily large number of modules, and may not fit on a single chip. Hence, it is desirable to divide the modules into smaller groups that are integrated on the same chip. The primary concerns include reducing the signal transmission across chips (minimizing cut), and allocating nearly equal number of modules to each chip (balanced partitioning). Partitioning of modules is also essential for a divide and conquer strategy in VLSI design. Several techniques have been used for circuit partitioning.

Move based approach. The classical Kernighan-Lin scheme (Kernighan and Lin, 1970), originally proposed for graphs, has been often used in practice for hypergraph partitioning. This is an iterative approach, where in each iteration, one sequentially scans the vertices, and assigns each vertex to a cluster such that the objective function is optimized.

The process gets quite cumbersome for very large number of vertices. This is typically tackled through a multi-level paradigm (Karypis and Kumar, 2000). Here, one coarsens the hypergraph by repeatedly merging strongly connected vertices in a hierarchical manner. Once, the number of vertices are significantly reduced, one can use an iterative scheme to obtain a partition. An uncoarsening phase follows, where the merged vertices are split and the partitioning is refined. This approach is known to achieve significant computational advantages.

Hypergraph reduction techniques. An alternative strategy is to reduce the hypergraph to a weighted or unweighted graph (Hadley, 1995). Though it is known that cut properties of a hypergraph cannot be always retained through such reductions (Ihler et al., 1993), yet this approach is popular since it allows one to rely on the extensive graph partitioning literature to solve the hypergraph partitioning problem.

Agarwal et al. (2006) classifies hypergraph reduction techniques into two principal approaches. In the *clique expansion* of a hypergraph, every edge e is replaced by $\binom{|e|}{2}$ pairwise edges, one for every pair vertices in e. Often these new edges assigned with some weights (Agarwal et al., 2005). The *star expansion* of a hypergraph involves addition of new vertices, one for every edge e. Subsequently, the edge is replaced $|e|$ pairwise edges, each connecting the vertex for e to every vertex $i \in e$.

Spectral algorithms. Typically, such algorithms involve a hypergraph reduction technique

followed by a spectral graph partitioning algorithm. To this end, it is known that the spectral properties of both clique and star expansions are closely related (Agarwal et al., 2006).

There are several alternative circuit partitioning approaches (Alpert and Kahng, 1995), but the hypergraph reduction strategy combined with spectral techniques undoubtedly has the widest reach over several domains. The non-uniform hypergraph partitioning algorithms discussed in Chapter 5 can be described as clique or star expansion based techniques.

2.4.2 Categorical data clustering and attribute clustering

This is one of the less popular hypergraph partitioning applications. We describe the problem through an example. Table 2.2 lists few animals and their characteristics. For the purpose clustering the animals based on their characteristics, one may consider a following hypergraph problem. Let the animals correspond to the vertices, and a feature value (for example, "cannot swim") corresponds to an edge among all animals with the particular feature value.

Table 2.2: Example of categorical dataset.

	Characteristics			
Animal	Domestic	Swims	Agility	Size
Cat	Yes	No	Medium	Small
Elephant	No	No	Slow	Enormous
Whale	No	Yes	Slow	Enormous
Horse	Yes	No	Fast	Medium
Piranha	No	Yes	Fast	Small
⋮				

A related problem of attribute clustering has also been studied, where the purpose is to initially cluster the attributes based on their co-occurrence. Subsequently, these clusters can be used to group the rows of the database based on their coincidence with each cluster. In case of attribute clustering, the attribute value acts as vertices and every row corresponds to an edge among these values. Note that this leads to a uniform hypergraph. For instance, Table 2.2 corresponds to a 4-uniform hypergraph among the attribute values: 'domestic', 'wild', 'can swim', 'fast', 'slow' etc.

Extension of such a formulation to co-authorship networks or related networks has also been suggested in the literature. Few hypergraph partitioning approaches have been been studied for such clustering problems. These include multi-level schemes (Han et al., 1997), iterative approaches based on evolution of dynamical systems (Gibson et al., 2000).

2.4.3 Subspace clustering and geometric grouping

The study of the subspace clustering problem has led to an elevated interest in hypergraph partitioning in machine learning. The problem is formulated in the following way.

Consider a collection of n points $Y_1, Y_2, \ldots, Y_n \in \mathbb{R}^{r_a}$ in an high dimensional ambient space. Assume that there exist k subspaces, each of dimension at most $r < r_a$, such that one can represent Y_i as

$$Y_i = \widetilde{Y}_i + \eta_i \ ,$$

where $\widetilde{Y}_i$ lies in one of the k subspaces, and η_i is a noise term. The objective of a subspace clustering algorithm is to group $Y_1, \ldots, Y_n$ into k disjoint clusters such that each cluster corresponds to exactly one of the k low-dimensional subspaces.

A hypergraph based approach for the subspace clustering problem (Agarwal et al., 2005) involves construction of a weighted m-uniform hypergraph such that $m \geq (r+2)$ and the weight of an edge $e = \{i_1, \ldots, i_m\}$ is given by

$$w(e) = w(\{i_1, \ldots, i_m\}) = \exp\left(-\frac{f_r(Y_{i_1}, \ldots, Y_{i_m})}{\sigma^2}\right) \ . \tag{2.21}$$

Here, $f_r(\cdot)$ computes the error of fitting a r-dimensional subspace for the given m points, and σ is a scaling parameter. The reason for considering $m \geq (r+2)$ is because in the absence of noise, any $(r+1)$ points fit a r-dimensional subspace. However, when $m = (r+2)$ or higher, the fitting error $f_r(\cdot)$ is zero only if the points belong to the same subspace, *i.e*, they belong to the same cluster. As a consequence, the edge weight $w(e)$ is high for vertices in the same cluster, otherwise it can be considerably small.

Different choices for $f_r(\cdot)$ has been considered in the literature based on Euclidean distance of points from the estimated subspace (Jain and Govindu, 2013), polar curvature of the points (Chen and Lerman, 2009) among others.

The problem has an immediate extension to applications, where the underlying clusters may not correspond to subspaces but can be general manifolds. This problem is referred to as geometric grouping (Govindu, 2005). The hypergraph formulation extends identically, but the choice of $f_r(\cdot)$ depends on the underlying geometric structure and plays a crucial role in the appropriateness of a hypergraph partitioning approach.

The subspace clustering literature is quite broad, and several solution techniques have been studied. We mention some of the uniform hypergraph partitioning techniques that have been proposed for this problem.

Spectral algorithms. Agarwal et al. (2005) first observed that one could formulate the

subspace clustering problem in terms of hypergraph partitioning. The authors proposed an algorithm that performs spectral partitioning of the clique expansion of a hypergraph. Later, spectral techniques have been used in other approaches that compute a certain matrix from the weighted adjacencies of the hypergraph (Arias-Castro et al., 2011).

Tensor decomposition based methods. The subspace clustering problem is modeled in terms of a uniform hypergraph, which in turn provides a tensorial flavor to the problem. This class of algorithms attempts to exploit the spectral properties of adjacency tensor to obtain the partition. In this respect, one may call these methods as uniform hypergraph generalization of spectral clustering.

Govindu (2005) proposed to use the orthonormal matrix obtained from higher order singular value decomposition (see Defintion 2.6) as a generalization of the eigenvector matrix used in spectral clustering. On the other hand, Shashua et al. (2006) suggested an approximation of the normalized adjacency tensor by a rank k CP-decomposition as defined in Definition 2.5. In Chapter 3, we analyze the former method in considerable detail, whereas Chapter 4 shows that the latter method is a special case of a general partitioning framework.

More refined and efficient variants of these tensor based methods have been studied (Chen and Lerman, 2009; Jain and Govindu, 2013). A major challenge addressed in the literature is reduction of the computational complexity of tensor based methods. We address this aspect in Chapter 6.

Optimization techniques. As in the case of graphs, the hypergraph partitioning problem essentially involves the optimization of a certain objective function under certain constraints. Both hypergraph reduction as well as tensor decomposition methods solve a spectral relaxation of a certain optimization problem. One may alternatively directly solve the optimization problem. A wide variety of objective functions have been proposed, where the ℓ_2-norm constraint of spectral methods is often replaced by ℓ_1-norm constraints. Rota Bulo and Pelillo (2013) formulate the partitioning problem as an evolutionary game, and the resulting optimization is similar to the tensor ℓ_1-eigenvalue problem in (2.8). Improvements of this formulation have been suggested that impose additional constraints (Liu et al., 2010).

2.4.4 Hypergraph coloring

We now deviate from clustering applications based on hypergraphs, and discuss other instances of hypergraph partitioning. Here, we discuss extensions of the graph coloring problem to the case of hypergraphs. Hypergraph coloring has received considerable attention in theoretical studies (e.g., Achlioptas and Coja-Oghlan, 2008). Hypergraph coloring algorithms have also

been used in several applications such as DNF counting, resource allocation, scheduling etc. (Lu, 2004; Capitanio et al., 1995; Ahuja and Srivastava, 2002).

Recall that in graph coloring, the two vertices in every edge needs to be different colors. This problem can be generalized in several ways. In the *strong coloring* problem, one needs to color every vertex in an edge with a different color, *i.e*, no edge is allowed to have repeated colors. One can observe that a strong coloring problem can be reduced to the coloring problem for the clique expansion of the graph. Hence, this is typically solved using graph coloring techniques.

A more interesting problem is that of *weak coloring*, where one needs to only ensure that no edge is monochromatic, *i.e*, every edge consists of vertices from at least two color classes. Unlike the case of graphs, the presence of larger edges makes the weak coloring problem quite tricky even for two colors. Hardness results related to this problem are well known (Khot and Saket, 2014). Several combinatorial and spectral algorithms for graph coloring have been extended to solve this problem (Alon et al., 1996; Chen and Frieze, 1996).

Intermediate problems that lie between the two extreme cases of strong and weak coloring have also been studied in the hypergraph literature (Schmidt, 1987).

2.4.5 Hypergraph matching

Here, the problem of interest is that of finding one-one correspondences between two collection of points. We describe a simple version of it below. Consider two sets of points, each containing s points. Each collection typically corresponds to the features of interest in an image, and a solution to the matching problem finds correspondences between two images. One can see that there are s^2 candidate matches, out of which only s matches are correct.

Based on the theory of computer vision, one can conclude that if one image is a transformation of the other, then certain properties are preserved for correct pairings. For instance, let $\{1, \ldots, s\}$ and $\{1', \ldots, s'\}$ denote the two collections of points with i corresponding to i'. If the image is merely rotated, one can claim that $\|i - j\|_2 = \|i' - j'\|_2$. Even under more complex transformations, there are functions computed on three or four points (for instance, sine of angles formed or ratio of areas of triangles) that are preserved between the two images.

A tensor or hypergraph based formulation is often used for this problem. Let the vertices of be all the candidate matches, *i.e*, $\mathcal{V} = \{(i, j') : i = 1, \ldots, s; j' = 1', \ldots, s'\}$. If $g_m(\cdot)$ is a function of m points that is preserved under the transformation, then one constructs a m-uniform hypergraph with edge weights given by

$$w(e) = w(\{(i_1, j'_1), \ldots, (i_m, j'_m\}) = \exp\left(-\frac{|g_m(i_1, \ldots, i_m) - g_m(j'_1, \ldots, j'_m)|^2}{\sigma^2}\right), \tag{2.22}$$

where σ is an appropriately chosen value. One can note that the edge is close to one for correct matches of all m vertices, otherwise it is small.

Tensor eigenvalue problems. The problem is often solved by means of the adjacency tensor of the uniform hypergraph. Duchenne et al. (2011) defined a score function for the correspondences that is quite similar to the multilinear functional defined in (2.6). As a consequence, the associated optimization is identical to the variational form of the eigenvalue problem (2.8), and is solved via tensor power iterations. An alternative approach of Chertok and Keller (2010) formulates the correct matching as a solution of the higher order singular value decomposition stated in Definition 2.6.

Optimization techniques. The problem can be approached more directly by solving the associated optimization by numerical techniques (Liu et al., 2010). Other related methods based on tensor power iterations (Nguyen et al., 2015) and random walks on hypergraphs (Lee et al., 2011) are known to provide accurate solutions.

The above discussion follows the lines of the existing literature, where one formulates the problem as a hypergraph, but no direct connection is made with hypergraph partitioning. However, there exists an intrinsic relation that is revealed from a closer look at (2.22). The edge weights suggest the s vertices corresponding to the correct matches are strongly connected, and closely resembles a 'clique'. Thus, the hypergraph matching is similar, in spirit, to the planted clique problem.

2.5 Few important results

We conclude this chapter with few standard results that will be repeatedly used in the subsequent chapters of the thesis.

2.5.1 Matrix perturbation results

Matrix perturbation theory studies the variation of eigenvalues and eigenvectors of a matrix when the entries of the matrix are perturbed. The following results provide bounds on the deviation of the eigenvalues and eigenvectors under additive perturbation.

Theorem 2.7 (Weyl's inequality (Weyl, 1912))**.** *Let $\mathcal{K} \in \mathbb{R}^{n \times n}$ be a symmetric matrix, and K be an additive perturbation of $\mathcal{K}$, i.e,*

$$K = \mathcal{K} + H$$

for a symmetric matrix $H \in \mathbb{R}^{n\times n}$. Let the eigenvalues of $\mathcal{K}$ be $\lambda_1 \geq \ldots \geq \lambda_n$, the eigenvalues of K be $\nu_1 \geq \ldots \geq \nu_n$, and those for H be $\rho_1 \geq \ldots \geq \rho_n$. Then for each $i = 1, \ldots, n$,

$$\lambda_i + \rho_n \leq \nu_i \leq \lambda_i + \rho_1 \ .$$

As a consequence, $|\nu_i - \lambda_i| \leq \max\{|\rho_1|, |\rho_n|\} = \|H\|_2$.

A similar result also exists for asymmetric matrices, where the above inequalities hold for singular values. This is known as Mirsky's theorem (Stewart and Sun, 1990). The next theorem deals with the deviation of eigenvectors.

Theorem 2.8 ($\sin\Theta$ theorem (Davis and Kahan, 1970))**.** *Let $\mathcal{K} \in \mathbb{R}^{n\times n}$ be a symmetric matrix, and K be an additive perturbation of $\mathcal{K}$. Let $S \subset \mathbb{R}$ be any interval that contains exactly k eigenvalues of $\mathcal{K}$. Define*

$$\delta_0 = \min\{|\lambda - \lambda'| : \lambda \in S, \lambda' \notin S, \text{ and } \lambda, \lambda' \text{ are eigenvalues of } \mathcal{K}\}.$$

If $\delta_0 > 2\|K - \mathcal{K}\|_2$, then S also contains exactly k eigenvalues of K.

Let $X, \mathcal{X} \in \mathbb{R}^{n\times k}$ be orthonormal eigenvector matrices for the eigenvalues in S of $K, \mathcal{K}$ respectively. Then

$$\|\sin\Theta(X, \mathcal{X})\|_2 \leq \frac{\|K - \mathcal{K}\|_2}{\delta_0} \ ,$$

where $\sin\Theta(X, \mathcal{X}) \in \mathbb{R}^{k\times k}$ is diagonal with entries same as the sine of the canonical angles between the subspaces X and $\mathcal{X}$.

The first statement is a consequence of Weyl's inequality, and the second claim deals with deviation of the subspaces spanned by $X, \mathcal{X}$. The following corollary to Theorem 2.8 is more useful in our context. This result was proved in (Lei and Rinaldo, 2015). We provide a simpler proof.

Corollary 2.9. *Consider the quantities defined in Theorem 2.8. If $\delta_0 > 2\|K - \mathcal{K}\|_2$, then there is an orthonormal matrix $U \in \mathbb{R}^{k\times k}$ such that*

$$\|X - \mathcal{X}U\|_F \leq \frac{2\sqrt{2k}\|K - \mathcal{K}\|_2}{\delta_0} \ .$$

Proof. Let the angles in $\sin\Theta(X, \mathcal{X})$ be denoted by $\theta_1, \ldots, \theta_k \in [0, \frac{\pi}{2}]$, where $\theta_1 \geq \ldots \geq \theta_k$. Then $\|\sin\Theta(X, \mathcal{X})\|_2 = \sin\theta_1$. On the other hand, one can see that the singular values for the matrix $X^T\mathcal{X}$ are given by $\cos\theta_1, \ldots \cos\theta_k$. Thus, if $X^T\mathcal{X} = U_1\Sigma U_2^T$ is the singular value

decomposition of $X^T\mathcal{X}$, then

$$\begin{aligned}\|X - \mathcal{X}U_2U_1^T\|_F^2 &= \text{Trace}\left((X - \mathcal{X}U_2U_1^T)^T(X - \mathcal{X}U_2U_1^T)\right) \\ &= 2\text{Trace}(I - U_1\Sigma U_1^T) \\ &= 2\sum_{i=1}^{k}(1 - \cos\theta_i) \leq 2\sum_{i=1}^{k}(1 - \cos^2\theta_i) \leq 2k\sin^2\theta_1 \ .\end{aligned}$$

Hence, we can conclude that for $\delta_0 > 2\|K - \mathcal{K}\|_2$,

$$\|X - \mathcal{X}U\|_F \leq \sqrt{2k}\frac{\|K - \mathcal{K}\|_2}{\delta_0} \ , \tag{2.23}$$

where $U = U_1U_2^T$. □

2.5.2 Concentration inequalities

Concentration inequalities play a vital role in various branches of probability and statistics, and have been central to the development of the theory of learning algorithms. There is a vast literature on concentration bounds, but in this thesis, we will use only two results, which deal with sums of random variables or matrices.

Theorem 2.10 (Bernstein inequality)**.** *Let $Y_1, Y_2, \ldots, Y_N$ be N real-valued independent random variables with finite second moments and $|Y_i - \mathsf{E}[Y_i]| \leq R$ almost surely for all i. If $Y = \sum_{i=1}^{N} Y_i$, then for all $t > 0$,*

$$\mathsf{P}\left(|Y - \mathsf{E}[Y]| \geq t\right) \leq 2\exp\left(\frac{-t^2}{2\mathsf{Var}(Y) + \frac{2}{3}Rt}\right).$$

Recently, it was observed that the above result can also be extended to study concentration of random matrices (Tropp, 2012; Chung and Radcliffe, 2011).

Theorem 2.11 (Matrix Bernstein inequality)**.** *Consider a finite sequence of independent, random, symmetric matrices $Y_1, Y_2, \ldots, Y_N \in \mathbb{R}^{n\times n}$. Assume that each random matrix satisfies $\|Y_i - \mathsf{E}[Y_i]\|_2 \leq R$ almost surely. Define $Y = \sum_{i=1}^{N} Y_i$, and let $\mathsf{Var}(Y) = \mathsf{E}\left[(Y - \mathsf{E}[Y])^2\right]$, where we assume all the above expectations exist. Then for all $t > 0$,*

$$\mathsf{P}\left(\|Y - \mathsf{E}[Y]\|_2 \geq t\right) \leq 2n\exp\left(\frac{-t^2}{2\mathsf{Var}(Y) + \frac{2}{3}Rt}\right).$$

2.5.3 A performance guarantee for k-means algorithm

Let $Y \in \mathbb{R}^{n\times r}$ be a data matrix, where each row corresponds to a r-dimensional data instance. The formal problem associated with k-means is as follows:

$$\underset{S\in\mathcal{M}_{n\times r}(k)}{\text{minimize}} \|Y - S\|_F \ , \tag{2.24}$$

where $\mathcal{M}_{n\times r}(k)$ is the set of all $n \times r$ matrices with at most k distinct rows. In practice, the rows of S correspond to the centers of the obtained clusters. Achieving a global optimum for this problem is NP-hard. However, there are algorithms (Kumar et al., 2004; Ostrovsky et al., 2012) that can provide a solution S^* from the above class of matrices such that

$$\|Y - S^*\|_F \leq \gamma \min_{S\in\mathcal{M}_{n\times r}(k)} \|Y - S\|_F \tag{2.25}$$

for some $\gamma > 1$. The factor γ depends on the algorithm under consideration. For instance, γ grows with k in the case of (Kumar et al., 2004), while Ostrovsky et al. (2012) showed that a constant factor approximation is possible if the data (rows of Y in our case) is *well-separated*.

To be precise, define $\eta_k(Y)$ to be the minimum of the objective function when k clusters are found, *i.e*,

$$\eta_k(Y) = \min_{S\in\mathcal{M}_{n\times r}(k)} \|Y - S\|_F \ . \tag{2.26}$$

The rows of Y is said to be ϵ-separated if $\eta_k(Y) \leq \epsilon\eta_{k-1}(Y)$. We state here Theorem 4.15 in (Ostrovsky et al., 2012), which provides a performance guarantee for an approximate k-means algorithm.

Theorem 2.12. *Assume that $\eta_k(Y) \leq \epsilon\eta_{k-1}(Y)$, where $\epsilon \leq 0.015$. Then the k-means algorithm of Ostrovsky et al. (2012) returns a solution S^* such that*

$$\|Y - S^*\|_F \leq \gamma\eta_k(Y)$$

with probability $(1 - O(\sqrt{\epsilon}))$ in time $O(nrk + rk^3)$. Here, $\gamma = \sqrt{\dfrac{1-\epsilon^2}{1-37\epsilon^2}}$.

We note that the above result holds for slightly modified variant of k-means, which runs in $O(nkr + k^3r)$ time. We provide an informal description of the algorithm here, and refer the interested reader to (Ostrovsky et al., 2012) for the exact procedure. Given data points $y_1, \ldots, y_n \in \mathbb{R}^d$, one performs the following steps:

1. Sample $O(k)$ points sequentially to define the initial centers. Let any step, the sampled

points be $\hat{s}_1, \ldots, \hat{s}_j$, then the next point is sampled with probability proportional to its minimum distance from $\hat{s}_1, \ldots, \hat{s}_j$. This step chooses $O(k)$ random centers that reasonably far apart.

2. Compute the clustering cost (objective in (2.24)) when all the $O(k)$ clusters are used, and then recursively remove centers from the sampled set such that increase in the clustering cost is minimum. These recursions are carried out this there are only k centers.

3. Let $\hat{s}_1, \ldots, \hat{s}_k$ be the surviving k centers. For each center $\hat{s}_i$, compute its distance $\hat{d}_i$ to the nearest of the other $(k-1)$ centers. Compute the centroid $\bar{s}_i$ of all points in a ball of radius $\hat{d}_i/3$ centered at $\hat{s}_i$.

4. Return the clustering that corresponding to the Voronoi diagram with the k points $\bar{s}_1, \ldots, \bar{s}_k$.

So many things are possible just as long as you don't know they're impossible.

Norton Juster, *The Phantom Tollbooth*

Chapter 3

A Tensor Spectral Method for Uniform Hypergraphs

The moment has arrived to embark on our mission of analyzing spectral methods for hypergraph partitioning. We simplify the matters at hand by considering uniform hypergraphs, and study one of the earliest works on higher order learning.

Through this chapter, we aim to establish the need for a planted partition model for hypergraphs. This task will certainly be incomplete without a discussion on the nature of existing studies on hypergraphs. Hence, after describing the algorithm in Section 3.1, we present a perturbation based analysis in Section 3.2 that extends the techniques of Ng et al. (2002) to hypergraphs. Subsequently, we discuss the limitations of this analysis, and present our analysis based on a planted partition model. Section 3.3 describes the model, and Section 3.4 provides an analysis of the algorithm under this model. In particular, we state and prove a consistency result. The technical lemmas used here are proved in the appendix to this chapter, Appendix 3.A.

3.1 Tensor decomposition and partitioning

Let $(\mathcal{V}, \mathcal{E}, w)$ be a weighted m-uniform hypergraph on $|\mathcal{V}| = n$ vertices, *i.e*, every edge in $\mathcal{E}$ is of size m. Recall that the edge weights of this hypergraph can be expressed in terms of a symmetric tensor $\mathbf{A}$ of order m and dimension n. The algorithm studied in this chapter is based on the the higher order singular value decomposition of $\mathbf{A}$ (see Definition 2.6), and hence, we prefer to refer to this method as Algorithm HOSVD.

This method was proposed by Govindu (2005) as a multi-way extension of spectral cluster-

ing, which relies on the following idea. Spectral clustering embeds the vertices of a graph in an Euclidean space by means of the dominant eigenvectors of the adjacency matrix, or its normalized equivalent. This approach can be replicated in uniform hypergraphs by using columns of the orthonormal matrices obtained from higher order SVD of the adjacency tensor $\mathbf{A}$. More precisely, one would be interested in the dominant eigen pairs, and in this respect, Definition 2.6 suggests the use of the dominant left singular vectors of the flattened matrix (2.4) for $\mathbf{A}$, which we denote by $\widetilde{\mathbf{A}} \in \mathbb{R}^{n \times n^{m-1}}$.

Note here that it is not clear how this idea can be extended to normalized adjacencies. Govindu (2005) proposed to use the following heuristic, which is listed in Algorithm HOSVD. One defines a symmetric matrix $A \in \mathbb{R}^{n \times n}$ as $A = \widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^T$, and normalizes A following the degree normalization in spectral clustering. Subsequently the dominant eigenvectors of the normalized matrix are computed, and used for distance based clustering. The construction of $A = \widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^T$ is natural for transforming the SVD into an eigen decomposition, but the matrix normalization step has not been justified in the literature.

Algorithm HOSVD : Partitioning via higher order SVD

Input: Affinity tensor $\mathbf{A}$ of the m-uniform hypergraph $(\mathcal{V}, \mathcal{E}, w)$.

1: Let $\widetilde{\mathbf{A}}$ be flattened matrix of $\mathbf{A}$, and $D \in \mathbb{R}^{n \times n}$ diagonal with $D_{ii} = \sum_{j=1}^{n} \sum_{\ell=1}^{n^{m-1}} \widetilde{\mathbf{A}}_{i\ell}\widetilde{\mathbf{A}}_{j\ell}$.

2: Compute k dominant orthonormal eigenvectors of $D^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^T D^{-1/2}$, denoted by $X \in \mathbb{R}^{n \times k}$.

3: Normalize rows of X to have unit norm, and denote this matrix as $\overline{X}$.

4: Run k-means on the rows of $\overline{X}$.

Output: Partition of $\mathcal{V}$ that corresponds to the clusters obtained from k-means.

The subsequent steps of the algorithm are similar to spectral clustering. Hence, one can observe that HOSVD is equivalent to reducing the hypergraph to a graph with weighted adjacency matrix A, and performing normalized spectral clustering on this graph. Variants of the above algorithm, in particular spectral curvature clustering (Chen and Lerman, 2009), are often used in practice. We postpone the discussion on such variants to Chapter 6.

The performance of HOSVD and its variants in practice has been established in the literature (Govindu, 2005; Chen and Lerman, 2009; Chen and Lerman, 2009; Jain and Govindu, 2013; Ghoshdastidar and Dukkipati, 2014). One can also refer to the qualitative results in Figure 8.1, as well as other numerical studies in Chapter 8. This chapter deals with the theoretical guarantees of HOSVD, studied from different perspectives. In particular, if ψ and ψ', respectively, are the true labeling function and the output labeling from HOSVD, then we derive upper bounds on the clustering error defined in (2.20). We denote this error by $\text{Error}_{\text{HOSVD}}(\psi, \psi')$.

3.2 A perturbation based analysis

In this section, we consider an analysis that is along the lines of the results in (Ng et al., 2002). More precisely, we extend the deterministic analysis of spectral clustering to HOSVD, and then discuss the limitations of such a study. The analysis described here is partly based on the works of Chen and Lerman (2009), who studied HOSVD under the name of theoretical spectral curvature clustering.

The principal approach of this analysis is to study the method in an ideal case, where HOSVD provides a perfect clustering. This observation is extended to derive error bounds for more general hypergraphs that satisfy certain conditions.

Consider a hypergraph $(\mathcal{V}, \mathcal{E})$ on $|\mathcal{V}| = n$ vertices, and let $\psi_1, \ldots, \psi_n \in \{1, \ldots, k\}$ denote the labels of the vertices corresponding to the desired partition. Let $n_1, \ldots, n_k$ be the cluster size, and without loss of generality, we may assume $n_1 \geq \ldots \geq n_k$. We define an ideal hypergraph corresponding to ψ as follows.

Definition 3.1. For a set of vertices $\mathcal{V}$ with labels $\psi_1, \ldots, \psi_n$, and ideal m-uniform hypergraph $(\mathcal{V}, \widetilde{\mathcal{E}})$ is such that for any set of m vertices $\{i_1, \ldots, i_m\} \subset \mathcal{V}$, there is an edge $e = \{i_1, \ldots, i_m\} \in \widetilde{\mathcal{E}}$ if and only if $\psi_{i_1} = \ldots = \psi_{i_m}$.

In other words, $(\mathcal{V}, \widetilde{\mathcal{E}})$ consists of k disjoint components, each being a complete m-uniform hypergraph on the vertices belonging to a particular class.

The first observation, adapted from (Chen and Lerman, 2009, Proposition 4.1), studies the performance of HOSVD on an ideal hypergraph. Let $\widetilde{K} = D^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^T D^{-1/2}$ be the matrix computed in HOSVD when the input is an ideal hypergraph, and let $\widetilde{X}$ denote the matrix of its k dominant eigenvectors

Lemma 3.2. *The largest eigenvalue of $\widetilde{K}$ is one, with multiplicity k, and the other eigenvalues lie in the interval $\left[\frac{m-1}{(n_1-1)(n_1+1-m)}, \frac{m-1}{(n_k-1)(n_k+1-m)}\right]$.*

If $Z \in \{0,1\}^{n \times k}$ is the assignment matrix corresponding to the label ψ, i.e, $Z_{i\psi_i} = 1$, and zero otherwise, then the eigenvector matrix $\widetilde{X}$ is given by $\widetilde{X} = (Z^T Z)^{-1/2} Z$, ignoring rotation of the matrices.

Observe that the matrix $(Z^T Z)^{-1/2} Z$ has exactly k distinct rows that are orthogonal to each other, and each distinct row corresponds to a particular cluster. Hence, any distance based clustering algorithm can identify ψ from the rows of the row normalized version of $\widetilde{X}$. However, recall that the actual input to HOSVD is the hypergraph $(\mathcal{V}, \mathcal{E})$ instead of the ideal one $(\mathcal{V}, \widetilde{\mathcal{E}})$. Let $K = D^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^T D^{-1/2}$ denote the matrix computed in HOSVD for the hypergraph

$(\mathcal{V}, \mathcal{E})$. Hence, one needs to identify the clusters from the rows of $\overline{X}$ that corresponds to the row normalized dominant eigenvector matrix for K. To this end, the following consequence of Corollary 2.9 is useful.

Lemma 3.3. *If the given hypergraph $(\mathcal{V}, \mathcal{E})$ satisfies*

$$\|K - \widetilde{K}\|_2 \leq \frac{n_k(n_k - m)}{2(n_k - 1)(n_k + 1 - m)} , \tag{3.1}$$

then there is an orthonormal matrix $Q \in \mathbb{R}^{k\times k}$ such that

$$\|\overline{X} - ZQ\|_F \leq \sqrt{8kn_1}\frac{(n_k - 1)(n_k + 1 - m)}{n_k(n_k - m)}\|K - \widetilde{K}\|_2 . \tag{3.2}$$

The above result suggests that the rows of $\overline{X}$ are quite close to the rows of Z after an appropriate rotation. Hence, one can expect a good recovery of the partition from the rows of $\overline{X}$. Standard tricks for analyzing the solution of k-means (Rohe et al., 2011) will be discussed later in the chapter, which can be used to argue that if the global optimum of k-means is achieved then the k-means error is bounded from above by $2\|\overline{X} - ZQ\|_F^2$. Thus, we can state the following result for HOSVD.

Theorem 3.4. *Let a m-uniform $(\mathcal{V}, \mathcal{E})$ be partitioned using HOSVD. Let the true cluster sizes be $n_1 \geq \ldots \geq n_k$, and assume that the global optimum of k-means can be achieved.*

There exists $\zeta > 0$, such that, if $n_k > m$ and

$$\zeta < \frac{n_k(n_k - m)}{2(n_k - 1)(n_k + 1 - m)} , \tag{3.3}$$

then clustering error of HOSVD is bounded as

$$\text{Error}_{\text{HOSVD}}(\psi, \psi') \leq 64kn_1\left(\frac{\zeta(n_k - 1)(n_k + 1 - m)}{n_k(n_k - m)}\right)^2 = O(kn_1\zeta^2) . \tag{3.4}$$

Proof of Lemma 3.2 can be found in (Chen and Lerman, 2009). We do not provide proofs for Lemma 3.3 or Theorem 3.4 since these can be derived from the arguments used in the proof of Theorem 3.7, presented later in this chapter. We now discuss the implications and limitations of Theorem 3.4.

Note that the quantity ζ corresponds to $\|K - \widetilde{K}\|_2$, and hence, quantifies the distance of the given hypergraph from an ideal hypergraph with the desired partition. As is expected, a better error rate is guaranteed for smaller ζ, but for large ζ, which violates (3.3), the error bound does

not hold. To this end, observe that even when (3.3) is marginally satisfied, the error bound is not useful, and hence, Theorem 3.4 makes sense only in regime $\zeta = O(\frac{1}{n})$. The above result can be extended to weighted hypergraphs as well, where a similar definition of ideal hypergraph can be stated.

3.2.1 Limitations of Theorem 3.4

Perhaps the limitations of the present analysis is evident from Theorem 3.4. We still elaborate of the various factors. To this end, we note that the main result in (Chen and Lerman, 2009) also has a similar flavor, but the resulting bound is on $\|\overline{X} - ZQ\|_F$ instead of $\text{Error}_{\text{HOSVD}}(\psi, \psi')$, and hence, less useful from a practical perspective.

The major limitation of Theorem 3.4 arises from the limited allowable hypergraphs that can be analyzed using this approach. The condition on ζ implies that the result is applicable only for hypergraphs that are close to a "ideal hypergraph" with k disjoint components. This assumption is quite strict, and cannot be expected to be true in general. Interestingly, the above analysis overlooks the fact that HOSVD is able to perfectly cluster certain hypergraphs that need not be ideal. Examples of such hypergraphs will be evident from the discussions in the next section.

Furthermore, in Theorem 3.4, we assume that the global optimum of k-means is achieved. This assumption does not hold in practice. While approximate k-means strategies are known that provide near optimal solutions, their success probability is less than one. More importantly, the optimality and running time of such approaches depend critically on the number of classes k, and require careful considerations. In the next section, we analyze HOSVD when the k-means step is performed using the algorithm of Ostrovsky et al. (2012), and thereby avoid any assumption on the performance of k-means. A more detailed discussion on the effect of various assumptions on the k-means step can be found in Chapter 5.

Arias-Castro et al. (2011) analyzed a different tensor based approach for the problem of geometric grouping, where a weighted uniform hypergraph is constructed from multi-way similarities among data instances that are generated from an union of manifolds. As discussed in Section 2.4.3, the problem gets more interesting when the given data is perturbed by random noise. Due to the inherent randomness of the model, the analysis of Arias-Castro et al. (2011) provides a high probability guarantee on perfect cluster assignments by their algorithm under certain restrictions on the model and the noise level. Such an analysis can be derived from Theorem 3.4 by restating assumptions of the result in terms of the parameters of the underlying model. However, the obtained result typically involves strong conditions, similar to

Theorem 3.4, and it does not provide any error bound even under minor violations of these conditions.

3.3 Planted partition in uniform hypergraphs

We now set aside the above non-statistical analysis, and present theoretical guarantees of HOSVD under a stochastic framework, where the hypergraphs are generated from a random model. The model presented in this section is a natural extension of the stochastic block model discussed in Chapter 2.

We consider the following random model for generating a m-uniform hypergraph $(\mathcal{V}, \mathcal{E})$ for a fixed integer $m \geq 2$. Let $\mathcal{V} = \{1, 2, \ldots, n\}$ be the set of n vertices, and $\psi : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, k\}$ be a (hidden) partition of the vertices into k classes. For a vertex i, we denote its class by ψ_i. We allow the number of clusters k to grow with n, though this is not made explicit in the notation.

Let $\alpha_m \in [0, 1]$, and $\mathbf{B}^{(m)} \in [0, 1]^{k \times k \times \ldots \times k}$ be a symmetric k-dimensional tensor of order m. The edge set $\mathcal{E}$ is constructed as follows. For every $e = \{i_1, i_2, \ldots, i_m\} \subset \mathcal{V}$, we assume that the event $\{e \in \mathcal{E}\}$ occurs with probability

$$\mathsf{P}(e \in \mathcal{E}) = \mathsf{P}(\{i_1, i_2, \ldots, i_m\} \in \mathcal{E}) = \alpha_m \mathbf{B}^{(m)}_{\psi_{i_1} \psi_{i_2} \ldots \psi_{i_m}} . \tag{3.5}$$

Furthermore, the collection of all such events is assumed to be mutually independent.

It is easy to relate this model to the stochastic block model. The term α_m governs the sparsity of the hypergraph and is allowed to vary with n, whereas $\mathbf{B}^{(m)}$ specifies the label dependent edge probabilities. The only difference arises from the fact that one needs to generate edges of size m, and hence, $\mathbf{B}^{(m)}$ is a tensor of order m instead of a matrix. We also make the following remark.

Remark 3.5. In the model, we allow k and α_m to vary with n, though this is not made clear in the notations. However, the entries in $\mathbf{B}^{(m)}$ are assumed to be $\Theta(1)$, *i.e*, depend only on the class labels and not on n.

We also present an extension of the model to weighted m-uniform hypergraph $(\mathcal{V}, \mathcal{E}, w)$. To this end, observe that an unweighted m-uniform hypergraph may be viewed as a special case of $(\mathcal{V}, \mathcal{E}, w)$, where $\mathcal{E}$ is the collection of all subsets of $\mathcal{V}$ of size m, and $w : \mathcal{E} \to \{0, 1\}$. We say an edge e is present if $w(e) = 1$. From this perspective, one can describe the above model for planted hypergraphs in terms of a collection of independent Bernoulli random variables

$\{w(e) : e \in \mathcal{E}\}$ with

$$\mathsf{E}[w(\{i_1, i_2, \ldots, i_m\})] = \alpha_m \mathbf{B}^{(m)}_{\psi_{i_1}\psi_{i_2}\ldots\psi_{i_m}} \tag{3.6}$$

for all $e = \{i_1, i_2, \ldots, i_m\} \in \mathcal{E}$. For convenience, we use $w_e = w(e)$ to denote the weight of an edge $e \in \mathcal{E}$.

We extend this model to an arbitrary weight function $w : \mathcal{E} \to [0, 1]$ by assuming that $\{w_e : e \in \mathcal{E}\}$ are mutually independent random variables with first moment given by (3.6). Note here that, in this case, the sparsity factor α_m controls the expected total weight of all edges, and a smaller α_m corresponds to the presence of a large number of edges with considerably small weights.

The condition $w_e \in [0, 1]$ has been imposed for convenience, and can be relaxed to any bounded non-negative weight function. However, in this practice, one may normalize the edge weights to the interval $[0, 1]$, thereby satisfying the above condition.

As in the case of graphs, in the above setting, the objective of a partitioning algorithm is to estimate ψ from a given random instance of the m-uniform hypergraph $(\mathcal{V}, \mathcal{E}, w)$. If the labels obtained from the algorithm are given by $\psi'_1, \psi'_2, \ldots, \psi'_n$, then we bound the clustering error between ψ and ψ' as defined in (2.20). Before proving such a result, we briefly relate the above model to hypergraphs arising from practical problems.

3.3.1 Modeling hypergraph applications

We have earlier associated uniform hypergraphs with three applications: attribute clustering, subspace clustering and hypergraph matching. We show that the above planted partition model can be used to generate instances of these applications.

Attribute clustering. Here, $\mathcal{V}$ corresponds to the set of all possible values of all attributes, and each entry in the database is an edge with a fixed size, say m (see Section 2.4.2). This corresponds to an unweighted hypergraph $(\mathcal{V}, \mathcal{E})$, and can be modeled by a planted model, where $\mathbf{B}^{(m)}$ specifies the edge probabilities. The factor α_m controls the growth rate of the number of entries in the database as the number of attribute values increase. Formally, $|\mathcal{E}| = \Theta(\alpha_m |\mathcal{V}|^m)$.

Subspace clustering. Recall that in the subspace clustering problem (Section 2.4.3), the n vertices corresponds points in the Euclidean space, and edge weight are computed using (2.21) based on the error of fitting a subspace through the points.

Observe that in the absence of noise, the uniform hypergraph corresponds to an ideal hypergraph with k disjoint components, where m vertices are connected by an edge if and only

if they span a subspace, or in other words, belong to the same cluster (see Definition 3.1). However, in the presence of noise, the edge weights are no longer binary, and take values within $[0, 1]$. An edge can be expected to have a large weight if all vertices belong to the same group, else the weight is expected to be smaller. Thus, in a simplified setting, one may assume that for some $p, q \in [0, 1]$, $(p + q) \leq 1$, the tensor $\mathbf{B}^{(m)}$ is such that $\mathbf{B}^{(m)}_{i\ldots i} = p + q$ for all $i = 1, \ldots, k$, and $B_{i_1 \ldots i_m} = q$ for all the other entries.

The sparsity factor α_m also contributes to the model. Observe that if all clusters are of equal size and there is no added noise, then there are only $|\mathcal{E}| = k\binom{n/k}{m} = \Theta\left(\frac{n^m}{k^{m-1}}\right)$ edges. Thus, the above fact implies that one should let $\alpha_m = O(k^{1-m})$, which decreases if k grows with n.

Hypergraph matching. Consider the problem described in Section 2.4.5, where a weighted hypergraph $(\mathcal{V}, \mathcal{E}, w)$ is constructed with $\mathcal{V}$ being set of all candidate matches, and edge weights given by (2.22).

As noted in the problem description, this problem is quite similar to finding cliques. Hence, the corresponding planted hypergraph may be viewed as a generalization of the planted clique problem to the case of m-uniform hypergraphs. Here, $\alpha_m = 1$ and there are two classes with $\mathbf{B}^{(m)}$ such that $B_{11\ldots1} = p \approx 1$, and $B_{i_1 i_2 \ldots i_m} = q < p$ otherwise.

3.4 Consistency under planted partition model

We now derive an upper bound on the clustering error achieved by HOSVD under the above random model. The result presented here is in its general form. Special cases are considered in the next two chapters, where we analyze alternative approaches, and compare HOSVD with these methods.

Before presenting the main result, we provide some intuitive arguments about why HOSVD is expected to find the true partition. The principle idea behind stochastic block model is the block structure of the population adjacency matrix. It is worth investigating whether a similar structure exists in the aforementioned model for uniform hypergraphs. This is indeed the case as revealed by (3.6). If $\mathbf{A}$ is the random adjacency tensor of order m, it is evident that $\mathbf{A}$ is symmetric with

$$\mathsf{E}[\mathbf{A}_{i_1, i_2, \ldots, i_m}] = \alpha_m \mathbf{B}^{(m)}_{\psi_{i_1} \psi_{i_2} \ldots \psi_{i_m}} \tag{3.7}$$

whenever $\{i_1, i_2, \ldots, i_m\}$ are distinct. To clarify further, let us define the matrix $Z \in \{0, 1\}^{n \times k}$ such that $Z_{i\psi_i} = 1$, and zero otherwise. Note that Z denotes the assignment matrix correspond-

ing to the true labels. With this notation, it is easy to see that one can write

$$\mathsf{E}[\mathbf{A}] \cong \alpha_m \mathbf{B}^{(m)} \times_1 Z \times_2 Z \times_3 \ldots \times_m Z \,, \tag{3.8}$$

where we use $\cong$ to denote that the two tensors are essentially equal except at entries with repeated indices. For such entries, the population adjacency is zero but not the term on the right. For graphs ($m = 2$), the difference occurs only on the diagonal.

Figures 3.1 illustrates the situation for $m = 3$, where we assume $k = 2$ and $\alpha_m = 1$, and observe that $\mathsf{E}[\mathbf{A}]$ essentially has a block structure that can be decomposed as in (3.7). Furthermore, the decomposition on the right shows the higher order SVD of $\mathsf{E}[\mathbf{A}]$ where, under certain conditions (discussed later), the core tensor has $\mathbf{B}^{(m)}$ in the first principal block, and zero everywhere else. Corresponding columns of the associated orthonormal columns are similar to Z, up to normalization. This loosely explains the significance of considering the dominant eigenvectors of $\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^T$, or its normalized form as used in Algorithm HOSVD.

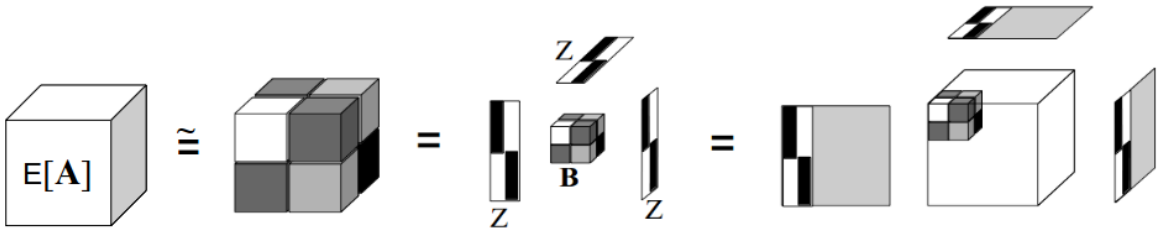

Figure 3.1: Illustration of the block structure in the population adjacency tensor, $\mathsf{E}[\mathbf{A}]$, for $m = 3$, $k = 2$ and $\alpha_m = 1$.

3.4.1 The main result

We now formally prove the above arguments. To simplify the notation, we define the matrix $A = \widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^T$. Note that $D_{ii} = \sum_j A_{ij}$. Let us also define $\mathcal{A} = \mathsf{E}[A]$ and $\mathcal{D} = \mathsf{E}[D]$. We call $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$ the population version of the normalized matrix $D^{-1/2}AD^{-1/2}$ considered in HOSVD. Also, let $n_1, \ldots, n_k$ denote the size of the k clusters, *i.e*, $n = \sum_\ell n_\ell$.

The following result formalizes the above discussions by providing a characterization of $\mathcal{A}$, revealing its block structure.

Lemma 3.6. *Let $Z \in \{0,1\}^{n\times k}$ denote the assignment matrix corresponding to the partition ψ. Then there exist matrices $G \in \mathbb{R}^{k\times k}$ and $J \in \mathbb{R}^{n\times n}$ diagonal with $J_{ii} = J_{jj}$ whenever $\psi_i = \psi_j$,*

such that $\mathcal{A}$ *can be expressed as*

$$\mathcal{A} = ZGZ^T - J\,. \tag{3.9}$$

Above lemma shows that $\mathcal{A}$ is essentially of rank k, except for the diagonal entries. Owing to the first term in (3.9), one does expect $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$, to have k eigenvectors whose entries are constant in each community. Later we show that these k eigenvectors correspond to the dominant eigenvalues under the condition $\delta > 0$. Here δ is given by

$$\delta = \left(\lambda_{\min}(G) \min_{1\le i\le n} \frac{n_{\psi_i}}{\mathcal{D}_{ii}}\right) - \max_{1\le i,j\le n} \left|\frac{J_{ii}}{\mathcal{D}_{ii}} - \frac{J_{jj}}{\mathcal{D}_{jj}}\right|\,, \tag{3.10}$$

where n_{ψ_i} is the size of the community in which node i belongs.

While the above discussions show the correctness of using the dominant eigenvectors in the population version, we still need to argue our case when one only has access to the random hypergraph. This is stated in the following theorem.

Theorem 3.7. *Let* $(\mathcal{V}, \mathcal{E}, w)$ *be a* m*-uniform hypergraph on* $|\mathcal{V}| = n$ *vertices generated from a random model with* k *planted classes. Let* $\mathcal{A}_{\min} = \min\{\mathcal{A}_{ij} : \mathcal{A}_{ij} > 0\}$, δ *be as defined in* (3.10)*, and the cluster sizes be* $n_1 \ge \ldots \ge n_k$*. Also assume* n *to be sufficiently large, and the algorithm of Ostrovsky et al. (2012) is used in the* k*-means step.*

There exists an absolute constant $C > 0$*, such that, if* $\delta > 0$ *and*

$$\mathcal{A}_{\min} > \frac{Ckn_1(\ln n)^2}{n_k\delta^2}, \tag{3.11}$$

then with probability $(1 - o(1))$*, the clustering error of HOSVD*

$$\mathrm{Error}_{\mathrm{HOSVD}}(\psi, \psi') = O\left(\frac{kn_1 \ln n}{\delta^2 \mathcal{A}_{\min}}\right) = o(n). \tag{3.12}$$

The above bound immediately implies weak consistency of HOSVD. We note here that apart from the absolute constant C and the order of the hypergraph m, other quantities vary with n. To this end, the error bound in (3.12) is true only when the quantities vary in an appropriate fashion with n such that (3.11) is satisfied.

We postpone further discussions on the implications of Theorem 3.7 to Corollary 4.6 presented in the next chapter. For the moment, we assure the reader that in the setting of subspace clustering mentioned earlier, if we assume all clusters to be of equal size, then the condition in (3.11) holds for all $k = O(\ln n)$. Thus, even for growing number of subspaces, the above

result shows that HOSVD is weakly consistent for the subspace clustering problem.

3.4.2 Proof of Theorem 3.7

We now provide an outline of the proof of the above result through a series of lemmas, which are proved in the appendix to this chapter. We start by formally proving the correctness of the algorithm in the population case discussed in Lemma 3.6.

Lemma 3.8. *If $\delta > 0$, then there exists an orthonormal matrix $U \in \mathbb{R}^{k\times k}$ such that the k dominant orthonormal eigenvectors of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$ correspond to the columns of the matrix $\mathcal{X} = Z(Z^TZ)^{-1/2}U$.*

The row normalized matrix computed in the subsequent step is given by ZU, which can be trivially clustered using k-means without incurring any error.

Thus, we can conclude that when HOSVD is performed on the population adjacency tensor $\mathsf{E}[\mathbf{A}]$, then perfect clustering is obtained. In other words, the tensor $\mathsf{E}[\mathbf{A}]$ replaces the ideal adjacency tensor used in the analysis of Section 3.2.

We now extend this observation to the random case. We still rely on matrix perturbation results, and this, in turn, requires a bound on the deviation between the random matrix $D^{-1/2}AD^{-1/2}$ and its population version $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$.

Lemma 3.9. *Let $\mathcal{A}_{\min} = \min\{\mathcal{A}_{ij} : \mathcal{A}_{ij} > 0\}$. If $\mathcal{A}_{\min} > 9(m-1)!\ln n$, then*

$$\|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2 \leq 12\sqrt{\frac{(m-1)!\ln n}{\mathcal{A}_{\min}}} \tag{3.13}$$

with probability $(1 - O(n^{-1}))$.

Due to the above result, Theorem 2.8 allows us to claim the following.

Lemma 3.10. *If $\mathcal{A}_{\min} > 9(m-1)!\ln n$ and $\delta > 24\sqrt{\frac{(m-1)!\ln n}{\mathcal{A}_{\min}}}$, then the following statements hold with probability $(1 - O(n^{-1}))$.*

1. *The matrix X does not have any row with zero norm, and hence, its row normalized form, denoted by $\overline{X}$, is well-defined.*

2. *There is an orthonormal matrix $Q \in \mathbb{R}^{k\times k}$ such that*

$$\left\|\overline{X} - ZQ\right\|_F \leq \frac{24}{\delta}\sqrt{\frac{(m-1)!2kn_1\ln n}{\mathcal{A}_{\min}}}\,. \tag{3.14}$$

From Lemmas 3.8 and 3.10, one can argue that since the rows of ZQ can be correctly clustered by k-means, and $\overline{X}$ does not deviate significantly from ZQ, hence the error can be bounded from above. We could proceed follow the lines of Theorem 3.4, and derive an error bound under an assumption on the performance of the k-means. On the other hand, we do not assume any such algorithmic guarantees and derive a bound on the error incurred by the k-means step when one uses the approximate method of Ostrovsky et al. (2012).

The performance guarantee of approximate k-means is stated in Theorem 2.12, where a sub-optimal solution is achieved if the data is well-separated. The following result shows that in our case, the rows of $\overline{X}$ are indeed well-separated.

Lemma 3.11. *If condition in (3.11) holds, then rows of $\overline{X}$ are ϵ-separated with $\epsilon = (\ln n)^{-1/2}$.*

As a consequence of Lemma 3.11, it follows that if n is sufficiently large, *i.e*, ϵ is small enough, then the result of (Ostrovsky et al., 2012) holds. Moreover, one can also observe that for large n, we have $\gamma = O(1)$. Finally, one needs to combine the above results in order to prove Theorem 3.7. For this, define the set $\mathcal{V}_{err} \subset \mathcal{V}$ as

$$\mathcal{V}_{err} = \left\{ i \in \mathcal{V} : \|S^*_{i\cdot} - Z_{i\cdot}Q\|_2 \geq \frac{1}{\sqrt{2}} \right\} . \tag{3.15}$$

Rohe et al. (2011) used a similar definition for the number of incorrectly assigned vertices, and discussed the intuition behind this definition. In the following result, we formally prove that the vertices that are not in $\mathcal{V}_{err}$ are correctly assigned. We also provide an upper bound on the size of $\mathcal{V}_{err}$.

Lemma 3.12. *Let $i, j \notin \mathcal{V}_{err}$ and $S^*_{i\cdot} = S^*_{j\cdot}$, then $\psi_i = \psi_j$. Hence,*

$$\text{Error}_{\text{HOSVD}}(\psi, \psi') \leq |\mathcal{V}_{err}|.$$

In addition,

$$|\mathcal{V}_{err}| \leq 4(1+\gamma^2)\|\overline{X} - ZQ\|_F^2 .$$

Theorem 3.7 follows by combining the above bound with (3.14), and using the fact $\gamma = O(1)$.

3.A Proofs for results in this chapter

As mentioned earlier, we do not provide proofs for results in Section 3.2. The proof of Lemma 3.3 is quite similar to that of Lemma 3.10, proved here, while Theorem 3.4 follows from by com-

bining Lemma 3.3 with arguments used in the proof of Lemma 3.12.

Proof of Lemmas 3.6 and 3.8

We begin by characterizing the matrices $A = \widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^T$ and $\mathcal{A} = \mathsf{E}[A]$. Observe that

$$A_{ij} = \sum_{\ell=1}^{n^{m-1}} \widetilde{\mathbf{A}}_{i\ell}\widetilde{\mathbf{A}}_{j\ell} = (m-1)! \sum_{\substack{i_2<\ldots<i_m \\ i,j\notin\{i_2,\ldots,i_m\}}} \mathbf{A}_{ii_2\ldots i_m}\mathbf{A}_{ji_2\ldots i_m} \ , \tag{3.16}$$

where we ignore the zero entries of $\widetilde{\mathbf{A}}$ corresponding to repeated indices. We also account for the fact that for every $i_2 < \ldots < i_m$, there are $(m-1)!$ copies due to permutation of indices. The representation in (3.16) is useful since it expresses A_{ij} as a sum of independent random random variables. Based on (3.16), one can conclude that for any $i \neq j$,

$$\mathcal{A}_{ij} = (m-1)!\alpha_m^2 \sum_{\substack{i_2<\ldots<i_m \\ i,j\notin\{i_2,\ldots,i_m\}}} \mathbf{B}^{(m)}_{\psi_i\psi_{i_2}\ldots\psi_{i_m}}\mathbf{B}^{(m)}_{\psi_j\psi_{i_2}\ldots\psi_{i_m}} \ , \tag{3.17}$$

Note that the above sum remains same if i, j are replaced by some i', j' such that $\psi_i = \psi_{i'}$ and $\psi_j = \psi_{j'}$. This is true since the terms in (3.17) depend on ψ_i, ψ_j instead of i, j. This observation motivates us to define the matrix $G \in \mathbb{R}^{k\times k}$ such that for any $i, j \in \mathcal{V}$, $i \neq j$,

$$G_{\psi_i\psi_j} = (m-1)!\alpha_m^2 \sum_{\substack{i_2<\ldots<i_m \\ i',j'\notin\{i_2,\ldots,i_m\}}} \mathbf{B}^{(m)}_{\psi_{i'}\psi_{i_2}\ldots\psi_{i_m}}\mathbf{B}^{(m)}_{\psi_{j'}\psi_{i_2}\ldots\psi_{i_m}} \ , \tag{3.18}$$

where i', j' are two distinct arbitrary vertices satisfying $\psi_i = \psi_{i'}$ and $\psi_j = \psi_{j'}$. Hence, one can write $\mathcal{A}_{ij} = (ZGZ^T)_{ij}$ for all $i \neq j$, where Z is the assignment matrix. However,

$$\mathcal{A}_{ii} = (m-1)! \sum_{\substack{i_2<\ldots<i_m \\ i\notin\{i_2,\ldots,i_m\}}} \mathsf{E}[\mathbf{A}^2_{ii_2\ldots i_m}] \neq G_{\psi_i\psi_i} \ .$$

So, one can write the matrix $\mathcal{A}$ as $\mathcal{A} = ZGZ^T - J$, where $J \in \mathbb{R}^{n\times n}$ is a diagonal matrix defined as $J_{ii} = G_{\psi_i\psi_i} - \mathcal{A}_{ii}$. We also note that for i, i' in the same group, *i.e*, $\psi_i = \psi_{i'}$, we have $\mathcal{D}_{ii} = \mathcal{D}_{i'i'}$ and $J_{ii} = J_{i'i'}$. So we can define matrices $\widetilde{\mathcal{D}}, \widetilde{J} \in \mathbb{R}^{k\times k}$ diagonal such that $\mathcal{D}_{ii} = \widetilde{\mathcal{D}}_{\psi_i\psi_i}$ and $J_{ii} = \widetilde{J}_{\psi_i\psi_i}$ for all $i \in \mathcal{V}$. It is easy to see that $\mathcal{D}Z = Z\widetilde{\mathcal{D}}$ and $JZ = Z\widetilde{J}$.

Using above definitions, we now characterize the eigenpairs of the matrix $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$. First, observe that since $G \in \mathbb{R}^{k\times k}$, $\mathcal{A}$ is composed of a matrix of rank at most k that is

perturbed by the diagonal matrix J. We show that the orthonormal basis for the range space of ZGZ^T are the eigenvectors that are of interest to us. For this, consider the matrix $\mathcal{G} = (\widetilde{\mathcal{D}}^{-1}Z^TZ)^{1/2}G(Z^TZ\widetilde{\mathcal{D}}^{-1})^{1/2} - \widetilde{J}\widetilde{\mathcal{D}}^{-1} \in \mathbb{R}^{k\times k}$, and suppose its eigen-decomposition is given by $\mathcal{G} = U\Lambda_1 U^T$, where $U \in \mathbb{R}^{k\times k}$ contains the orthonormal eigenvectors and $\Lambda_1 \in \mathbb{R}^{k\times k}$ is a diagonal matrix of eigenvalues of $\mathcal{G}$. Defining $\mathcal{X} = Z(Z^TZ)^{-1/2}U \in \mathbb{R}^{n\times k}$, we can write that

$$\begin{aligned}
\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\mathcal{X} &= \mathcal{D}^{-1/2}(ZGZ^T - J)\mathcal{D}^{-1/2}Z(Z^TZ)^{-1/2}U \\
&= \mathcal{D}^{-1/2}(ZG(Z^TZ)^{1/2} - Z(Z^TZ)^{-1/2}\widetilde{J})\widetilde{\mathcal{D}}^{-1/2}U \\
&= Z(Z^TZ)^{-1/2}\mathcal{G}U \\
&= Z(Z^TZ)^{-1/2}U\Lambda_1 = \mathcal{X}\Lambda_1,
\end{aligned}$$

which implies that the columns of $\mathcal{X}$ are the eigenvectors of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$ corresponding to the k eigenvalues in Λ_1. Note that the above equalities are derived by repeated use of the facts that diagonal matrices commute and $\mathcal{D}Z = Z\widetilde{\mathcal{D}}$, $JZ = Z\widetilde{J}$. Also, since U is orthonormal, it is easy to verify that the columns of $\mathcal{X}$ are orthonormal. We need to derive conditions under which $\mathcal{X}$ contain the dominant eigenvectors of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$. Equivalently, we need to show that the eigenvalues in Λ_1 are strictly larger than other eigenvalues of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$.

Since, $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$ is symmetric and hence, diagonalizable, we can conclude that remaining eigenvectors of the matrix are orthogonal to columns of $\mathcal{X}$. Let the columns of $Y \in \mathbb{R}^{n\times(n-k)}$ be the matrix of the remaining orthonormal eigenvectors of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$, with corresponding eigenvalues given by the diagonal matrix $\Lambda_2 \in \mathbb{R}^{(n-k)\times(n-k)}$. So $Y^TZ(Z^TZ)^{-1/2}U = 0$. Due to the non-singularity of Z^TZ or U, it follows that $Z^TY = 0$, and

$$Y\Lambda_2 = \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}Y = -\mathcal{D}^{-1}JY,$$

that is, the columns of Y are eigenvectors of $(-\mathcal{D}^{-1}J)$. Further, since $\mathcal{D}^{-1}J$ is diagonal, the eigenvalues in Λ_2 are a subset of the entries of $(-\mathcal{D}^{-1}J)$. Thus, to ensure that $\mathcal{X}$ are the leading eigenvectors, one needs to ensure $\min_i(\Lambda_1)_{ii} > \max_i(\Lambda_2)_{ii}$, and hence, one may define $\widetilde{\delta}$ as the eigen-gap,

$$\widetilde{\delta} = \min_{1\le i\le k}(\Lambda_1)_{ii} - \max_{1\le i\le(n-k)}(\Lambda_2)_{ii}. \tag{3.19}$$

The condition $\widetilde{\delta} > 0$ ensures that columns of $\mathcal{X}$ are the dominant eigenvectors of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$. Though the above definition of $\widetilde{\delta}$ suffices, it cannot be easily verified for a given model. Below,

we show that $\widetilde{\delta} \geq \delta$, where the latter is as defined in (3.10). Note that

$$\max_{1\leq i\leq (n-k)} (\Lambda_2)_{ii} \leq \max_{1\leq i\leq n} \left(-\frac{J_{ii}}{\mathcal{D}_{ii}}\right) = \min_{1\leq i\leq n} \frac{J_{ii}}{\mathcal{D}_{ii}}$$

On the other hand, using Weyl's inequality, we have

$$\min_{1\leq i\leq k} (\Lambda_1)_{ii} = \lambda_{\min}(\mathcal{G}) \geq \lambda_{\min}((\widetilde{\mathcal{D}}^{-1}Z^TZ)^{1/2}G(Z^TZ\widetilde{\mathcal{D}}^{-1})^{1/2}) - \|\widetilde{J}\widetilde{\mathcal{D}}^{-1}\|_2 \ ,$$

where $\lambda_{\min}(\mathcal{G})$ denotes the minimum eigenvalue of $\mathcal{G}$. The inequality follows by viewing $\mathcal{G}$ as the matrix $(\widetilde{\mathcal{D}}^{-1}Z^TZ)^{1/2}G(Z^TZ\widetilde{\mathcal{D}}^{-1})^{1/2}$ perturbed by $-\widetilde{\mathcal{D}}^{-1}\widetilde{J}$. To simplify further, we note

$$\|\widetilde{J}\widetilde{\mathcal{D}}^{-1}\|_2 = \max_{1\leq i\leq k} \frac{\widetilde{J}_{ii}}{\widetilde{\mathcal{D}}_{ii}} = \max_{1\leq i\leq n} \frac{J_{ii}}{\mathcal{D}_{ii}},$$

and using Rayleigh's principle, one can show that

$$\lambda_{\min}((\widetilde{\mathcal{D}}^{-1}Z^TZ)^{1/2}G(Z^TZ\widetilde{\mathcal{D}}^{-1})^{1/2}) \geq \lambda_{\min}(G) \min_{1\leq i\leq k} \frac{(Z^TZ)_{ii}}{\widetilde{\mathcal{D}}_{ii}}.$$

Combining the above bounds, we conclude that $\widetilde{\delta} \geq \delta$. Here, we use the observation that $(Z^TZ)_{jj}$ equals the size of the j^{th} community. Thus, $\delta > 0$ is a sufficient condition for the first claim of Lemma 3.8.

The second claim is straightforward. Since $\mathcal{X} = (Z^TZ)^{-1/2}ZU$, it is easy to verify that the norm of the i^{th} row of $\mathcal{X}$ is $\|\mathcal{X}_{i\cdot}\|_2 = \frac{1}{\sqrt{Z^TZ_{\psi_i\psi_i}}}$. Hence, row normalization of $\mathcal{X}$ provides the matrix ZU. Moreover, the nature of the matrix Z implies that $(ZU)_{i\cdot} = U_{\psi_i\cdot}$, and hence, there are only k distinct rows in ZU each corresponding to a particular cluster. As a consequence, k-means trivially provides the optimal result.

Proof of Lemma 3.9

We observe that

$$\begin{aligned}
&\|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2 \\
&\quad\leq \|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}AD^{-1/2}\|_2 + \|\mathcal{D}^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}A\mathcal{D}^{-1/2}\|_2 \\
&\qquad\qquad + \|\mathcal{D}^{-1/2}(A-\mathcal{A})\mathcal{D}^{-1/2}\|_2 \\
&\quad\leq \|I - D^{1/2}\mathcal{D}^{-1/2}\|_2 + \|D^{1/2}\mathcal{D}^{-1/2}\|_2\|I - D^{1/2}\mathcal{D}^{-1/2}\|_2 + \|\mathcal{D}^{-1}(A-\mathcal{A})\|_2
\end{aligned}$$

since $\|D^{-1/2}AD^{-1/2}\|_2 = 1$.

Let $t = 3\sqrt{\frac{(m-1)!\ln n}{\mathcal{A}_{\min}}}$, and observe that $t < 1$ under the condition $\mathcal{A}_{\min} > 9(m-1)!\ln n$. We claim that for every $i = 1, \ldots, n$,

$$\mathsf{P}\left(\sum_{j=1}^{n} |A_{ij} - \mathcal{A}_{ij}| > t\mathcal{D}_{ii}\right) \leq \frac{2}{n^2}\,. \tag{3.20}$$

Assuming that (3.20) is true, we can observe that the following hold with probability $(1 - \frac{2}{n})$:

$$\max_i \frac{1}{\mathcal{D}_{ii}} \sum_{j=1}^{n} |A_{ij} - \mathcal{A}_{ij}| \leq t\,. \tag{3.21}$$

Due to this, one can also verify that

$$\|\mathcal{D}^{-1}(A - \mathcal{A})\|_2 \leq \max_i \frac{1}{\mathcal{D}_{ii}} \sum_{j=1}^{n} |A_{ij} - \mathcal{A}_{ij}| \leq t \tag{3.22}$$

and

$$\begin{aligned} \|I - D^{1/2}\mathcal{D}^{-1/2}\|_2 &= \max_i \left|\sqrt{\frac{D_{ii}}{\mathcal{D}_{ii}}} - 1\right| \\ &= \max_i \left|\frac{D_{ii}}{\mathcal{D}_{ii}} - 1\right| \leq \max_i \frac{1}{\mathcal{D}_{ii}} \sum_{j=1}^{n} |A_{ij} - \mathcal{A}_{ij}| \leq t. \end{aligned} \tag{3.23}$$

Similarly, $\|D^{1/2}\mathcal{D}^{-1/2}\|_2 \leq \|I + D^{1/2}\mathcal{D}^{-1/2}\|_2 \leq (1 + t) < 2$ since $t < 1$. Thus, we can conclude that

$$\|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2 \leq 4t$$

with probability $1 - O(n^{-1})$, which is the claim of the lemma.

We complete the proof by deriving the bound in (3.20). Note that $\mathcal{D}_{ii} = \sum_j \mathcal{A}_{ij} = \sum_{j:\mathcal{A}_{ij}>0} \mathcal{A}_{ij}$. Hence, we can bound

$$\mathsf{P}\left(\sum_{j=1}^{n} |A_{ij} - \mathcal{A}_{ij}| > t\mathcal{D}_{ii}\right) \leq \sum_{j:\mathcal{A}_{ij}>0} \mathsf{P}\left(|A_{ij} - \mathcal{A}_{ij}| > t\mathcal{A}_{ij}\right)\,.$$

Thus, it suffices to show that for every i, j

$$\mathsf{P}\left(|A_{ij} - \mathcal{A}_{ij}| > t\mathcal{A}_{ij}\right) \leq \frac{2}{n^3}\,. \tag{3.24}$$

To prove (3.24), we recall from (3.16) that A_{ij} is a sum of independent random variables, and we can use Bernstein inequality (Theorem 2.10) to bound its deviation from $\mathcal{A}_{ij}$. So, we have

$$\begin{aligned}
&\mathsf{P}\left(|A_{ij} - \mathcal{A}_{ij}| > t\mathcal{A}_{ij}\right) \\
&\leq \mathsf{P}\left(\left|\sum_{\substack{i_2<\ldots<i_m \\ i,j\notin\{i_2,\ldots,i_m\}}} \mathbf{A}_{ii_2\ldots i_m}\mathbf{A}_{ji_2\ldots i_m} - \mathsf{E}[\mathbf{A}_{ii_2\ldots i_m}\mathbf{A}_{ji_2\ldots i_m}]\right| > \frac{t\mathcal{A}_{ij}}{(m-1)!}\right) \\
&\leq 2\exp\left(\frac{-\frac{t^2\mathcal{A}_{ij}^2}{2(m-1)!}}{\sum_{\substack{i_2<\ldots<i_m \\ i,j\notin\{i_2,\ldots,i_m\}}} \mathsf{Var}(\mathbf{A}_{ii_2\ldots i_m}\mathbf{A}_{ji_2\ldots i_m}) + \frac{t\mathcal{A}_{ij}}{3(m-1)!}}\right)\,.
\end{aligned}$$

Recall the assumption that the edge weight lie in $[0, 1]$. Hence, one can argue that

$$\mathsf{Var}(\mathbf{A}_{ii_2\ldots i_m}\mathbf{A}_{ji_2\ldots i_m}) \leq \mathsf{E}[\mathbf{A}_{ii_2\ldots i_m}\mathbf{A}_{ji_2\ldots i_m})].$$

So, one can bound the sum of variances from above by $\mathcal{A}_{ij}$. Using the condition $t < 1$ and $\mathcal{A}_{ij} \geq \mathcal{A}_{\min} > 9(m-1)!\ln n$, one arrives at (3.24).

Proof of Lemma 3.10

This proof is a typical example of the use of the matrix perturbation results, Theorems 2.7 and 2.8, in the analysis of spectral methods.

Note that in order to perform a valid row normalization of X, first we need to ensure that the rows of X are non-zero with high probability. We use standard graph theoretic arguments to prove this. For this, consider a graph with weighted adjacency matrix A. Then D_{ii} corresponds to the degree of vertex i, and the subsequent steps of HOSVD correspond to normalized spectral clustering. The bound (3.23) shows that for all $i \in \mathcal{V}$,

$$D_{ii} \geq \mathcal{D}_{ii}\left(1 - 3\sqrt{\frac{(m-1)!\ln n}{\mathcal{A}_{\min}}}\right) > 0$$

under the condition on $\mathcal{A}_{\min}$. Thus, the associated graph does not have any isolated vertex

with probability $(1 - O(n^{-1}))$.

von Luxburg (2007) studied the normalized graph Laplacian $L = I - D^{-1/2}AD^{-1/2}$, and concludes the following:

- L is positive semi-definite with eigenvalues in $[0, 2]$.
- The multiplicity of 0 eigenvalue of L is equal to the number of connected components of the hypergraph.
- Provided that $D_{ii} > 0$ for all i, for each zero eigenvalue of L, there is an eigenvector with non-zero coordinates for one connected component.

Though we deal with the matrix $D^{-1/2}AD^{-1/2}$ instead of the Laplacian, one can note that these matrices are equivalent in the sense that if (λ, v) is an eigen pair of $D^{-1/2}AD^{-1/2}$, then $(1 - \lambda, v)$ is an eigen pair of L. Hence, the dominant k eigenvectors of $D^{-1/2}AD^{-1/2}$, given in X, are same as the leading eigenvectors of L.

From the condition $\delta > 0$ in Lemma 3.8, there is a strictly positive eigen-gap between the k^{th} and $(k+1)^{th}$ eigenvalues of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$, and hence, $\mathcal{L} = I - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$ has atmost k zero eigenvalues. Denoting $\lambda_i(L), \lambda_i(\mathcal{L})$ as the i^{th} smallest eigenvalues of $L, \mathcal{L}$ respectively, we have $\delta \leq \widetilde{\delta} = \lambda_{k+1}(\mathcal{L}) - \lambda_k(\mathcal{L})$, where $\widetilde{\delta}$ is defined in (3.19). Also, we can use Weyl's inequality (Theorem 2.7) to claim that for all $i = 1, \ldots, n$,

$$|\lambda_i(L) - \lambda_i(\mathcal{L})| \leq \|L - \mathcal{L}\|_2 = \|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2 \leq 12\sqrt{\frac{(m-1)!\ln n}{\mathcal{A}_{\min}}} < \frac{\delta}{2}$$

if δ and $\mathcal{A}_{\min}$ satisfy the prescribed condition. Thus

$$\lambda_{k+1}(L) \geq \lambda_{k+1}(\mathcal{L}) - \frac{\delta}{2} = \lambda_k(\mathcal{L}) + \frac{\delta}{2} > 0,$$

which means L has at most k zero eigenvalues, *i.e*, at most k connected components. Since, all vertices have positive degrees almost surely, hence, every vertex corresponds to a connected component. Due to third property of L, for every vertex, at least one of the k leading eigenvectors has a non-zero component, and hence, every row of X is non-zero. Thus, $\overline{X}$ is well-defined.

We now prove the second claim of the lemma. Let us view $D^{-1/2}AD^{-1/2}$ as an additive perturbation of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$. Corollary 2.9 derived from the Davis-Kahan perturbation theorem suggests that for an orthonormal matrix $U_1 \in \mathbb{R}^{k\times k}$,

$$\|X - \mathcal{X}U_1\|_F \leq \sqrt{2k}\frac{\|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2}{\delta},$$

and hence,

$$\|X - Z(Z^TZ)^{-1/2}Q\|_F \leq \sqrt{2k}\frac{\|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2}{\delta} \,, \tag{3.25}$$

where $Q = UU_1$ is also orthonormal.

One can see that i^{th} row of $Z(Z^TZ)^{-1/2}Q$ is $Z_{i\cdot}(Z^TZ)^{-1/2}Q$, and the norm of i^{th} row is $(Z^TZ)^{-1/2}_{\psi_i\psi_i}$. Thus, on row normalization of this matrix, one obtains ZQ. Hence,

$$\begin{aligned}
\|\overline{X} - ZQ\|_F^2 &= \sum_{i=1}^{n} \left\| \tfrac{1}{\|X_{i\cdot}\|_2} X_{i\cdot} - Z_{i\cdot}Q \right\|_2^2 \\
&= \sum_{i=1}^{n} \left\| \left(\tfrac{1}{\|X_{i\cdot}\|_2} - (Z^TZ)^{1/2}_{\psi_i\psi_i} \right) X_{i\cdot} + (Z^TZ)^{1/2}_{\psi_i\psi_i} \left(X_{i\cdot} - Z_{i\cdot}(Z^TZ)^{-1/2}Q \right) \right\|_2^2
\end{aligned}$$

Now,

$$\begin{aligned}
&\left\| \left(\tfrac{1}{\|X_{i\cdot}\|_2} - (Z^TZ)^{1/2}_{\psi_i\psi_i} \right) X_{i\cdot} + (Z^TZ)^{1/2}_{\psi_i\psi_i} \left(X_{i\cdot} - Z_{i\cdot}(Z^TZ)^{-1/2}Q \right) \right\|_2^2 \\
&\leq \sqrt{(Z^TZ)_{\psi_i\psi_i}} \left(\left| \|Z_{i\cdot}(Z^TZ)^{-1/2}Q\|_2 - \|X_{i\cdot}\|_2 \right| + \|X_{i\cdot} - Z_{i\cdot}(Z^TZ)^{-1/2}Q\|_2 \right) \\
&\leq 2\sqrt{(Z^TZ)_{\psi_i\psi_i}} \|X_{i\cdot} - Z_{i\cdot}(Z^TZ)^{-1/2}Q\|_2 \,.
\end{aligned}$$

Substituting this bound above, we get

$$\begin{aligned}
\|\overline{X} - ZQ\|_F^2 &\leq 4\sum_{i=1}^{n} (Z^TZ)_{\psi_i\psi_i} \|X_{i\cdot} - Z_{i\cdot}(Z^TZ)^{-1/2}Q\|_2^2 \\
&\leq 4n_1 \|X - Z(Z^TZ)^{-1/2}Q\|_F^2 \,,
\end{aligned}$$

where $n_1 = \max_j (Z^TZ)_{jj}$ is the size of the largest cluster since we assumed that $n_1 \geq \ldots \geq n_k$. The bound in (3.14) follows by combining above bound with (3.25) and Lemma 3.9.

Proof of Lemma 3.11

From (3.14), we have an upper bound on $\|\overline{X} - ZQ\|_F$ with probability $(1 - O(n^{-1}))$. For convenience, let us denote this upper bound by β. The condition in (3.11) implies that

$$\beta \leq 24\sqrt{\frac{2(m-1)!n_k}{C \ln n}} \leq \frac{\epsilon\sqrt{n_k}}{2}$$

if C is chosen sufficiently large. For large enough n, above inequality implies

$$\beta \leq \epsilon \left(\sqrt{n_k} - \beta\right), \tag{3.26}$$

which will be used to prove ϵ-separability, *i.e*, $\eta_k(\overline{X}) \leq \epsilon\eta_{k-1}(\overline{X})$. Since ZQ has exactly k distinct rows, we have $\eta_k(\overline{X}) \leq \|X - ZQ\|_F \leq \beta$. On the other hand, observe that all matrices in $\mathcal{M}_{n\times k}(r)$ have rank at most r. Hence,

$$\eta_{k-1}(\overline{X}) = \min_{S\in\mathcal{M}_{n\times k}(k-1)} \|\overline{X} - S\|_F \geq \min_{\text{rank}(S)\leq(k-1)} \|\overline{X} - S\|_F \ .$$

It is well known that the minimum of the last quantity is $\lambda_k(\overline{X})$, which is the smallest singular value of $\overline{X}$. Also Mirsky's theorem (Stewart and Sun, 1990) gives a bound on the perturbation of singular values, and hence, we have

$$\left|\lambda_i(\overline{X}) - \lambda_i(ZQ)\right| \leq \|\overline{X} - ZQ\|_2 \leq \beta$$

for $i = 1, \ldots, k$. Note here that the singular values of ZQ are $\lambda_i(ZQ) = \sqrt{n_i}$, where the ordering of the values is due to our assumption $n_1 \geq \ldots \geq n_k$. From above arguments

$$\eta_{k-1}(\overline{X}) \geq \lambda_k(\overline{X}) \geq (\lambda_k(ZQ) - \beta) = (\sqrt{n_k} - \beta) \ .$$

Hence, it follows that ϵ-separability holds when (3.26) is satisfied, which is true under the assumption in (3.11).

Proof of Lemma 3.12

We observe from the proof of Lemma 3.10 that the k distinct rows of ZQ, form an orthonormal set of vectors in $\mathbb{R}^k$. Hence, for any $i, j \in \mathcal{V}$, $\|Z_{i\cdot}Q - Z_{j\cdot}Q\|_2 = 0$ or $\sqrt{2}$, where the former occurs if $Z_{i\cdot} = Z_{j\cdot}$, *i.e*, $\psi_i = \psi_j$, and the latter occurs if $\psi_i \neq \psi_j$. Now, consider $i, j \notin \mathcal{V}_{err}$. We have $\|S^*_{i\cdot} - Z_{i\cdot}Q\|_2 < \frac{1}{\sqrt{2}}$, $\|S^*_{j\cdot} - Z_{j\cdot}Q\|_2 < \frac{1}{\sqrt{2}}$, and hence,

$$\|Z_{i\cdot}Q - Z_{j\cdot}Q\|_2 \leq \|S^*_{i\cdot} - Z_{i\cdot}Q\|_2 + \|S^*_{j\cdot} - Z_{j\cdot}Q\|_2 + \|S^*_{i\cdot} - S^*_{j\cdot}\|_2 < \sqrt{2}$$

whenever $S^*_{i\cdot} = S^*_{j\cdot}$. So, from the previous observation, $\|Z_{i\cdot}Q - Z_{j\cdot}Q\|_2 = 0$, *i.e*, $\psi_i = \psi_j$, which proves the first claim.

To prove the second claim, note that for all $i \in \mathcal{V}_{err}$, we have $2\|S^*_{i\cdot} - Z_{i\cdot}Q\|_2^2 \geq 1$. Therefore,

$$|\mathcal{V}_{err}| = \sum_{i \in \mathcal{V}_{err}} 1 \leq 2 \sum_{i \in \mathcal{V}_{err}} \|S^*_{i\cdot} - Z_{i\cdot}Q\|_2^2 \leq 2\|S^* - ZQ\|_F^2 \,. \tag{3.27}$$

Since, S^* is a sub-optimal solution satisfying (2.25), we can write $\|\overline{X} - S^*\|_F \leq \gamma\|\overline{X} - S\|_F$, for all $S \in \mathcal{M}_{n\times k}(k)$. In particular, $ZQ \in \mathcal{M}_{n\times k}(k)$ and so $\|\overline{X} - S^*\|_F \leq \gamma\|\overline{X} - ZQ\|_F$. Hence,

$$\|S^* - ZQ\|_F \leq \|\overline{X} - ZQ\|_F + \|\overline{X} - S^*\|_F \leq (1+\gamma)\|\overline{X} - ZQ\|_F \,.$$

The lemma follows by combining above inequality with (3.27), and using the relation $(1+\gamma)^2 \leq 2(1+\gamma^2)$.

'As you yourself have said, what other explanation can there be?'
Poirot stared straight ahead of him. 'That is what I ask myself,' he said. 'That is what I never cease to ask myself.'

Agatha Christie, *Murder on the Orient Express*

Chapter 4

Revisiting Uniform Hypergraph Partitioning

We begin this chapter with our minds reset to the starting point of the hypergraph partitioning problem. We attempt to formulate the problem from the scratch following the lines of the graph partitioning formulation developed in Section 2.2. Extensions of the cut minimization problems (2.12)–(2.13) have been studied for both uniform (Hu and Qi, 2012) as well as non-uniform hypergraphs (Bolla, 1993; Zhou et al., 2007). The latter extension will be discussed in Chapter 5, while the former is known to be quite complex and applicable only for even uniform hypergraphs. Hence, the results in (Hu and Qi, 2012) do not immediately lead to practical algorithms.

Instead of discussing hypergraph cut formulations, we dedicate this chapter for extension of the associativity maximization problems (2.14)–(2.15) to uniform hypergraphs. Section 4.1 presents this extension, while Section 4.2 shows that our formulation has connections with a wide variety of approaches in machine learning as well as some tensor problems. We then present a spectral algorithm in Section 4.3, and establish its consistency, and superiority over HOSVD. The proofs of the technical results are given in Appendix 4.A. Further extension to non-uniform hypergraphs is postponed to Chapter 5.

4.1 Tensor trace maximization for uniform hypergraphs

We define the notion of associativity and volume in the case of a weighted hypergraph $(\mathcal{V}, \mathcal{E}, w)$. The corresponding definitions for unweighted hypergraph can be obtained by simply considering a binary weight function $w : \mathcal{E} \to \{0, 1\}$. For ease of notation, we use $w_e = w(e)$ to denote the

weight of an edge $e \in \mathcal{E}$.

The degree of a vertex $i \in \mathcal{V}$ is the total weight of edges on which i is incident, *i.e*, $\deg(i) = \sum_{e \in \mathcal{E}: i \in e} w_e$. For any collection of vertices $\mathcal{V}_1 \subseteq \mathcal{V}$, let the volume be defined as in graphs, $\mathrm{Vol}(\mathcal{V}_1) = \sum_{i \in \mathcal{V}_1} \deg(i)$, and the associativity of $\mathcal{V}_1$ be given by $\mathrm{Assoc}(\mathcal{V}_1) = \sum_{e \in \mathcal{E}: e \in \mathcal{V}_1} w_e |e|$. We define the normalized hypergraph associativity of a partition $\mathcal{V}_1, ..., \mathcal{V}_k$ as

$$\text{NH-Assoc}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \sum_{\ell=1}^{k} \frac{\mathrm{Assoc}(\mathcal{V}_\ell)}{\mathrm{Vol}(\mathcal{V}_\ell)} = \sum_{\ell=1}^{k} \sum_{e \in \mathcal{E}: e \subset \mathcal{V}_\ell} \frac{w_e |e|}{\mathrm{Vol}(\mathcal{V}_\ell)} \,. \tag{4.1}$$

Observe that the above definition of normalized associativity coincides with the same quantity defined in graphs (Shi and Malik, 2000), up to a factor of 2. For a m-uniform hypergraph, the associativity of a cluster $\mathcal{V}_1$ is the total weight of edges contained in $\mathcal{V}_1$ scaled by a factor m. The reason for this additional will be discussed in Remark 5.3 in Chapter 5, where we deal with non-uniform hypergraphs.

We now consider the problem of finding a partition of $\mathcal{V}$ that maximizes the normalized hypergraph associativity (4.1). Section 2.2 discusses a reformulation of the problem in terms of the adjacency matrix. Subsequently, a trace maximization[1] objective can be stated, whose spectral relaxation leads to the spectral clustering algorithm. A similar approach is possible in the case of m-uniform hypergraphs. Let $|\mathcal{V}| = n$ and $\mathbf{A} \in \mathbb{R}^{n \times \ldots \times n}$ be the weighted adjacency tensor of order m such that

$$\mathbf{A}_{i_1 i_2 \ldots i_m} = \begin{cases} w_e & \text{if there exists } e \in \mathcal{E} \text{ such that } e = \{i_1, i_2, \ldots, i_m\}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

We make the following claim, which is derived in Appendix 4.A.

Proposition 4.1. *Consider* $\beta_1, \ldots, \beta_m \in [0, 1]$ *such that* $\sum_{r=1}^{m} \beta_r = 1$, *and define the matrices* $\overline{Y}^{(1)}, \ldots, \overline{Y}^{(m)} \in \mathbb{R}^{n \times k}$ *with*

$$\overline{Y}^{(r)}_{i\ell} = \left(\frac{\mathbb{1}\{i \in \mathcal{V}_\ell\}}{\mathrm{Vol}(\mathcal{V}_\ell)} \right)^{\beta_r} . \tag{4.3}$$

Then, one may express the normalized associativity (4.1) *as*

$$\text{NH-Assoc}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \frac{1}{(m-1)!} \mathrm{Trace} \left(\mathbf{A} \times_1 \overline{Y}^{(1)^T} \times_2 \overline{Y}^{(2)^T} \times_3 \ldots \times_m \overline{Y}^{(m)^T} \right) . \tag{4.4}$$

[1] The trace minimization for normalized graph Laplacian (2.18) is equivalent to a trace maximization problem for the normalized adjacency matrix $D^{-1/2} A D^{-1/2}$.

Thus, for a chosen set of parameters $\beta_1, \ldots, \beta_m$, one can pose the normalized associativity maximization problem as the that of multiplying a tensor with appropriate orthogonal matrices such that the trace of the resultant tensor is maximized. More precisely, the optimization problem at hand is the following:

$$\underset{\overline{Y}^{(1)},\ldots,\overline{Y}^{(m)}}{\text{maximize}} \ \text{Trace}\left(\mathbf{A} \times_1 \overline{Y}^{(1)^T} \times_2 \overline{Y}^{(2)^T} \times_3 \ldots \times_m \overline{Y}^{(m)^T}\right) , \tag{4.5}$$

where $\overline{Y}^{(1)}, \ldots, \overline{Y}^{(m)}$ are of the form given in (4.3). We call this a tensor trace maximization (TTM) problem, and show that this problem lies at the heart of higher-order learning. One can also extend the definition of ratio associativity (2.15) to uniform hypergraphs, where $\text{Vol}(\mathcal{V}_\ell)$ is replaced by $|\mathcal{V}_\ell|$ in (4.1).

4.2 Connection with existing works

The purpose of this section is to demonstrate the ubiquity of the problem of maximizing NH-Assoc$(\mathcal{V}_1, \ldots, \mathcal{V}_k)$. This formulation unifies a variety of uniform hypergraph partitioning techniques that are apparently quite different from each other. Moreover, the trace maximization problem also finds use in signal processing, and has connections to the tensor eigenvalue problem.

4.2.1 Relation with popular partitioning algorithms

Spectral clustering. Recall the formulation for normalized spectral clustering (Ng et al., 2002) discussed in Section 2.2. The problem in (2.18) can be expressed for maximization of normalized associativity (2.14), which reads as

$$\underset{Y}{\text{minimize}} \ \text{Trace}\left(Y^T D^{-1/2} A D^{-1/2} Y\right) , \tag{4.6}$$

where $Y \in \mathbb{R}^{n\times k}$ is of the form given in (2.17).

Consider the TTM problem (4.5) for $m = 2$, and choose $\beta_1 = \beta_2 = \frac{1}{2}$. It immediately follows that (4.5) is equivalent to (4.6), where $\overline{Y}^{(1)} = \overline{Y}^{(2)} = D^{-1/2}Y$. Modifications of the objective function in (4.5) retrieves other variants of spectral clustering such as maximizing ratio associativity (2.15), spectral relaxation of the k-means algorithm Zha et al. (2001) or or spectral clustering with doubly stochastic normalization Zass and Shashua (2006). For instance, ratio associativity maximization follows from (4.5) by choosing $\beta_1 = \beta_2 = \frac{1}{2}$ and

replacing $\mathrm{Vol}(\mathcal{V}_\ell)$ by $|\mathcal{V}_\ell|$.

Non-negative tensor factorization. Shashua et al. (2006) generalized the use of non-negative matrix factorization for clustering to the case of tensors. The objective is to approximate a normalized version of the adjacency tensor $\mathbf{A}$ by a sum of k rank-one tensors similar to the CP-decomposition (Definition 2.5). The approximation is justified using probabilistic arguments, and the associated optimization problem is stated as

$$\underset{y_1,\ldots,y_k\in[0,\infty)^n \text{ orthonormal}}{\text{minimize}} \left\| \mathbf{A}' - \sum_{j=1}^{k} y_j^{\otimes m} \right\|_F^2 , \tag{4.7}$$

where $\mathbf{A}'$ is a normalized version $\mathbf{A}$ and $\|\cdot\|_F^2$ is the sum of squares of entries in the tensor (similar to the matrix Frobenius norm). Ignoring the normalization, one can observe that

$$\left\| \mathbf{A} - \sum_{j=1}^{k} y_j \otimes y_j \otimes \ldots \otimes y_j \right\|_F^2 = \|\mathbf{A}\|_F^2 + k - 2\,\mathrm{Trace}\left(\mathbf{A} \times_1 Y^T \times_2 \ldots \times_m Y^T\right),$$

where $Y = [y_1 \ldots y_k]$. Thus, (4.7) is equivalent to a relaxation of (4.5), where $\overline{Y}^{(1)} = \ldots = \overline{Y}^{(m)} = Y$ is allowed to be a non-negative matrix with orthonormal columns.

Hypergraph reduction by clique expansion. In Chapter 2, we discussed the approach of reducing a hypergraph to a graph. Agarwal et al. (2006) showed that various reduction strategies and hypergraph Laplacian definitions primarily rely on a clique expansion or a star expansion of a hypergraph. Moreover, both expansions have similar spectral properties in the case of uniform hypergraphs.

Consider the spectral approach of (Agarwal et al., 2005), where the authors propose a spectral partitioning of the graph obtained via clique expansion. This is equivalent to solving (2.18) or (4.6) for a graph with adjacency matrix $A \in \mathbb{R}^{n\times n}$ given by $A_{ij} = \sum_{e\in\mathcal{E}: e\ni i,j} w_e$. For a m-uniform hypergraph, this can be written in terms of the adjacency tensor $\mathbf{A}$ as

$$A_{ij} = \frac{1}{(m-2)!} \sum_{i_3,\ldots,i_m=1}^{n} \mathbf{A}_{iji_3\ldots i_m} . \tag{4.8}$$

A similar reduction, without the constant scaling, was is used in the tensor based approach geometric grouping method of (Arias-Castro et al., 2011).

The combined formulation of (4.8) followed by (4.6) is a special case of the TTM problem (4.5), where $\beta_1 = \beta_2 = \frac{1}{2}$ and $\beta_3 = \ldots = \beta_m = 0$. This follows since for $\beta_r = 0$, $\overline{Y}^{(r)}$ is

a constant matrix of ones, which in turn leads to summation of all entries of $\mathbf{A}$ along the r^{th} dimension. Hence, one obtains the matrix A in (4.8) for $\beta_3 = \ldots = \beta_m = 0$.

Hypergraph matching via tensor power iterations. The hypergraph matching problem, described in Section 2.4.5, is to find one-one correspondences between two sets, each consisting of s points. As discussed in Section 2.4.5, the problem is similar to partitioning a m-uniform hypergraph with $n = s^2$ vertices.

Let $\mathbf{A}$ be the associated weighted adjacency tensor, and let $S \in \{0,1\}^{s \times s}$ denote the correspondence matrix. Note that $\|S\|_F^2 = s$. Typically, one solves this problem by optimizing a score function as (Duchenne et al., 2011; Lee et al., 2011)

$$\underset{y \in \{0,1\}^{s^2} : \|y\|_2^2 = s}{\text{maximize}} \sum_{i_1, i_2, \ldots, i_m = 1}^{n} \mathbf{A}_{i_1 i_2 \ldots i_m} y_{i_1} y_{i_2} \ldots y_{i_m} \ . \tag{4.9}$$

where $y \in \{0,1\}^{s^2}$ is a vectorized form of S.

Observe that the objective in (4.9) is same as $\mathbf{A} \times_1 y^T \times_2 \ldots \times_m y^T$, and hence, the above problem is identical to (4.5), where one finds a single cluster $\mathcal{V}_1$ with $|\mathcal{V}_1| = s$ vertices that maximizes normalized associativity, *i.e*, $k = 1$. Duchenne et al. (2011) relaxed the problem to the space $y \in \mathbb{R}^{s^2}$, and proposed a tensor power iteration based algorithm to solve the relaxed optimization.

Optimization with ℓ_p-norm constraint. This approach for hypergraph partitioning is quite similar to the relaxation of (4.9) studied in (Duchenne et al., 2011), and has been used various applications such as molecular network alignment (Michoel and Nachtergaele, 2012), subspace clustering (Rota Bulo and Pelillo, 2013) and hypergraph matching (Liu et al., 2010),

Given the adjacency tensor $\mathbf{A}$ of a m-uniform hypergraph, the principle approach involves finding a cluster $\mathcal{V}_1 \subset \mathcal{V}$ by solving the problem

$$\underset{y \in \mathbb{R}^n : \|y\|_p = 1}{\text{maximize}} \sum_{i_1, i_2, \ldots, i_m = 1}^{n} \mathbf{A}_{i_1 i_2 \ldots i_m} y_{i_1}^{1/p} y_{i_2}^{1/p} \ldots y_{i_m}^{1/p} \ . \tag{4.10}$$

The cluster is estimated from the entries of the optimal y and removed from the hypergraph. This procedure is continued till all the clusters are found.

One class of methods (Rota Bulo and Pelillo, 2013; Liu et al., 2010) views y as a probability distribution, that leads to restriction of the search space to non-negative vectors with unit ℓ_1-norm constraint. On the other hand, a particular instance of the approach in (Michoel and

Nachtergaele, 2012)[1] is obtained by setting $p = m$ in (4.10).

For $p = m$, we draw connection of this optimization problem to (4.5) by choosing $\beta_1 = \ldots = \beta_m = \frac{1}{m}$ and $k = 1$. In this case, if $Y^{(r)} \in \mathbb{R}^{n\times 1}$ in (4.3) is defined using $|\mathcal{V}_1|$ instead of $\text{Vol}(\mathcal{V}_1)$, then (4.10) coincides with a relaxation of this TTM problem. On the other hand, the ℓ_1-norm version is similar to assuming $\beta_1 = \ldots = \beta_m = 1$. Alternatively, one may also view it as a ℓ_2-norm problem involving a $2m$-uniform hypergraph on $\mathcal{V}$ with the adjacency tensor $\mathbf{A}'$ of order $2m$ such that

$$\mathbf{A}'_{i_1 i_2 \ldots i_{2m}} = \mathbf{A}_{i_1 i_3 \ldots i_{2m-1}} \mathbb{1}\{i_\ell = i_{\ell+1} \forall \ell = 1, 3, .., m-1\}.$$

for all $i_1, i_2, \ldots, i_{2m} \in \mathcal{V}$. The ℓ_1-norm constraint gets modified to ℓ_2-norm by defining $y' \in \mathbb{R}^n$ with $y'_i = \sqrt{y_i}$ for $i = 1, \ldots, n$. This shows that (4.10) with ℓ_1-norm constraint is similar to the ℓ_2-norm problem (4.9). A further connection of these formulations to the tensor eigenvalue problem is established below.

4.2.2 Related problems in tensor literature

Due to the form of the matrix $\overline{Y}^{(r)}$ in (4.3), one can note that the optimization in (4.5) is essentially that of maximizing the trace of a tensor via orthogonal transformations. This problem has been previously studied in the signal processing community for blind source separation (Comon, 2001). One also solves a variant of this where the sum of squares of diagonal elements is maximized, and hence, one finds an approximate solution in the form of

$$\underset{\overline{Y}^{(1)},\ldots,\overline{Y}^{(m)}}{\text{maximize}} \sum_{\ell=1}^{k} \left(\mathbf{A} \times_1 \overline{Y}^{(1)^T} \times_2 \overline{Y}^{(2)^T} \times_3 \ldots \times_m \overline{Y}^{(m)^T} \right)^2_{\ell\ell\ldots\ell}, \tag{4.11}$$

One can argue that such an objective leads to an higher order SVD based approach (Definition 2.6). To this end, the formulation in (4.5) complements the discussions in Chapter 3 in the sense that both methods try to perform higher-order clustering by formulating the problem as two well-known tensor problems related to maximization of diagonal terms.

The normalized associativity maximization formulation (4.5) is also related to the tensor eigenvalue problem of Lim (2005) presented in (2.8), where the m^{th}-order tensor $\mathbf{A}$ is viewed as a m-linear functional. This connection follows from (4.9) and (4.10), where we showed that these approaches are closely related to (4.5). A more general result can be stated in this respect.

[1] Michoel and Nachtergaele (2012) proposed their method for non-uniform hypergraphs, where the exponent of y is $1/|e|$, and the unit ℓ_p-norm constraint uses a fixed parameter p.

Corollary 4.2. *Consider a special case of (4.5), where we impose the condition* $\overline{Y}^{(1)} = \ldots = \overline{Y}^{(m)} = Y$*, and let* $y_1, \ldots, y_k$ *be the columns of* $Y \in \mathbb{R}^{n \times k}$*. The objective in (4.5) simplifies as*

$$\text{Trace}\left(\mathbf{A} \times_1 Y^T \ldots \times_m Y^T\right) = \sum_{\ell=1}^{k} \mathbf{A} \times_1 y_\ell^T \times_2 y_\ell^T \times_3 \ldots \times_m y_\ell^T, \tag{4.12}$$

where each term in the sum is the normalized associativity of individual clusters.

Subsequently, if the problem (4.5) is relaxed with the only constraint being $\|y_\ell\|_p = 1$ *for* $\ell = 1, \ldots, k$*, then the stationary points of the optimization correspond to matrices* Y*, whose columns correspond to* ℓ_p*-eigenvectors of* $\mathbf{A}$*.*

4.3 A consistent spectral algorithm

A spectral relaxation of (4.5) is a two-fold procedure, where first we construct a matrix from the weighted adjacency tensor $\mathbf{A}$, and then relax the problem into a matrix spectral decomposition type objective. This principle is reminiscent of the classical technique for studying spectral properties of hypergraphs (Chung, 1992; Bolla, 1993), and is closely related to approach of clustering graph approximations of hypergraphs (Agarwal et al., 2006).

Algorithm TTM : Spectral relaxation of tensor trace maximization problem

Input: Affinity tensor $\mathbf{A}$ of the m-uniform hypergraph $(\mathcal{V}, \mathcal{E}, w)$ with $|\mathcal{V}| = n$.

1: Define the matrix $A \in \mathbb{R}^{n \times n}$ as $A_{ij} = \sum_{i_3, \ldots, i_m = 1}^{n} \mathbf{A}_{iji_3 \ldots i_m}$.

2: Let $D \in \mathbb{R}^{n \times n}$ be diagonal with $D_{ii} = \sum_{j=1}^{n} A_{ij}$.

3: Compute k dominant orthonormal eigenvectors of $D^{-1/2} A D^{-1/2}$, denoted by $X \in \mathbb{R}^{n \times k}$.

4: Normalize rows of X to have unit norm, and denote this matrix as $\overline{X}$.

5: Run k-means on the rows of $\overline{X}$.

Output: Partition of $\mathcal{V}$ that corresponds to the clusters obtained from k-means.

4.3.1 Consistency under planted partition model

In the rest of the section, we study the consistency of the above spectral algorithm under the planted partition model described in Section 3.3. Recall that if ψ and ψ' denote the true and output labels, the the clustering error is defined as in (2.20). The following result shows the consistency of Algorithm TTM.

Theorem 4.3. *Let $(\mathcal{V}, \mathcal{E}, w)$ be a m-uniform hypergraph on $|\mathcal{V}| = n$ vertices generated from a random model with k planted vertex classes, where n is sufficiently large. Define $d = \min_{1 \le i \le n} \mathsf{E}[\deg(i)]$ and, without loss of generality, assume that the cluster sizes are $n_1 \ge n_2 \ge \ldots \ge n_k$. Also assume that the algorithm uses the approximate k-means method (Ostrovsky et al., 2012).*

There exists an absolute constant $C > 0$ and a quantity δ (function of n), such that, if $\delta > 0$ and

$$d > \frac{Ckn_1(\ln n)^2}{n_k \delta^2}\ , \tag{4.13}$$

then with probability $(1 - o(1))$, the clustering error of TTM is

$$\text{Error}_{\text{TTM}}(\psi, \psi') = O\left(\frac{kn_1 \ln n}{\delta^2 d}\right) = o(n). \tag{4.14}$$

Note that d obviously grows with n though this dependence is not made explicit in the notation. We also allow k to vary with n. The lower bound of d in (4.13) ensures that the hypergraph is sufficiently dense so that the following three conditions hold, respectively: (i) the matrix A computed in Algorithm TTM concentrates near its expectation, (ii) the k dominant eigenvectors of $D^{-1/2}AD^{-1/2}$ contain information about the partition, and (iii) the k-means step provides a near optimal solution. The result proves a weak consistency of the TTM algorithm. However, we show later in Section 4.3.2, stronger results can be achieved in certain cases.

4.3.2 A Special Case

To gain insights into the implications of Theorem 4.3, we consider the following special case of the planted partition model. The partition ψ is defined such that the k clusters are of equal size. Moreover, the tensor $\mathbf{B}^{(m)}$ in (3.6) is given by $\mathbf{B}^{(m)}_{j_1 j_2 \ldots j_m} = (p + q)$ if $j_1 = j_2 = \ldots = j_m$, and q otherwise, where $p, q \in [0, 1]$ with $q \le (1 - p)$. Thus, in this model, edges residing within each cluster have a high weight (in the expected sense) as compared to other edges[1]. We state the following consistency result for dense hypergraphs.

Corollary 4.4. *Let $\alpha_m = 1$ and $k = O\left(\frac{n^{1/4}}{\ln n}\right)$. Then with probability $(1 - o(1))$,*

$$\text{Error}_{\text{TTM}}(\psi, \psi') = O\left(\frac{n^{(3-m)/2}}{(\ln n)^{2m-3}}\right)\ . \tag{4.15}$$

[1] The model considered here may be viewed as the four parameter stochastic block model (Rohe et al., 2011) defined by the parameters (n, k, p_n, q_n), where n vertices are divided into k parts of equal size. Edges within a cluster occur with probability $(p_n + q_n)$, while inter-cluster edges occur with probability q_n. Here, we set $p_n = \alpha_m p$ and $q_n = \alpha_m q$ for some constants $p, q \in [0, 1]$ with $q \le (1 - p)$.

According to the notions of consistency defined in (Mossel et al., 2013b), it can be seen that for $m = 2$, Algorithm TTM is weakly consistent, *i.e*, $\text{Error}_{\text{TTM}}(\psi, \psi') = o(n)$. We note here that, in this sense, the algorithm is not worse than spectral clustering that is also known to be weakly consistent (Rohe et al., 2011). However, for $m \geq 3$, $\text{Error}_{\text{TTM}}(\psi, \psi') = o(1)$ for Algorithm TTM, which implies that it is strongly consistent in this case. In other words, the algorithm can exactly recover the partition for large n. Intuitively, this conclusion seems appropriate since in this case, uniform hypergraphs for large m have a large number of edges that provides 'more' information about the partition. Hence, a faster decay rate for the error should be expected.

In the sparse regime, the question one is interested in is the minimum level of sparsity under which weak consistency of an algorithm can be proved. The following result answers this question.

Corollary 4.5. *Let $k = O(\ln n)$. There exists an absolute constant $C > 0$, such that, if*

$$\alpha_m \geq \frac{C(\ln n)^{2m+1}}{n^{m-1}}\,, \tag{4.16}$$

then $\text{Error}_{\text{TTM}}(\psi, \psi') = O\left(\frac{n}{(\ln n)^2}\right) = o(n)$ *with probability* $(1 - o(1))$.

In the case of graphs ($m = 2$), Lei and Rinaldo (2015) showed that weak consistency is achieved by spectral clustering for $\alpha_m \geq \frac{C \ln n}{n}$. It is important to note that our bound is worse by logarithm factors, but at the same time, Corollary 4.5 makes no assumption about the performance of k-means.

We also comment on the theoretical performance of TTM in comparison with the Algorithm HOSVD. In particular, we focus on the settings of Corollaries 4.4 and 4.5, and argue that in both cases, TTM is provably better than HOSVD.

Corollary 4.6. *Under the setting of Corollary 4.4, the error bound for the HOSVD algorithm is*

$$\text{Error}_{\text{HOSVD}}(\psi, \psi') = O\left(\frac{n^{(4-m)/2}}{(\ln n)^{2m-1}}\right)\,.$$

Thus TTM has a smaller error bound than HOSVD.

Similarly, in the case of Corollary 4.5, the lower bound on sparsity for HOSVD is

$$\alpha_m \geq \frac{C'(\ln n)^{m+1.5}}{n^{(m-1)/2}}\,,$$

which is larger than the allowable sparsity for TTM.

4.3.3 Proof of Theorem 4.3

Here, we give an outline of the proof of Theorem 4.3 using a series of technical lemmas. The proofs of these results are given in Appendix 4.A. The proof has a modular structure which consists of (i) deriving certain conditions on the model parameters such that Algorithm TTM incurs no error in the expected case, (ii) subsequent use of matrix concentration inequalities and spectral perturbation bounds to claim that (almost surely) the dominant eigenvectors in the random case do not deviate much from the expected case, and (iii) finally, the proof of correctness of the k-means step.

To analyze the algorithm in the expected case, let $\mathcal{A} = \mathsf{E}[A]$ and $\mathcal{D} = \mathsf{E}[D]$, where A and D are the matrices computed in Algorithm TTM. Observe that if the expected affinity tensor is input to the system, then $\mathcal{A}$ corresponds to the matrix computed in the first step of the algorithm, and $\mathcal{D}_{ii} = \sum_{j=1}^{n} \mathcal{A}_{ij}$. From the definition of the model, it can be seen that $\mathcal{A}_{ii} = 0$ for all i, and for $i \neq j$,

$$\mathcal{A}_{ij} = (m-2)! \sum_{\substack{i_3 < i_4 < \ldots < i_m, \\ i,j \notin \{i_3,\ldots,i_m\}}} \alpha_m \mathbf{B}^{(m)}_{\psi_i \psi_j \psi_{i_3} \ldots \psi_{i_m}} , \tag{4.17}$$

where the factor $(m-2)!$ takes into account all permutations of $\{i_3, \ldots, i_m\}$. The key observation here is that $\mathcal{A}_{ij} = \mathcal{A}_{i'j'}$ whenever $\psi_i = \psi_{i'}$ and $\psi_j = \psi_{j'}$, which holds since, under the present model, vertices in the same cluster are statistically identical. Thus, one can define a matrix $G \in \mathbb{R}^{k \times k}$ such that $\mathcal{A}_{ij} = G_{\psi_i \psi_j}$ for all $i \neq j$. This implies that, ignoring the diagonal entries, $\mathcal{A}$ is essentially of rank k.

Let $Z \in \{0,1\}^{n \times k}$ be the assignment matrix corresponding to partition ψ, *i.e*, $Z_{ij} = \mathbb{1}\{i \in \psi^{-1}(j)\}$, and let the sizes of the k clusters be $n_1 \geq n_2 \geq \ldots \geq n_k$. We define

$$\delta = \lambda_k(G) \min_{1 \leq i \leq n} \frac{n_{\psi_i}}{\mathcal{D}_{ii}} - \max_{1 \leq i,j \leq n} \left| \frac{G_{\psi_i \psi_i}}{\mathcal{D}_{ii}} - \frac{G_{\psi_j \psi_j}}{\mathcal{D}_{jj}} \right| , \tag{4.18}$$

where $\lambda_k(G)$ is the smallest eigenvalue of G. Also define $\mathcal{D}_{\min} = \min_{1 \leq i \leq n} \mathcal{D}_{ii}$. One can argue that $\mathcal{D}_{\min} = (m-1)!d$. Hence, for the subsequent analysis as well as for all the proofs, it is more convenient to state (4.13) as

$$\mathcal{D}_{\min} > \frac{Ckn_1(\ln n)^2}{n_k \delta^2} , \tag{4.19}$$

where the constant C takes into account the additional factor of $(m-1)!$. Similarly, one can replace d by $\mathcal{D}_{\min}$ in the error bound (4.14). We now state the following result, which is proved

in Appendix 4.A.

Lemma 4.7. *If δ in (4.18) satisfies $\delta > 0$, then there exists an orthonormal matrix $U \in \mathbb{R}^{k\times k}$ such that the k dominant orthonormal eigenvectors of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$ correspond to the columns of the matrix $\mathcal{X} = Z(Z^TZ)^{-1/2}U$.*

It is easy to see that $\mathcal{X}$ has k distinct rows, each corresponding to a true cluster. Hence, clustering the rows of $\mathcal{X}$ (or its row normalized form) using k-means results in an accurate clustering of the vertices. The subsequent results show that the eigenvector matrix X computed from a random realization of the hypergraph is close to $\mathcal{X}$ almost surely, and hence, one can expect a good clustering even in that case.

Lemma 4.8 proves a concentration bound for the normalized affinity matrix L computed in Algorithm TTM. The proof, given in the appendix, relies on an useful characterization of the matrix A. To describe this representation, we define for each edge $e \in \mathcal{E}$, a matrix $R_e \in \{0,1\}^{n\times n}$ as $(R_e)_{ij} = 1$ if $i, j \in e, i \neq j$, and zero otherwise. Quite similar to the representation of (4.17), one can note that

$$A = (m-2)!\sum_{e\in\mathcal{E}} w_e R_e \,. \tag{4.20}$$

This characterization is quite useful since the independence of $(w_e)_{e\in\mathcal{E}}$ ensures that A is represented as a sum of independent random matrices, and hence, one can use matrix concentration inequalities (Tropp, 2012) to derive a tail bound for $\|A - \mathcal{A}\|_2$.

Lemma 4.8. *If $\mathcal{D}_{\min} > 9(m-1)!\ln n$, then with probability $(1 - O(n^{-2}))$*

$$\|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2 \leq 12\sqrt{\frac{(m-1)!\ln n}{\mathcal{D}_{\min}}}\,. \tag{4.21}$$

Notice the similarity of the above result with Lemma 3.9 stated for the HOSVD algorithm, where we had $\mathcal{A}_{\min}$ instead of $\mathcal{D}_{\min}$. As a consequence, the subsequent analysis of HOSVD presented in Lemmas 3.10–3.12 can be directly borrowed to prove Theorem 4.3.

4.A Proofs for results in this chapter

Proof of Proposition 4.1

Observe that the claim follows if we show that

$$\frac{\mathrm{Assoc}(\mathcal{V}_\ell)}{\mathrm{Vol}(\mathcal{V}_\ell)} = \frac{1}{(m-1)!}\left(\mathbf{A}\times_1 \overline{Y}^{(1)^T} \times_2 \overline{Y}^{(2)^T} \times_3 \ldots \times_m \overline{Y}^{(m)^T}\right)_{\ell\ell\ldots\ell}\,.$$

This holds since

$$\left(\mathbf{A} \times_1 \overline{Y}^{(1)^T} \times_2 \overline{Y}^{(2)^T} \times_3 \ldots \times_m \overline{Y}^{(m)^T}\right)_{\ell\ell\ldots\ell} = \sum_{i_1,i_2,\ldots,i_m=1}^{n} \mathbf{A}_{i_1 i_2 \ldots i_m} \overline{Y}^{(1)}_{i_1\ell} \overline{Y}^{(2)}_{i_2\ell} \ldots \overline{Y}^{(m)}_{i_m\ell}$$
$$= \sum_{i_1,i_2,\ldots,i_m=1}^{n} \mathbf{A}_{i_1 i_2 \ldots i_m} \frac{\mathbb{1}\{i_1 \in \mathcal{V}_\ell, \ldots, i_m \in \mathcal{V}_\ell\}}{\mathrm{Vol}(\mathcal{V}_\ell)}$$

as $\sum_r \beta_r = 1$, and taking power of indicator does not change the function. Note that in the above summation, the terms with repeated indices are zero, while the other terms correspond to each edge $e = \{i_1, \ldots, i_m\}$ and have $m!$ identical copies. Thus, we can write

$$\left(\mathbf{A} \times_1 \overline{Y}^{(1)^T} \times_2 \overline{Y}^{(2)^T} \times_3 \ldots \times_m \overline{Y}^{(m)^T}\right)_{\ell\ell\ldots\ell} = m! \sum_{e\in\mathcal{E}} w_e \frac{\mathbb{1}\{e \subset \mathcal{V}_\ell\}}{\mathrm{Vol}(\mathcal{V}_\ell)}$$
$$= (m-1)! \frac{\mathrm{Assoc}(\mathcal{V}_\ell)}{\mathrm{Vol}(\mathcal{V}_\ell)} \,,$$

which completes the proof.

Proof of Corollary 4.2

The relation in (4.12) follows from the definition of mode-k product using computations similar to the above proof. For the second claim, observe that if the orthogonality constraint on $y_1, \ldots, y_k$ is not imposed, then one could separately maximize each term of the summation. Hence, the claim follows from the definition of ℓ_p-eigenvectors (see (2.8)).

Proof of Corollary 4.4

We begin by computing $\mathcal{A}$ as defined in (4.17)

$$\mathcal{A}_{ij} = \begin{cases} (m-2)!\alpha_m \left(p\binom{\frac{n}{k}-2}{m-2} + q\binom{n-2}{m-2}\right) & \text{if } i \neq j, \psi_i = \psi_j \\ (m-2)!\alpha_m q\binom{n-2}{m-2} & \text{if } i \neq j, \psi_i \neq \psi_j \\ 0 & \text{if } i = j. \end{cases}$$

From the definition of G, $\mathcal{D}_{\min}$ and δ, one can compute that

$$\mathcal{D}_{\min} = (m-1)!\alpha_m \left(p\binom{\frac{n}{k}-1}{m-1} + q\binom{n-1}{m-1} \right) , \tag{4.22}$$

and

$$\delta = \lambda_k(G)\frac{n}{k\mathcal{D}_{\min}} = \frac{(m-2)!\alpha_m pn}{k\mathcal{D}_{\min}}\binom{\frac{n}{k}-2}{m-2} \tag{4.23}$$

We need to validate that the conditions in (4.13), or equivalently (4.19), are satisfied. Given $\alpha_m = 1$, one can see that $\mathcal{D}_{\min} = \Theta(n^{m-1})$. Also

$$\delta^2\mathcal{D}_{\min} = \Theta\left(\left(\frac{n}{k}\right)^{2m-2}\frac{1}{\mathcal{D}_{\min}}\right) = \Theta\left(\frac{n^{m-1}}{k^{2m-2}}\right) = \Omega\left(n^{(m-1)/2}(\ln n)^{2m-2}\right) ,$$

taking into account that $k = O\left(\frac{n^{1/4}}{\ln n}\right)$. Thus, the condition in (4.19) holds for large n and for all $m \geq 2$. Subsequently, one can applying the bound in (4.14) to claim the result.

Proof of Corollary 4.5

Assume the condition $k = O(\ln n)$ as stated. Then one can see that (4.19) holds if

$$\mathcal{D}_{\min} = \Omega\left(\frac{(\ln n)^3}{\delta^2}\right) .$$

From (4.22) and (4.23), we have $\mathcal{D}_{\min} = \Theta(\alpha_m n^{m-1})$ and $\delta^2\mathcal{D}_{\min} = \Omega\left(\alpha_m n^{m-1}(\ln n)^{2-2m}\right)$. Hence, choosing $\alpha_m \geq \frac{C(\ln n)^{2m+1}}{n^{m-1}}$ for sufficiently large C ensures that (4.19) is satisfied. Subsequently with probability $\left(1 - O\left(n^{-2} + (\ln n)^{-1/4}\right)\right) = (1 - o(1))$, we obtain an error bound

$$\mathrm{Error}_{\mathrm{TTM}}(\psi, \psi') = O\left(\frac{n\ln n}{\delta^2\mathcal{D}_{\min}}\right) = O\left(\frac{n\ln n}{\alpha_m n^{m-1}(\ln n)^{2-2m}}\right) = O\left(\frac{n}{(\ln n)^2}\right) = o(n) ,$$

which completes the proof.

Proof of Corollary 4.6

From the balanced k-partition model described in the section, one can verify that in the case of HOSVD, $\mathcal{A}_{\min} = \Theta(\alpha_m^2 n^{m-1})$ and $\delta = \Theta(k^{-m})$.

Let us first look at the setting of Corollary 4.4 with $\alpha_m = 1$ and $k = O\left(\frac{n^{1/4}}{\ln n}\right)$. Then the

right hand side of (3.11) is

$$\frac{Ck(\ln n)^2}{\delta^2} = \Theta(k^{2m+1}(\ln n)^2) = o(n^{m-1})$$

and hence, eventually must be smaller than $\mathcal{A}_{\min}$. So, the condition of Theorem 3.7 is satisfied and the error bound is

$$\text{Error}_{\text{HOSVD}}(\psi, \psi') = O\left(\frac{n \ln n}{\delta^2 \mathcal{A}_{\min}}\right) = O\left(\frac{k^{2m} \ln n}{n^{m-2}}\right) ,$$

which simplifies to the stated error bound.

In the sparse case, *i.e*, setting of Corollary 4.5, the right hand side of (3.11) is $\Theta((\ln n)^{2m+3})$, while $\mathcal{A}_{\min} = \Theta(\alpha_m^2 n^{m-1})$. Thus, the condition (3.11) holds only if the stated lower bound on α_m is satisfied.

Proof of Lemma 4.7

From the discussions following (4.17), one can see that the matrix $\mathcal{A}$ may be expressed as

$$\mathcal{A} = ZGZ^T - J ,$$

where $J \in \mathbb{R}^{n \times n}$ is diagonal with $J_{ii} = G_{\psi_i \psi_i}$. From the proof of Lemma 3.8, one can see that there is a matrix $\mathcal{G} \in \mathbb{R}^{k \times k}$ with eigen decomposition $\mathcal{G} = U \Lambda_1 U^T$ such that

$$\mathcal{D}^{-1/2} \mathcal{A} \mathcal{D}^{-1/2} \mathcal{X} = \mathcal{X} \Lambda_1,$$

where $\mathcal{X} = Z(Z^T Z)^{-1/2} U$, and it satisfies $\mathcal{X}^T \mathcal{X} = I$. Thus, the columns of $\mathcal{X}$ are orthonormal eigenvectors of $\mathcal{D}^{-1/2} \mathcal{A} \mathcal{D}^{-1/2}$ corresponding to the eigenvalues in Λ_1. The other $(n-k)$ orthonormal eigenvectors correspond to eigenvalues from the set $\left\{-\frac{J_{ii}}{\mathcal{D}_{ii}} : 1 \le i \le n\right\}$. The claim follows by proceeding along the lines of the proof of Lemma 3.8.

Proof of Lemma 4.8

We begin the proof with the claims that if $\mathcal{D}_{\min} > 9(m-1)! \ln n$, then

$$\mathsf{P}\left(\max_{1 \le i \le n} \left|\frac{D_{ii}}{\mathcal{D}_{ii}} - 1\right| > 3\sqrt{\frac{(m-1)! \ln n}{\mathcal{D}_{\min}}}\right) \le \frac{2}{n^2} , \tag{4.24}$$

and

$$\mathsf{P}\left(\|\mathcal{D}^{-1/2}(A-\mathcal{A})\mathcal{D}^{-1/2}\|_2 > 3\sqrt{\frac{(m-1)!\ln n}{\mathcal{D}_{\min}}}\right) \leq \frac{2}{n^2}\,. \tag{4.25}$$

We now bound $\|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2$ using arguments as in Lemma 3.9, and can write

$$\begin{aligned}&\|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2\\ &\leq \max_{1\leq i\leq n}\left|\frac{D_{ii}}{\mathcal{D}_{ii}}-1\right|\left(2+\max_{1\leq i\leq n}\left|\frac{D_{ii}}{\mathcal{D}_{ii}}-1\right|\right) + \|\mathcal{D}^{-1/2}A\mathcal{D}^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2.\end{aligned} \tag{4.26}$$

Using the bounds in (4.24) and (4.25) along with the fact that $3\sqrt{\frac{(m-1)!\ln n}{\mathcal{D}_{\min}}} < 1$, one arrives at the claim.

We now prove the concentration bound in (4.24). Observe that

$$D_{ii} = \sum_{j=1}^{n} A_{ij} = \sum_{i_2,\ldots,i_m=1}^{n} \mathbf{A}_{ii_2\ldots i_m} = (m-1)!\sum_{e\in\mathcal{E}:e\ni i} w_e\,,$$

where the last equality holds since the summation over all $i_2,\ldots,i_m$ counts each edge containing the i^{th} vertex $(m-1)!$ times. Since, D_{ii} is a sum of independent random variables, we can use Bernstein inequality to obtain for any $t>0$,

$$\begin{aligned}\mathsf{P}\left(|D_{ii}-\mathcal{D}_{ii}| > t\mathcal{D}_{ii}\right) &= \mathsf{P}\left(\left|\sum_{e\in\mathcal{E}:e\ni i} w_e - \mathsf{E}[w_e]\right| > \frac{t\mathcal{D}_{ii}}{(m-1)!}\right)\\ &\leq 2\exp\left(\frac{-\left(\frac{t\mathcal{D}_{ii}}{(m-1)!}\right)^2}{2\sum\limits_{e\in\mathcal{E}:e\ni i}\mathsf{Var}(w_e) + \frac{2}{3}\frac{t\mathcal{D}_{ii}}{(m-1)!}}\right)\,.\end{aligned} \tag{4.27}$$

Since $w_e\in[0,1]$, we have

$$\sum_{e\in\mathcal{E}:e\ni i}\mathsf{Var}(w_e) \leq \sum_{e\in\mathcal{E}:e\ni i}\mathsf{E}[w_e] = \frac{\mathcal{D}_{ii}}{(m-1)!}\,.$$

Substituting this in (4.27), we have

$$\mathsf{P}\left(|D_{ii}-\mathcal{D}_{ii}| > t\mathcal{D}_{ii}\right) \leq 2\exp\left(-\frac{t^2\mathcal{D}_{ii}}{3(m-1)!}\right) \leq 2\exp\left(-\frac{t^2\mathcal{D}_{\min}}{3(m-1)!}\right).$$

The bound in (4.24) follows from above by setting $t = 3\sqrt{\frac{(m-1)!\ln n}{\mathcal{D}_{\min}}}$, and using a union bound over all $i = 1, \ldots, n$.

Finally, we derive (4.25) using a matrix version of the Bernstein inequality (see Theorem 2.11). Owing to the representation in (4.20), one can write

$$\mathcal{D}^{-1/2}(A - \mathcal{A})\mathcal{D}^{-1/2} = \sum_{e \in \mathcal{E}} (m-2)! \, (w_e - \mathsf{E}[w_e]) \, \mathcal{D}^{-1/2} R_e \mathcal{D}^{-1/2}$$

as a sum of independent, zero mean random matrices. One can verify that $\|R_e\|_2 \leq (m-1)$, and hence,

$$\left\| (m-2)! \, (w_e - \mathsf{E}[w_e]) \, \mathcal{D}^{-1/2} R_e \mathcal{D}^{-1/2} \right\|_2 \leq \frac{(m-1)!}{\mathcal{D}_{\min}} \, .$$

In addition, one can bound

$$\begin{aligned}
&\left\| \sum_{e \in \mathcal{E}} \mathsf{E}\left[\left((m-2)! \, (w_e - \mathsf{E}[w_e]) \, \mathcal{D}^{-1/2} R_e \mathcal{D}^{-1/2} \right)^2 \right] \right\|_2 \\
&= ((m-2)!)^2 \left\| \sum_{e \in \mathcal{E}} \mathsf{Var}(w_e) \mathcal{D}^{-1/2} R_e \mathcal{D}^{-1} R_e \mathcal{D}^{-1/2} \right\|_2 \\
&= ((m-2)!)^2 \left\| \sum_{e \in \mathcal{E}} \mathsf{Var}(w_e) \left(\mathcal{D}^{-1} R_e \right)^2 \right\|_2 \\
&\leq ((m-2)!)^2 \max_{1 \leq i \leq n} \sum_{j=1}^{n} \sum_{e \in \mathcal{E}} \mathsf{Var}(w_e) \left(\left(\mathcal{D}^{-1} R_e \right)^2 \right)_{ij} \\
&\leq \frac{((m-2)!)^2}{\mathcal{D}_{\min}} \max_{1 \leq i \leq n} \frac{1}{\mathcal{D}_{ii}} \sum_{e \in \mathcal{E}} \mathsf{E}(w_e) \sum_{j=1}^{n} \left(R_e^2 \right)_{ij} \, .
\end{aligned}$$

Here, the first inequality holds due to Gerschgorin's theorem. Observing that the row sum of R_e^2 is at most $(m-1)^2$, the expression can be simplified to show that the quantity is bounded from above by $\frac{(m-1)!}{\mathcal{D}_{\min}}$. Setting $t = 3\sqrt{\frac{(m-1)!\ln n}{\mathcal{D}_{\min}}}$ in Theorem 2.11, and combining above arguments, one arrives at (4.25). This completes the proof.

> Whence, I often asked myself, did the principle of life proceed? It was a bold question and one which has ever been considered as a mystery; yet with how many things are we upon the brink of becoming acquainted, if cowardice or carelessness did not restrain our inquiries.
>
> Mary Shelley, *Frankenstein*

Chapter 5

Partitioning Non-uniform Hypergraphs

Non-uniform hypergraphs do not come with simple representations in terms of symmetric adjacency matrices or tensors. The matrix that arises naturally in this case is the incidence matrix $H \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{E}|}$, where $H_{ie} = 1$ if the vertex i is contained in the edge e, and 0 otherwise. One can note that the degree of any vertex i can be written as $\deg(i) = \sum_{e\in\mathcal{E}} H_{ie}$. Similarly, the cardinality of any edge e is $|e| = \sum_{i\in\mathcal{V}} H_{ie}$.

We start our discussions in Section 5.1 by extending the planted partition model to non-uniform hypergraphs. Note here that the incidence matrix does not provide information about edge weights. But Section 5.1 discusses an alternative representation that allows the possiblity of edge weights. However, for simplicity, we restrict the subsequent analysis in this chapter to unweighted non-uniform hypergraphs $(\mathcal{V}, \mathcal{E})$. We then present two approaches for non-uniform hypergraph partitioning in Section 5.2. The consistency results are stated and proved in Section 5.3, while its consequences under specific examples of the planted model are studied in Section 5.4. The technical lemmas and corollaries in this chapter are proved in Appendix 5.A.

5.1 Planted partition in non-uniform hypergraphs

Let the set of vertices be $\mathcal{V} = \{1, 2, \dots n\}$, and let $\psi : \{1, 2, \dots, n\} \to \{1, 2, \dots, k\}$ be a partition of the vertices into k classes. A direct approach to define a model for planted non-uniform hypergraphs would be in terms of a model for the random incidence matrix. This turns out to be quite tricky as a block structure cannot be immediately seen from H. Instead, we observe the following.

Remark 5.1. Let $M \geq 2$ be an integer, representing the range or the maximum edge cardinality in the hypergraph $(\mathcal{V}, \mathcal{E})$. For each $m = 2, \dots, M$, define $\mathcal{E}_m = \{e \in \mathcal{E} : |e| = m\}$. Then

$\mathcal{E} = \bigcup_{m=2}^{M} \mathcal{E}_m$, and one may view $(\mathcal{V}, \mathcal{E})$ as a collection of m-uniform hypergraphs $(\mathcal{V}, \mathcal{E}_m)$ for $m = 2, \ldots, M$.

We assume that there is no edge of size 0 or 1, which are not useful in a partitioning problem. Remark 5.1 implies that it is possible to construct planted models for non-uniform hypergraphs from a collection of uniform hypergraph models. Based on this observation, we consider the planted partition model described below. In view of practical situations, we allow both k and M to vary with n though this dependence is not made explicit in the notation. One may set $M = n$ to allow occurrence of all possible edges, but in practice, one often finds that $M = O(\ln n)$ (see Section 8.1).

For every n and for each $m = 2, \ldots, M$, let $\alpha_m \in [0, 1]$, and let $\mathbf{B}^{(m)} \in [0, 1]^{k \times k \times \ldots \times k}$ be a symmetric tensor of order m. A random hypergraph on $\mathcal{V}$ is generated as follows. For each $m = 2, \ldots, M$, and for every set $\{i_1, i_2, \ldots, i_m\} \subset \mathcal{V}$, an edge is included independently with probability $\alpha_m \mathbf{B}^{(m)}_{\psi_{i_1} \psi_{i_2} \ldots \psi_{i_m}}$. This process generates a random hypergraph of maximum edge cardinality M. The tensor $\mathbf{B}^{(m)}$ contains the probabilities of forming m-way edges among the different classes if $\alpha_m = 1$. On the other hand, α_m allows for a sparsity scaling that does not depend on the partition. In real-world non-uniform hypergraphs, one often finds that the density of 2 or 3-way edges is much more than edges of larger size (say, 10). To account for this generality, we allow α_m to vary both with m and n[1]. For instance, if $\alpha_2 = 1$ and $\alpha_m = \frac{1}{n^{m-1}}$ for all $m > 2$, then the generated hypergraph contains $O(n^2)$ number of 2-way edges, but only $O(n)$ number of m-way edges for every $m > 2$.

As a special case, note that for graphs, $M = 2$ for all n, and the model corresponds to the sparse stochastic block model, where an edge $\{i, j\}$ is formed with probability $\alpha_2 \mathbf{B}^{(2)}_{\psi_i \psi_j}$. In other words, if $Z \in \{0, 1\}^{n \times k}$ denotes the assignment matrix, then the probability of edge $\{i, j\}$ is same as the corresponding entry of $\alpha_2 Z \mathbf{B}^{(2)} Z^T$. For m-uniform uniform hypergraphs, one has $\alpha_r = 0$ for all $r \neq m$. Thus, the above model specifies a hypergraph of range M as a collection of uniform hypergraphs of orders $m = 2, \ldots, M$. Each m-uniform hypergraph is specified in terms of α_m and $\mathbf{B}^{(m)}$. Following the lines of Chapters 3 and 4, one can easily extend this model to weighted hypergraphs.

5.2 Spectral algorithms for non-uniform hypergraphs

We now turn our focus to hypergraph partitioning. It would be quite natural to continue our discussions from Chapter 4, where we partition a hypergraph by maximizing the normalized

[1] The dependence on n is not made explicit in the notation.

hypergraph associativity (4.1).

5.2.1 Normalized hypergraph associativity maximization

Viewing a non-uniform hypergraphs as a collection of uniform hypergraphs motivates one to extend TTM. We present this extended algorithm, listed in Algorithm TTM-ext, without tensorial terminology and in terms of the incidence matrix H. However, Proposition 5.2 shows this method indeed solves a relaxation of normalized associativity maximization problem (4.5).

Algorithm TTM-ext : Extension of TTM to non-uniform hypergraphs

---

Input: Incidence matrix H of a hypergraph $(\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$.

1: Let $D \in \mathbb{R}^{n\times n}, \Delta^{|\mathcal{E}|\times|\mathcal{E}|}$ be diagonal with $D_{ii} = \sum_{\ell=1}^{|\mathcal{E}|} H_{i\ell}$, and $\Delta_{\ell\ell} = \sum_{i=1}^{n} H_{i\ell}$.

2: Define the matrix $A \in \mathbb{R}^{n\times n}$ as

$$A_{ij} = \begin{cases} \left(H(\Delta - I)^{-1}H^T\right)_{ij} & \text{if } i \neq j, \\ 0 & \text{if } i = j. \end{cases} \tag{5.1}$$

3: Compute k dominant orthonormal eigenvectors of $D^{-1/2}AD^{-1/2}$, denoted by $X \in \mathbb{R}^{n\times k}$.

4: Normalize rows of X to have unit norm, and denote this matrix as $\overline{X}$.

5: Run k-means on the rows of $\overline{X}$.

Output: Partition of $\mathcal{V}$ that corresponds to the clusters obtained from k-means.

Proposition 5.2. *For a non-uniform hypergraph $(\mathcal{V}, \mathcal{E})$, Algorithm TTM-ext is equivalent to either of the following two procedures:*

1. **(Spectral relaxation of trace maximization)**
   Let $\mathbf{A}^{(m)}$, $m = 2, \ldots, M$, denote the adjacency tensors for the m-uniform components of $(\mathcal{V}, \mathcal{E})$ described in Remark 5.1. Then Algorithm TTM-ext is a standard spectral relaxation (see (4.6)) of the following optimization:

$$\underset{\hat{\mathcal{V}}_1,\ldots,\hat{\mathcal{V}}_k}{\text{maximize}} \sum_{m=2}^{M} \frac{1}{(m-1)!} \text{Trace}\left(\mathbf{A}^{(m)} \times_1 \overline{Y}^{(1)^T} \times_2 \overline{Y}^{(2)^T} \times_3 \ldots \times_m \overline{Y}^{(m)^T}\right), \tag{5.2}$$

   where $\overline{Y}^{(r)} \in \mathbb{R}^{n\times k}$ is given by $\overline{Y}^{(r)}_{i\ell} = \left(\frac{\mathbb{1}\{i \in \mathcal{V}_\ell\}}{\text{Vol}(\mathcal{V}_\ell)}\right)^{\beta_r}$ with $\beta_1 = \beta_2 = \frac{1}{2}$ and $\beta_r = 0$ for all $r \geq 3$.

2. **(Spectral partitioning of clique expansion)**
   Ignoring the edge scaling due to $(\Delta - I)^{-1}$, the matrix A computed in (5.1) is identical to

the adjacency matrix of the clique expansion defined in (Rodríguez, 2002). Thus, Algorithm TTM-ext corresponds to normalized spectral clustering on the clique expansion of a hypergraph.

Note that the objective in (5.2) is same as the the normalized hypergraph associativity, NH-Assoc$(\mathcal{V}_1, \ldots, \mathcal{V}_k)$ (4.1), when there are edges of varying size. Thus, the above result shows that TTM-ext is a spectral relaxation of the associativity maximization problem. It also indicates that from the perspective of normalized associativity maximization, simply expanding each edge into a clique (Rodríguez, 2002) may not be sufficient, and one should put additional weights for the constructed edges.

5.2.2 Normalized hypergraph cut minimization

Till now, we have only dealt with the associativity maximization problem. We now discuss extensions of the normalized cut minimization problem to the case of hypergraphs. While it was argued in Chapter 2 that both problems are equivalent in the case of graphs, their extensions to hypergraphs do not result in identical formulations.

Several notions of hypergraph cut and hypergraph Laplacian have been proposed in the literature for both uniform (Hu and Qi, 2012) as well as non-uniform hypergraphs (Bolla, 1993; Rodríguez, 2002). We consider the generalization studied in (Bolla, 1993; Zhou et al., 2007), where the boundary of any set $\mathcal{V}_1 \subset \mathcal{V}$ is defined as $\partial\mathcal{V}_1 = \{e \in \mathcal{E} : e \cap \mathcal{V}_1 \neq \phi, e \cap \mathcal{V}_1^c \neq \phi\}$. Observe that, as in the case of graphs (see Section 2.2), the boundary denotes the set of edges that are cut when the vertices are divided into $\mathcal{V}_1$ and $\mathcal{V}_1^c = \mathcal{V} \backslash \mathcal{V}_1$. Subsequently, the cut is defined as

$$\text{Cut}(\mathcal{V}_1) = \sum_{e \in \partial\mathcal{V}_1} \frac{|e \cap \mathcal{V}_1||e \cap \mathcal{V}_1^c|}{|e|} \, .$$

Remark 5.3. In a weighted hypergraph $(\mathcal{V}, \mathcal{E}, w)$, one may define the cut for a set $\mathcal{V}_1$ as

$$\text{Cut}(\mathcal{V}_1) = \sum_{e \in \mathcal{E}} w_e \frac{|e \cap \mathcal{V}_1||e \cap \mathcal{V}_1^c|}{|e|} \, , \tag{5.3}$$

where we could replace $\partial\mathcal{V}_1$ by $\mathcal{E}$ since the edges outside $\partial\mathcal{V}_1$ do not contribute in the summation. This representation (5.3) also forms the basis for the definition of associativity used in Chapter 4. Recall that in graphs, the definitions of Cut$(\mathcal{V}_1)$ and Assoc$(\mathcal{V}_1)$ have an implicit similarity that while the former is the total edge weight between $\mathcal{V}_1$ and $\mathcal{V}_1^c$, the latter corrsponds to the total

edge weight between $\mathcal{V}_1$ and itself. Similarly, if $\mathcal{V}_1^c$ is replaced by $\mathcal{V}_1$ in (5.3), then one retrieves the definition of associativity.

We consider the problem of partitioning the vertex set $\mathcal{V}$ into k disjoint sets, $\mathcal{V}_1, \ldots, \mathcal{V}_k$, that minimizes the normalized hypergraph cut

$$\text{NH-Cut}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \sum_{j=1}^{k} \frac{\text{Cut}(\mathcal{V}_j)}{\text{Vol}(\mathcal{V}_j)} \,. \tag{5.4}$$

One can observe that for graphs, the above definition (5.4) retrieves the standard notion of a normalized cut in the case of graphs. Zhou et al. (2007) also define the notion of a normalized hypergraph Laplacian matrix $L \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ given by

$$L = I - D^{-1/2} H \Delta^{-1} H^T D^{-1/2}, \tag{5.5}$$

where the matrices $D \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}, \Delta \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ are diagonal with $D_{vv} = \deg(v)$ and $\Delta_{ee} = |e|$. The importance of (5.5) stems from the following fact.

Proposition 5.4. *The problem of minimizing* NH-cut$(\mathcal{V}_1, \ldots, \mathcal{V}_k)$ *is equivalent to*

$$\underset{\mathcal{V}_1,\ldots,\mathcal{V}_k}{\text{minimize}} \;\; \text{Trace}\left(\overline{Y}^T L \overline{Y}\right), \tag{5.6}$$

where $\overline{Y} \in \mathbb{R}^{|\mathcal{V}|\times k}$ *is such that* $\overline{Y}_{i\ell} = \sqrt{\frac{\deg(i)}{\text{Vol}(\mathcal{V}_\ell)}} \mathbb{1}\{i \in \mathcal{V}_\ell\}$, *and satisfies* $\overline{Y}^T \overline{Y} = I$.

Since the optimization in (5.6) is NP-hard, one considers a spectral relaxation of the problem by minimizing over all $X \in \mathbb{R}^{|\mathcal{V}|\times k}$ with orthonormal columns as discussed in Section 2.2, and the solution to this relaxed problem is the matrix of k leading orthonormal eigenvectors of L.

The above discussion motivates a spectral k-way partitioning approach based on minimizing NH-cut. The method is listed in Algorithm NH-Cut. The form of Laplacian matrix in (5.5) also suggests that the problem of minimizing NH-cut may be alternatively expressed as the problem of partitioning a graph with weighted adjacency matrix

$$A = H \Delta^{-1} H^T \,. \tag{5.7}$$

Such a graph is related to the star expansion of the hypergraph (Agarwal et al., 2006). The intuition behind the k-means step of Algorithm NH-Cut follows similar to that of Spectral Clustering, and we assume that the approximate k-means method of (Ostrovsky et al., 2012) is used, that provides a near optimal solution in a single iteration.

Algorithm NH-Cut : Normalized hypergraph cut minimization

Input: Incidence matrix H of a hypergraph $(\mathcal{V}, \mathcal{E})$.

1: Let $D \in \mathbb{R}^{n\times n}, \Delta^{|\mathcal{E}|\times|\mathcal{E}|}$ be diagonal with $D_{ii} = \sum_{\ell=1}^{|\mathcal{E}|} H_{i\ell}$, and $\Delta_{\ell\ell} = \sum_{i=1}^{n} H_{i\ell}$.

2: Define the matrix $L \in \mathbb{R}^{n\times n}$ as $L = I - D^{-1/2}H\Delta^{-1}H^T D^{-1/2}$.

3: Compute k leading orthonormal eigenvectors of L, denoted by $X \in \mathbb{R}^{n\times k}$.

4: Normalize rows of X to have unit norm, and denote this matrix as $\overline{X}$.

5: Run k-means on the rows of $\overline{X}$.

Output: Partition of $\mathcal{V}$ that corresponds to the clusters obtained from k-means.

5.3 Consistency of spectral hypergraph partitioning

We now analyze the spectral algorithms TTM-ext and NH-Cut. One can easily observe the similarity between both methods, particularly in adjacency matrices corresponding to the clique and star expansions, (5.1) and (5.7), respectively. Hence, it suffices to study either one of them. We consider the NH-Cut algorithm, and will show that its asymptotic properties are quite similar to TTM. Hence, a straightforward combination of the analysis of both NH-Cut and TTM easily leads to conclusions about TTM-ext.

5.3.1 The random hypergraph Laplacian

Unlike previous chapters, where we dealt with random adjacency tensors of dimension n, the size of the random incidence matrix H in this case is a not deterministic as it depends on the number of generated edges. This poses difficulties in working with the form of hypergraph Laplacian in (5.5), and so we rely on an alternative representation. The Laplacian can be written as

$$L = I - \sum_{e\in\mathcal{E}} \frac{1}{|e|} D^{-1/2} a_e a_e^T D^{-1/2} \ , \tag{5.8}$$

where for $e \subset \mathcal{E}$, $a_e \in \{0,1\}^n$ with $(a_e)_i = 1$, if vertex $i \in e$, and 0 otherwise.

Let $\beta_M = \sum_{m=2}^{M} \binom{n}{m}$. Note that β_M is the maximum number of edges the hypergraph can contain given the fact that its range is M. For convenience, we define a bijective map $\xi : \{1, 2, \ldots, \beta_M\} \to \{e \subset \mathcal{V} : 2 \le |e| \le M\}$, where each ξ_j refers to a subset of vertices, *i.e*, a

possible edge in the given hypergraph. Then the Laplacian can be expressed as

$$L = I - \sum_{j=1}^{\beta_M} \frac{\mathbb{1}\{\xi_j \in \mathcal{E}\}}{|\xi_j|} D^{-1/2} a_{\xi_j} a_{\xi_j}^T D^{-1/2}, \tag{5.9}$$

where the summation is over all possible edges of size at most M, but the missing edges do not contribute to the sum. Similarly, one can express the degree matrix D as

$$D_{ii} = \deg(i) = \sum_{e \in \mathcal{E}} (a_e)_i = \sum_{j=1}^{\beta_M} \mathbb{1}\{\xi_j \in \mathcal{E}\} (a_{\xi_j})_i. \tag{5.10}$$

The above representation corresponds to an 'extended' version of the incidence matrix as $\overline{H} \in \{0,1\}^{n \times \beta_M}$, whose j^{th} column is $\mathbb{1}\{\xi_j \in \mathcal{E}\} a_{\xi_j}$, *i.e*, $\overline{H}$ contains the columns of H with additional zero columns inserted to account for missing edges. This holds for any hypergraph of range M defined on the set $\mathcal{V}$. We use this representation to keep the number of columns as a deterministic quantity. We now discuss how the described planted partition model for hypergraphs, with maximum edge size M, can be expressed in terms of the extended incidence matrix $\overline{H} \in \{0,1\}^{n \times \beta_M}$. Let h_j, $j = 1, 2, \ldots, \beta_M$ be independent Bernoulli random variables that indicate the presence of the edge $\xi_j \subset \mathcal{V}$. By description of the model, if $\xi_j = \{i_1, i_2, \ldots, i_{m_j}\}$ for some $m_j \in \{2, \ldots, M\}$, then the random variable $h_j \sim \text{Bernoulli}\Big(\alpha_{m_j} \mathbf{B}^{(m_j)}_{\psi_{i_1} \psi_{i_2} \ldots \psi_{i_{m_j}}}\Big)$. The j^{th} column of $\overline{H}$ is $h_j a_{\xi_j}$, and hence, the Laplacian matrix for the random hypergraph is

$$L = I - \sum_{j=1}^{\beta_M} \frac{h_j}{|\xi_j|} D^{-1/2} a_{\xi_j} a_{\xi_j}^T D^{-1/2}, \qquad \text{where } D_{ii} = \sum_{j=1}^{\beta_M} h_j (a_{\xi_j})_i. \tag{5.11}$$

At this stage, we note that the above matrices depend on the number of vertices n. For ease of notation, we do not explicitly mention this dependence.

5.3.2 Consistency of NH-Cut algorithm

This section presents a bound on the error incurred by NH-Cut algorithm. As used in the previous chapters, we let $\psi' : \{1, \ldots, n\} \to \{1, \ldots, k\}$ denote the labels obtained from the algorithm, and the clustering error is defined as in (2.20).

We show that if **(i)** the partition is identifiable, and **(ii)** the hypergraph is not too sparse, then indeed $\text{Error}_{\text{NH-Cut}}(\psi, \psi')$ is bounded by a quantity that is at most sub-linear in n. Furthermore, the bound holds with probability $(1 - o(1))$. This immediately implies that NH-Cut

algorithm is weakly consistent. However, we show later that for particular model parameters, NH-Cut algorithm can even recover the partition exactly, *i.e*, $\text{Error}_{\text{NH-Cut}}(\psi, \psi') = o(1)$.

One typically analyzes the population version of a spectral algorithm, and then uses the fact that the spectral properties of the Laplacian eventually concentrates around those of the population Laplacian. From this point of view, we consider the population version of the hypergraph Laplacian (5.11) defined as

$$\mathcal{L} = I - \sum_{j=1}^{\beta_M} \frac{\mathsf{E}[h_j]}{|\xi_j|} \mathcal{D}^{-1/2} a_{\xi_j} a_{\xi_j}^T \mathcal{D}^{-1/2} \ , \tag{5.12}$$

where $\mathcal{D}$ is the expected degree matrix, *i.e*, $\mathcal{D}_{ii} = \sum_{j=1}^{\beta_M} \mathsf{E}[h_j](a_{\xi_j})_i$. We also define the quantity $d = \min_{i \in \{1,\ldots,n\}} \mathcal{D}_{ii}$. Without loss of generality, we may also assume that for a given n, the community sizes are $n_1 \geq n_2 \geq \ldots \geq n_k$

Before stating the main result, it is useful to elaborate on the aforementioned conditions under which the derived error bound holds. A lower bound on the sparsity of the hypergraph is a standard requirement to ensure that the concentration of the spectral properties eventually hold, and is often used in the graph literature (Lei and Rinaldo, 2015; Le et al., 2015), and was also required in the consistency results of previous chapters. In our setting, this can be stated in terms of the sparsity factors α_m, or more simply, in terms of the minimum expected degree d, that grows with n but at a rate controlled by the sparsity factors.

A more critical condition is the identifiability of the partition. This condition was implicit in the discussions of previous chapters, but we discuss it in more detail here. Note that the definition of the hypergraph Laplacian essentially implies that the hypergraph is reduced to a graph with self loops. Hence, the performance of NH-Cut algorithm crucially depends on the identifiability of the partition from $\mathcal{L}$, or rather from the reduced graph with the population adjacency matrix

$$\mathcal{A} = \mathsf{E}[A] = \sum_{j=1}^{\beta_M} \frac{\mathsf{E}[h_j]}{|\xi_j|} a_{\xi_j} a_{\xi_j}^T \ .$$

The following result provides a characterization of $\mathcal{L}$ and $\mathcal{A}$, which in turn helps to quantify the condition for identifiability of the vertex classes from $\mathcal{L}$.

Lemma 5.5. *Let $Z \in \{0,1\}^{n \times k}$ denote the assignment matrix corresponding to the partition*

ψ. Then the population hypergraph Laplacian is given by

$$\mathcal{L} = I - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2} \,, \tag{5.13}$$

where $\mathcal{A}$ can be expressed as

$$\mathcal{A} = ZGZ^T - J \,. \tag{5.14}$$

Here, $J \in \mathbb{R}^{n\times n}$ is diagonal with $J_{ii} = J_{jj}$ whenever $\psi_i = \psi_j$, and $G \in \mathbb{R}^{k\times k}$.

Furthermore, $\mathcal{L}$ contains k eigenvalues for which the corresponding orthonormal eigenvectors are the columns of the matrix $\mathcal{X} = Z(Z^TZ)^{-1/2}U$, where $U \in \mathbb{R}^{k\times k}$ is orthonormal.

The representation in (5.14) shows that $\mathcal{A}$ is essentially of rank k, except for the diagonal entries. Owing to the first term in (5.14), one does expect $\mathcal{L}$ to have k eigenvectors whose entries are constant in each community. As discussed later, a close inspection of $\mathcal{X}$ reveals that indeed the columns of $\mathcal{X}$ satisfy this property. Thus, if the spectral stage of NH-Cut algorithm can extract $\mathcal{X}$, then zero error can be achieved from the k-means step.

In general, $\mathcal{X}$ need not correspond to leading eigenvectors $\mathcal{L}$ (as computed in NH-Cut algorithm). This is true even for certain types of graphs, for instance k-colorable graphs (Alon and Kahale, 1997). This effect is more pronounced in non-uniform hypergraphs due to the presence of a large number of model parameters. To account for this factor, we define the following quantity

$$\delta = \left(\lambda_{\min}(G) \min_{1\le i\le n} \frac{n_{\psi_i}}{\mathcal{D}_{ii}}\right) - \max_{1\le i,j\le n} \left|\frac{J_{ii}}{\mathcal{D}_{ii}} - \frac{J_{jj}}{\mathcal{D}_{jj}}\right| \,, \tag{5.15}$$

where n_{ψ_i} is the size of the community in which vertex i belongs. We show that if $\delta > 0$, then the columns of $\mathcal{X}$ are the k leading eigenvectors of $\mathcal{L}$. Here, $\lambda_{\min}(G)$ refers to the smallest eigenvalue of G. Thus, we can state the consistency result for NH-Cut algorithm as below.

Theorem 5.6. *Consider a random hypergraph on n vertices generated according to the planted partition model described in Section 5.1. Assume that n is sufficiently large, and the size of the k classes are $n_1 \ge n_2 \ge \ldots \ge n_k$. Let d be the minimum expected degree, and δ be the quantity defined in (5.15).*

There exists an absolute constant $C > 0$, such that, if $\delta > 0$ and

$$d > C\frac{kn_1(\ln n)^2}{\delta^2 n_k} \tag{5.16}$$

then with probability at least $1 - O\left((\ln n)^{-1/4}\right)$,

$$\text{Error}_{\text{NH-Cut}}(\psi, \psi') = O\left(\frac{kn_1 \ln n}{\delta^2 d}\right). \tag{5.17}$$

Note here that the quantities δ, d and k can vary with n. On substituting the condition on d into (5.17), one can see that $\text{Error}_{\text{NH-Cut}}(\psi, \psi') = o(n)$ with probability $(1 - o(1))$. Hence, NH-Cut algorithm is weakly consistent if the conditions of the theorem are satisfied. However, we show later that in certain dense hypergraphs, the bound in (5.17) may eventually decay to zero. Thus, NH-Cut algorithm is guaranteed to exactly recover the communities in such cases.

In Section 5.4, we consider particular instances of the planted model, and illustrate the dependance of the above result on the model parameters. For instance, (5.16) implies that the result holds if the sparsity factor (α_m) is above a certain threshold (see Corollaries 5.12 and 5.13). Even when (5.16) holds, higher error is incurred for a sparse hypergraph (small d) or when the number of communities k is large.

One may note that $\delta > 0$ is the condition for identifiability of the partition, and is essential for success of the algorithm. Typically, one does find that $\delta \downarrow 0$ as $n \to \infty$. To this end, the condition (5.16) implies that δ cannot decay rapidly as $\delta^2 d$ needs to maintain a minimum growth rate. We also note that δ quantifies identifiability of the partition and $\text{Error}_{\text{NH-Cut}}(\psi, \psi')$ varies as $\frac{1}{\delta^2}$. Hence, if the model parameters are such that δ is small, for instance if the probability of inter-community edges is very close to that of within community edges, then $\text{Error}_{\text{NH-Cut}}(\psi, \psi')$ is larger.

Before presenting the proof of Theorem 5.6, we comment on the assumption of sufficiently large n. Note that the sole purpose of this assumption is to ensure the success of the k-means algorithm. If n is large enough, the condition (5.16) ensures that the approximate k-means method of Ostrovsky et al. (2012) provides a near optimal solution, which is worse by only a constant factor. Earlier works on spectral graph partitioning (Rohe et al., 2011; Lei and Rinaldo, 2015) assumed the existence of such a near optimal solution with probability 1. To demonstrate the effect of such an assumption, we state the following result, which is a modification of Theorem 5.6 under the above assumption.

Corollary 5.7. *Consider a random hypergraph on n vertices generated according to the planted partition model, and let the other quantities be as defined in Theorem 5.6. Assume that for a constant $\gamma > 1$, there is a γ-approximate[1] k-means algorithm that succeeds with probability 1.*

[1] Informally, a γ-approximate k-means methods returns a solution for which the objective of the k-means problem is at most γ times the global minimum, where $\gamma > 1$. See (2.25) for a formal definition.

There exists an absolute constant $C > 0$, such that, if $\delta > 0$ and

$$d > C\frac{\ln n}{\delta^2} \tag{5.18}$$

then with probability at least $1 - \frac{4}{n^2}$,

$$\text{Error}_{\text{NH-Cut}}(\psi, \psi') = O\left(\frac{kn_1 \ln n}{\delta^2 d}\right). \tag{5.19}$$

The result reveals that if a good k-means algorithm is available, then the success probability of NH-Cut algorithm increases, and the result is also applicable for more sparse hypergraph since the condition (5.18) is weaker than (5.16). However, based on the existing results in the k-means literature, one should consider the following remark.

Remark 5.8. If the data satisfies certain clusterability criterion[1], then the efficient variants of k-means (Kumar et al., 2004; Ostrovsky et al., 2012) provide a γ-approximate solution with a constant probability $\rho < 1$. Both γ and ρ depend on various factors including k, clusterability criterion etc.

In view of the above remark, Corollary 5.7 is too optimistic. Recently, Gao et al. (2015) pointed that if one uses the method of Kumar et al. (2004), then γ grows with k. In addition, one should also note that the success probability of this method is $\rho = c^k$ for an absolute constant $c \in (0, 1)$. Hence, a spectral partitioning algorithm using this method cannot succeed with probability $(1 - o(1))$. Instead, we use the method of Ostrovsky et al. (2012) to achieve a higher success rate as stated in Theorem 5.6. The only additional assumption is that of sufficiently large n. We note that this requirement, along with condition (5.16), can be relaxed if one only aims for a constant success probability. This is shown in the following modification of Theorem 5.6, where we assume that the k-means algorithm of Ostrovsky et al. (2012) is used.

Corollary 5.9. *Consider a random hypergraph on n vertices generated according to the planted partition model, and let the other quantities be as defined in Theorem 5.6.*

There exist absolute constants $C > 0$ and $\epsilon \in (0, 0.015)$, such that, if $\delta > 0$ and

$$d > \frac{C}{\epsilon^2}\frac{kn_1 \ln n}{\delta^2 n_k} \tag{5.20}$$

[1]Various clusterability criteria have been studied in the literature. In this work, we consider the notion of ϵ-separability proposed by Ostrovsky et al. (2012). See Section 2.5.3 for details.

then with probability at least $1 - O(\sqrt{\epsilon})$,

$$\text{Error}_{\text{NH-Cut}}(\psi, \psi') = O\left(\frac{kn_1 \ln n}{\delta^2 d}\right). \tag{5.21}$$

5.3.3 Proof of Theorem 5.6

We now present an outline of the proof of Theorem 5.6 using a series of lemmas. The proofs for these lemmas are given in Appendix 5.A.2. The result is obtained by proving the following facts:

1. If NH-Cut algorithm is performed on the population Laplacian $\mathcal{L}$, then under the condition of $\delta > 0$, the obtained partition is correct.

2. The deviation of L from $\mathcal{L}$ is bounded above, and the bound holds with probability at least $(1 - \frac{4}{n^2})$.

3. As a consequence of above facts, the standard matrix perturbation bounds (Stewart and Sun, 1990) imply that the eigenvalues and the corresponding eigenspaces of L concentrate about those of $\mathcal{L}$.

4. If (5.16) holds, then k-means stage of NH-Cut algorithm succeeds in obtaining a near optimal solution with probability at least $1 - O\left((\ln n)^{-1/4}\right)$.

5. The partitioning error can be expressed in terms of the above bounds, which leads to (5.17).

Corollaries 5.7 and 5.9 can be proved in similar manner. This is discussed in Appendix 5.A.2. We now prove the first two facts, and the subsequent ones follow similar to the arguments in Chapter 3. The following result extends Lemma 5.5.

Lemma 5.10. *If* $\delta > 0$, *then the* k *leading orthonormal eigenvectors of* $\mathcal{L}$ *correspond to the columns of the matrix* $\mathcal{X} = Z(Z^T Z)^{-1/2} U$.

In the above result, $Z^T Z$ is a diagonal matrix with entries being the sizes of the k vertex classes. Hence, both $Z^T Z$ and U are of the rank k. Due to this, one can observe that the matrix $\mathcal{X}$ contains exactly k distinct rows, each corresponding to a particular class, *i.e*, if $A_{i\cdot}$ denotes i^{th} row of a matrix A, then for any two vertices $i, j \in \mathcal{V}$,

$$\mathcal{X}_{i\cdot} = \mathcal{X}_{j\cdot} \iff Z_{i\cdot} = Z_{j\cdot} \iff \psi_i = \psi_j \ .$$

Moreover, since U is orthonormal, the distinct rows of $\mathcal{X}$ are orthogonal. Hence, after row normalization, the distinct rows correspond to k orthonormal vectors in $\mathbb{R}^k$, which can be easily clustered by k-means algorithm to obtain the true communities. Technically, δ is a lower bound on the eigen-gap between the k^{th} and $(k+1)^{th}$ smallest eigenvalues of $\mathcal{L}$. Since, it is difficult to obtain a simple characterization of the eigen gap, we resort to the use of δ as defined in (5.15).

Next, we bound the deviation of a random instance of L from the population Laplacian $\mathcal{L}$. This bound relies on the use of matrix Bernstein inequality (Tropp, 2012), also stated in Theorem 2.11. We note that for graphs, sharp deviation bounds have been used (Lei and Rinaldo, 2015), but such techniques cannot be directly extended to the case of hypergraphs.

Lemma 5.11. *If $d > 9 \ln n$, then with probability at least $(1 - \frac{4}{n^2})$,*

$$\|L - \mathcal{L}\|_2 \leq 12\sqrt{\frac{\ln n}{d}} \,. \tag{5.22}$$

One can now follow the lines of Lemmas 3.10–3.12 to arrive at the conclusion of Theorem 5.6.

5.4 Consistency in special cases

We now study the implications of Theorem 5.6 for partitioning particular models of uniform and non-uniform hypergraphs. We also discuss the conditions for identifiability in special cases.

5.4.1 Balanced partition in uniform hypergraph

Let the n vertices be divided into k groups such that each group contains $\frac{n}{k}$ vertices. We now consider a random m-uniform hypergraph on the vertices generated as follows. Let $p, q \in [0, 1]$ be constants with $(p + q) \leq 1$, and $\alpha_m \in (0, 1]$ be the sparsity factor dependent on n. For any m vertices from the same group, there is an edge among them with probability $\alpha_m(p + q)$. If all the m vertices do not belong to same group, then there is an edge with probability $\alpha_m q$.

In terms of the model in Section 5.1, one can see that $M = m$, and for all $r < m$, $\alpha_r = 0$. The m^{th} order k-dimensional tensor $\mathbf{B}^{(m)}$ is given by

$$\mathbf{B}^{(m)}_{j_1 j_2 \ldots j_m} = \begin{cases} p + q & \text{if } j_1 = j_2 = \ldots = j_m, \\ q & \text{otherwise.} \end{cases}$$

One can see that for $m = 2$, this model corresponds to the sparse stochastic block model

considered in (Lei and Rinaldo, 2015) with balanced community sizes, and if $\alpha_2 = 1$, one has the standard four parameter stochastic block model (Rohe et al., 2011). The following corollary to Theorem 5.6 shows the consistency of the NH-Cut algorithm.

Corollary 5.12. *In the above model,*

$$\delta = \frac{p\alpha_m n}{mkd}\binom{\frac{n}{k}-2}{m-2}, \tag{5.23}$$

and hence, the partition is identifiable for all $p > 0$. Moreover, if

$$\alpha_m \geq C\frac{k^{2m-1}n(\ln n)^2}{\binom{n}{m}} \tag{5.24}$$

for some absolute constant $C > 0$, then the conditions in Theorem 5.6 are satisfied, and hence, we have

$$\text{Error}_{\text{NH-Cut}}(\psi, \psi') = O\left(\frac{k^{2m-2}n^2 \ln n}{p^2\alpha_m\binom{n}{m}}\right) = o(n) \tag{5.25}$$

with probability $(1 - o(1))$.

The lower bound on α_m mentioned in Corollary 5.12 needs some discussion. One can verify that in the above model, the expected number of edges lie in the range $[q\alpha_m\binom{n}{m}, (p+q)\alpha_m\binom{n}{m}]$, *i.e*, it is about $\alpha_m\binom{n}{m}$ up to a constant scaling. The lower bound on α_m specifies that the number of edges must be at least $\Omega\left(k^{2m-1}n(\ln n)^2\right)$. This also indicates that for a larger m, more edges are required to ensure the error bound of Corollary 5.12. Since, $\alpha_m \leq 1$, one can see that the result is applicable for $k = O(n^{0.5-\epsilon})$ for all $\epsilon > \frac{1}{2(2m-1)}$. Even consistency results for graph partitioning require similar condition (Rohe et al., 2011; Choi et al., 2012).

A closer look at the condition (5.24) shows if k is constant or increases slowly, $k = O(\ln n)$, then a sufficient condition for weak consistency of Algorithm NH-Cut is $\alpha_m \geq C\frac{(\ln n)^{2m+1}}{n^{m-1}}$, where the constant C depends only on m. In case of graph partitioning, this level of sparsity is needed when one relies on matrix Bernstein inequality. However, recent results (Lei and Rinaldo, 2015) reduced the lower bound by using sharp concentration bounds for the binary adjacency matrix. Corollary 5.12 also indicates that if k increases at a higher rate, for example $k = n^a$, then consistency can be guaranteed only when hypergraph is more dense.

On the other extreme are dense uniform hypergraphs studied in Corollary 4.4, where $\alpha_m = 1$. In this case, if $k = O(\ln n)$ then $\text{Error}_{\text{NH-Cut}}(\psi, \psi') = O\left(\frac{(\ln n)^{2m-1}}{n^{m-2}}\right)$. Thus, the error decreases at a faster rate for m-uniform hypergraphs with larger m. In fact, for $r \geq 3$, above bound

indicates that $\text{Error}_{\text{NH-Cut}}(\psi, \psi') = o(1)$, *i.e*, Algorithm NH-Cut guarantees exact recovery of the partition for large n. Moreover, one can observe that the above error rate is similar to TTM, indicating that NH-Cut and TTM (and also TTM-ext) have similar performance, and are provably better than HOSVD. This fact will be later validated emprically in Chapter 8.

We also note that though we have only discussed about the growth rate of k, and the minimum density α_n, the condition in (5.24) can also be restated in terms of the minimum cluster sizes. Let $n_1 = \frac{n}{k}$ be the size of each cluster, then (5.24) is equivalent to stating $n_1 \geq C\sqrt{n}\left(\frac{\sqrt{n}(\ln n)^2}{\alpha_m}\right)^{1/(2m-1)}$. So, while in the dense regime ($\alpha_n = 1$), the minimum growth rate of clusters is $\Omega(n^{0.5+\epsilon})$ for $\epsilon > \frac{1}{2(2m-1)}$, a larger growth rate of clusters is required in the sparse regime.

Lastly, we discuss the effect of δ and the parameters p, q in this setting. Note that the case $q = 0$ is not interesting as there are no edges among different groups, and hence, the partition can be identified by a simple breadth-first search. On the other hand, $p = 0$ generates a random uniform hypergraph with all identical edges. Hence, the partition cannot be identified in this case. This can also be seen from (5.23), where $\delta = 0$. In general, p denotes the gap between the probability of edge occurrence among vertices from same community and the probability with which vertices from different communities form an edge. Since δ is linear in p, one can observe from Theorem 5.6 that $\text{Error}_{\text{NH-Cut}}(\psi, \psi')$ varies as $\frac{1}{p^2}$ with p. However, note that the model assumes that p does not vary with n, and may be treated as a constant in the asymptotic case.

5.4.2 Balanced partition in non-uniform hypergraph

We now consider the case of non-uniform hypergraph of range M, where M may vary with n. As in Section 5.4.1, assume that n vertices are equally split into k groups. Also let $p, q \in (0, 1)$ such that $(p+q) \leq 1$, and for $m = 2, \ldots, M$, let $\mathbf{B}^{(m)}$ be the m^{th}-order symmetric k-dimensional tensor with

$$\mathbf{B}^{(m)}_{j_1 j_2 \ldots j_m} = \begin{cases} p + q & \text{if } j_1 = j_2 = \ldots = j_m, \\ q & \text{otherwise.} \end{cases}$$

Setting $\alpha_m \in (0, 1]$ as the sparsity factors, we obtain a model, where the edges appear independently, and for each m, an edge on m vertices from the same group appears with probability $\alpha_m(p + q)$. For any set of m vertices from different groups, there is an edge among them with probability $\alpha_m q$.

Since, the non-unifom hypergraph is a superposition of the m-uniform hypergraphs for $m = 2, \ldots, M$, one can easily derive a consistency result in the non-uniform case by appling

Corollary 5.12 for each of the uniform components. However, observe that the number of edges of size m is $\Theta\left(\alpha_m \binom{n}{m}\right)$, and hence, the requirement $\alpha_m \binom{n}{m} \geq C_m k^{2m-1} n (\ln n)^2$ for each m implies that the number of m-size edges should increase with m. This contradicts the natural intuition in existing random models (Darling and Norris, 2005), where the hypergraph contains less edges of higher cardinality. The same phenomenon is also observed in practice (see Chapter 8). The following consistency result takes this fact into account.

Corollary 5.13. *The partition in the above model is identifiable for all $p > 0$. In addition, let $(\theta_m)_{m=2}^{\infty}$ be a non-negative sequence not dependent of n, and assume that for any $n \in \mathbb{N}$ and $m = 2, \ldots, M$, the sparsity factor*

$$\alpha_m = \frac{\theta_m n^a (\ln n)^b}{\binom{n}{m}}$$

for some $a \geq 1$ and $b \geq 2$. There exists an absolute constant C, such that, if

$$\sum_{m=r}^{M} m\theta_m \leq C \left(\frac{n^{a-1} (\ln n)^{b-2}}{k^{2r-1}} \right) \tag{5.26}$$

for $r = \min\{m : \theta_m > 0\}$, then $\mathrm{Error}_{\text{NH-Cut}}(\psi, \psi') = o(n)$ with probability $(1 - o(1))$.

In the above result, r denotes the smallest size of an edge in the hypergraph. In practice. $(\theta_m)_{m=2}^{\infty}$ is a decreasing sequence, and hence, the number of m-size edges also decreases with m. In particular, if $\theta_2 > 0$, $\sum_{m=2}^{\infty} m\theta_m < \infty$, and $k = O\left(n^{(a-1)/3} (\ln n)^{(b-2)/3}\right)$, then Algorithm NH-Cut is weakly consistent. Thus, if the hypergraph is sparse, *i.e*, $a = 1$, consistency is guaranteed only for logarithmic growth in k, whereas more vertex classes can be consistently detected only in dense hypergraphs. Observe that the problem gets harder if $r > 2$.

5.4.3 Identifiability of the partition

In the previous two sections, we considered problems where the partition is identifiable from $\mathcal{L}$. This need not hold for arbitrary model parameters. We now briefly discuss few cases, which show that the partition is typically identifiable under reasonable choice of model parameters.

Example 1. Consider a 3-uniform hypergraph on n vertices. For simplicity assume there are $k \geq 3$ clusters of equal size. We define $\mathbf{B}^{(3)}$ as follows

$$\mathbf{B}^{(3)}_{j_1 j_2 j_3} = \begin{cases} p_1 & \text{if } j_1 = j_2 = j_3, \\ p_2 & \text{if exactly two of them are identical,} \\ p_3 & \text{if } j_1, j_2, j_3 \text{ are all different.} \end{cases}$$

for some constants $p_1, p_2, p_3 \in [0,1]$. Observe that the above situation is the most general case provided that the clusters are statistically identical. In this setting, it is easy to see that the following statement holds.

Lemma 5.14. *Assume that n is a multiple of k. Then $\delta > 0$ if and only if*

$$(p_2 - p_3) + \frac{1}{k}(p_1 - 3p_2 + 2p_3) - \frac{2}{n}(p_1 - p_2) > 0. \tag{5.27}$$

In particular, $\delta > 0$ when $p_1 > p_2 > p_3$, or at most one inequality is replaced by equality.

Note that the setting of Section 5.4.1 follows when $p_1 > p_2 = p_3$, while the case $p_1 = p_2 = p_3$ corresponds to a random hypergraph with all edges following the same law. Obviously, the partition is not identifiable in the latter case. More generally, the order of probabilities $p_1 > p_2 > p_3$ is intuitive as it implies that an edge has a larger probability of occurrence if it has more vertices from the same community. One may compare this observation with the case of graphs, where partitioning based on the leading eigenvectors of Laplacian works only when edges within each community occur more frequently than edges across communities. The opposite scenario, found in colorable graphs, requires one to consider eigenvectors corresponding to the other end of the spectrum (Alon and Kahale, 1997). Moreover, if $p_2 > p_3$ and k grows with n, one can observe that δ mostly depends on the gap $(p_2 - p_3)$, and hence, the error $\text{Error}_{\text{NH-Cut}}(\psi, \psi')$ is proportional to $\frac{1}{(p_2-p_3)^2}$.

Example 2. We now modify the above model by allowing edges of size 2 to be present. In particular, assume $\alpha_2 = 1$ and $\mathbf{B}^{(2)} = I$, which means all pairwise edges within each community are present, and no two vertices from different communities form a pairwise edge. In addition, let $\alpha_3 \in [0,1]$ be arbitrary. Then, one can observe the following.

Lemma 5.15. *Assume that n is a multiple of k. Then $\delta > 0$ if and only if*

$$\frac{1}{2} + \frac{n\alpha_3}{3}\left((p_2 - p_3) + \frac{(p_1 - 3p_2 + 2p_3)}{k} - \frac{2(p_1 - p_2)}{n}\right) > 0. \tag{5.28}$$

It is easy to see that if $\alpha_3 = 0$, then the hypergraph is a graph with k disconnected components, and hence, the partition is identifiable. However, even when $\alpha_3 = o(\frac{1}{n})$, the pairwise edges eventually dominate and the partition can be identified for arbitrary values of p_1, p_2, p_3. On the other hand, if α_3 grows faster than $\frac{1}{n}$ (for instance $\alpha_3 = 1$), then the situation is eventually similar to that of Lemma 5.14. The critical case is $\alpha_3 = \Theta(\frac{1}{n})$, where the expected number of 2-way and 3-way edges are of similar order. In this case, (5.28) suggests that the partition can be identified ($\delta > 0$) even when $p_2 < p_3$ provided p_3 is sufficiently small.

Example 3. In the above cases, we restricted ourselves to communities of equal size. The arguments also hold for $\frac{n_1}{n_k} = O(1)$. However, if $n_k \ll n_1$ or the probability of edges vary across different communities, then the second term in (5.15) can lead to $\delta \leq 0$, or equivalently, may affect the identifiability of the partition. To study this effect, we consider the following model for m-uniform hypergraphs.

Let $\alpha_m = 1$, and there are $k = 2$ classes of size s and $(n - s)$. We assume $s = o(n)$, and define $\mathbf{B}^{(m)} \in \mathbb{R}^{2\times 2\times \ldots \times 2}$ as

$$\mathbf{B}^{(m)}_{j_1 j_2 \ldots j_r} = \begin{cases} 1 & \text{if } j_1 = j_2 = \ldots = j_r = 1, \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

For $m = 2$, the model is same as that of a s clique planted in a Erdös-Rényi graph. This model presents a high disparity in both community sizes and degree distributions. We make the following comment on the identifiability of the partition under this model.

Lemma 5.16. *For a given $m \geq 2$, there exists a finite constant s_m such that $\delta > 0$ for the above model for all $s \geq s_m$.*

Thus, when s grows with n, the partition can be eventually identified from $\mathcal{L}$. The proof of the above result shows that both the terms in (5.15) decay with n, but the ratio of the first term to the second grows as $\Omega(s)$. We believe that a similar observation can be made in more general situations, where this growth rate depends on the size of the smallest community.

In view of the above lemma, it is interesting to know whether Algorithm NH-Cut is able to detect small cliques in uniform hypergraphs. This is indeed true, but due to the generality of the approach, as presented in this thesis, the minimal growth rate for s needed to accurately find the clique from L is not optimal. More precisely, it is worse by a logarithmic factor in the case of graphs. However, Lemma 5.16 shows that one can use spectral techniques similar to (Alon et al., 1998) for finding planted cliques in hypergraphs.

5.A Proofs for results in this chapter

5.A.1 Proofs for results in Section 5.2

Proof of Proposition 5.2

In the first claim, note that for $r \geq 3$, $\beta_r = 0$ and hence, $\overline{Y}^{(r)}$ is a constant matrix of ones. Thus,

$$\text{Trace}\left(\mathbf{A}^{(m)} \times_1 \overline{Y}^{(1)^T} \times_2 \overline{Y}^{(2)^T} \times_3 \ldots \times_m \overline{Y}^{(m)^T}\right) = \sum_{\ell=1}^{k} \sum_{i_1,\ldots,i_m} \mathbf{A}^{(m)}_{i_1\ldots i_m} Y_{i_1\ell} Y_{i_2\ell} \ ,$$

where $Y_{i\ell} = \dfrac{\mathbb{1}\{i \in \mathcal{V}_\ell\}}{\sqrt{\text{Vol}(\mathcal{V}_\ell)}}$. Hence, one can restate the objective function in (5.2) as

$$\underset{Y}{\text{maximize}} \sum_{\ell=1}^{k} \sum_{i_1,i_2=1}^{n} Y_{i_1\ell} Y_{i_2\ell} \sum_{m=2}^{M} \sum_{i_3,\ldots,i_m=1}^{n} \frac{\mathbf{A}^{(m)}_{i_1\ldots i_m}}{(m-1)!} \equiv \underset{Y}{\text{maximize}} \ \text{Trace}(Y^T \overline{A} Y) \ ,$$

where $\overline{A} \in \mathbb{R}^{n\times n}$ with

$$\overline{A}_{ij} = \sum_{m=2}^{M} \sum_{i_3,\ldots,i_m=1}^{n} \frac{\mathbf{A}^{(m)}_{iji_3\ldots i_m}}{(m-1)!} \ . \tag{5.29}$$

Owing to the available discussion on spectral relaxation in (4.6), it suffices to show that the matrix A used in the algorithm is same as $\overline{A}$ (5.29). The diagonal entries $\overline{A}_{ii} = 0$ since the tensors $\mathbf{A}^{(m)}$ are zero at entries with repeated indices. Hence, A and $\overline{A}$ are same at the diagonal. For off-diagonal entries, observe that every term $\mathbf{A}^{(m)}_{iji_3\ldots i_m}$ corresponds to an edge $e \in \mathcal{E}$ with $|e| = m$ and $i, j \in e$. Also, the inner summation in (5.29) contains $(m-2)!$ copies of the same edge. Thus,

$$\overline{A}_{ij} = \sum_{m=2}^{M} \sum_{e\in\mathcal{E}_m : e \ni i,j} \frac{w_e}{m-1} = \sum_{e\in\mathcal{E} : e\ni i,j} \frac{w_e}{|e|-1} = \left(H(\Delta - I)^{-1} H^T\right)_{ij} = A_{ij} \ ,$$

which leads to the first claim.

For the second claim, we briefly describe the clique expansion of a hypergraph. Here, each edge $e \in \mathcal{E}$ is replaced by a clique among all the vertices in e, for every $i, j \in e$, an edge is added between them. Subsequently, the weighted graph formed is a super-position of all cliques.

Hence, the weighted adjacency matrix A is of the form

$$A_{ij} = |\{e \in \mathcal{E} : e \ni i, j\}| = (HH^T)_{ij} \, .$$

for $i \neq j$. This construction also implies $A_{ii} = 0$. Thus, the reduction used in TTM-ext is similar to clique expansion ignoring the factor of $(\Delta - I)^{-1}$, which essentially means that instead of counting all edges containing i, j, we take a weighted sum.

Proof of Proposition 5.4

Denoting the j^{th} column of $\widehat{Y}$ by $\widehat{Y}_{\cdot j}$, one can write

$$\text{Trace}\left(\widehat{Y}^T(D - H\Delta^{-1}H^T)\widehat{Y}\right) = \sum_{\ell=1}^{k} \widehat{Y}_{\cdot\ell}^T(D - H\Delta^{-1}H^T)\widehat{Y}_{\cdot\ell}.$$

Noting that

$$D_{ii} = \sum_{e\in\mathcal{E}} H_{ie} = \sum_{e\in\mathcal{E}}\sum_{j\in e} \frac{1}{|e|} H_{ie} = \sum_{e\in\mathcal{E}}\sum_{j\in\mathcal{V}} \frac{1}{|e|} H_{je}H_{ie} = \sum_{j\in\mathcal{V}} (H\Delta^{-1}H^T)_{ij}, \tag{5.30}$$

one obtains

$$\begin{aligned}
\widehat{Y}_{\cdot\ell}^T(D - H\Delta^{-1}H^T)\widehat{Y}_{\cdot\ell} &= \frac{1}{2}\sum_{i,j\in\mathcal{V}} (H\Delta^{-1}H^T)_{ij}(\widehat{Y}_{i\ell} - \widehat{Y}_{j\ell})^2 \\
&= \frac{1}{2}\sum_{e\in\mathcal{E}} \frac{1}{|e|\text{Vol}(\mathcal{V}_\ell)} \sum_{i,j\in e} (\mathbb{1}\{i \in \mathcal{V}_\ell\} - \mathbb{1}\{j \in \mathcal{V}_\ell\})^2.
\end{aligned}$$

The claim follows by observing that a term in the inner summation contributes only when $i \in \mathcal{V}_\ell, j \notin \mathcal{V}_\ell$ or vice-versa.

5.A.2 Proofs for results in Section 5.3

Proofs for Corollaries 5.7 and 5.9

The proofs are similar to that of Theorem 5.6. To prove Corollary 5.7, we proceed along the lines of the lemmas in Section 5.3. Due to the assumption on k-means, Lemma 3.11 is not required.

On the other hand, Corollary 5.9 follows when we use Lemma 3.11 with constant $\epsilon > 0$. The condition $\epsilon < 0.015$ follows immediately from the requirement of Theorem 2.12.

Proof of Lemma 5.5

We observe that for $i \neq j$,

$$\mathcal{A}_{ij} = \sum_{\ell=1}^{\beta_M} \frac{\mathsf{E}[h_\ell]}{|\xi_\ell|}(a_{\xi_\ell})_i(a_{\xi_\ell})_j = \sum_{m=2}^{M} \sum_{\substack{\ell:|\xi_\ell|=m,\\ i,j\in\xi_\ell}} \frac{\mathsf{E}[h_\ell]}{m}$$
$$= \sum_{m=2}^{M} \sum_{\substack{i_3<i_4<\ldots<i_m,\\ i,j\notin\{i_3,\ldots,i_m\}}} \frac{1}{m}\alpha_m \mathbf{B}^{(m)}_{\psi_i\psi_j\psi_{i_3}\ldots\psi_{i_m}} \ . \tag{5.31}$$

The last equality follows by noting that for every ξ_ℓ such that $|\xi_\ell| = m$ and $\xi_\ell \ni i, j$, we can write ξ_ℓ as $\xi_\ell = \{i, j, i_3, \ldots, i_m\}$, where the vertices $i, j, i_3, ..., i_m$ are distinct. It is interesting to note that the above sum remains same if i, j are replaced by some i', j' such that $\psi_i = \psi_{i'}$ and $\psi_j = \psi_{j'}$. This is true since the terms in (5.31) depend on ψ_i, ψ_j instead of i, j. This observation motivates us to define the matrix $G \in \mathbb{R}^{k\times k}$ such that for any $i, j \in \mathcal{V}$, $i \neq j$,

$$G_{\psi_i\psi_j} = \sum_{m=2}^{M} \sum_{\substack{i_3<i_4<\ldots<i_m,\\ i',j'\notin\{i_3,\ldots,i_m\}}} \frac{1}{m}\alpha_m \mathbf{B}^{(m)}_{\psi_{i'}\psi_{j'}\psi_{i_3}\ldots\psi_{i_m}} \ , \tag{5.32}$$

where i', j' are arbitrary vertices satisfying $\psi_i = \psi_{i'}$ and $\psi_j = \psi_{j'}$. Hence, one can write $\mathcal{A}_{ij} = (ZGZ^T)_{ij}$ for all $i \neq j$, where Z is the assignment matrix. However,

$$\mathcal{A}_{ii} = \sum_{\ell:i\in\xi_\ell} \frac{\mathsf{E}[h_\ell]}{|\xi_\ell|} \neq G_{\psi_i\psi_i}.$$

So, one can write the matrix $\mathcal{A}$ as $\mathcal{A} = ZGZ^T - J$, where $J \in \mathbb{R}^{n\times n}$ is a diagonal matrix defined as $J_{ii} = G_{\psi_i\psi_i} - \mathcal{A}_{ii}$. We also note that for i, i' in the same group, *i.e*, $\psi_i = \psi_{i'}$, we have $\mathcal{D}_{ii} = \mathcal{D}_{i'i'}$ and $J_{ii} = J_{i'i'}$. So we can define matrices $\widetilde{\mathcal{D}}, \widetilde{J} \in \mathbb{R}^{k\times k}$ diagonal such that $\mathcal{D}_{ii} = \widetilde{\mathcal{D}}_{\psi_i\psi_i}$ and $J_{ii} = \widetilde{J}_{\psi_i\psi_i}$ for all $i \in \mathcal{V}$. It is easy to see that $\mathcal{D}Z = Z\widetilde{\mathcal{D}}$ and $JZ = Z\widetilde{J}$.

Using above definitions, we now characterize the eigenpairs of the matrix $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$. To this end, note that (λ, v) is an eigenpair of $\mathcal{L}$ if and only if $((1-\lambda), v)$ is an eigenpair of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$. Hence, it suffices to consider the eigenvalues of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$, and their corresponding eigen spaces.

First, observe that since $G \in \mathbb{R}^{k\times k}$, $\mathcal{A}$ is composed of a matrix of rank at most k that is perturbed by the diagonal matrix J. We show that the orthonormal basis for the range space of ZGZ^T are the eigenvectors that are of interest to us. For this, consider the matrix

$\mathcal{G} = (\widetilde{\mathcal{D}}^{-1}Z^TZ)^{1/2}G(Z^TZ\widetilde{\mathcal{D}}^{-1})^{1/2} - \widetilde{J}\widetilde{\mathcal{D}}^{-1} \in \mathbb{R}^{k\times k}$, and suppose its eigen-decomposition is given by $\mathcal{G} = U\Lambda_1 U^T$, where $U \in \mathbb{R}^{k\times k}$ contains the orthonormal eigenvectors and $\Lambda_1 \in \mathbb{R}^{k\times k}$ is a diagonal matrix of eigenvalues of $\mathcal{G}$. Defining $\mathcal{X} = Z(Z^TZ)^{-1/2}U \in \mathbb{R}^{n\times k}$, we can write that

$$\begin{aligned}
\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\mathcal{X} &= \mathcal{D}^{-1/2}(ZGZ^T - J)\mathcal{D}^{-1/2}Z(Z^TZ)^{-1/2}U \\
&= \mathcal{D}^{-1/2}(ZG(Z^TZ)^{1/2} - Z(Z^TZ)^{-1/2}\widetilde{J})\widetilde{\mathcal{D}}^{-1/2}U \\
&= Z(Z^TZ)^{-1/2}\mathcal{G}U \\
&= Z(Z^TZ)^{-1/2}U\Lambda_1 = \mathcal{X}\Lambda_1,
\end{aligned}$$

which implies that the columns of $\mathcal{X}$ are the eigenvectors of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$ corresponding to the k eigenvalues in Λ_1. Alternatively, the columns of $\mathcal{X}$ are the eigenvectors of $\mathcal{L}$ corresponding to the k eigenvalues in $(I - \Lambda_1)$. Note that the above equalities are derived by repeated use of the facts that diagonal matrices commute and $\mathcal{D}Z = Z\widetilde{\mathcal{D}}$, $JZ = Z\widetilde{J}$. Also, since U is orthonormal, it is easy to verify that the columns of $\mathcal{X}$ are orthonormal.

Proof of Lemma 5.10

We continue from the proof of Lemma 5.5. Note that we need to derive conditions under which $\mathcal{X}$ contain the leading eigenvectors of $\mathcal{L}$. Equivalently, we need to show that the eigenvalues in Λ_1 are strictly larger than other eigenvalues of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$.

Since, $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$ is symmetric and hence, diagonalizable, we can conclude that remaining eigenvectors of the matrix are orthogonal to columns of $\mathcal{X}$. Let the columns of $Y \in \mathbb{R}^{n\times(n-k)}$ be the matrix of the remaining orthonormal eigenvectors of $\mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$, with corresponding eigenvalues given by the diagonal matrix $\Lambda_2 \in \mathbb{R}^{(n-k)\times(n-k)}$. So $Y^TZ(Z^TZ)^{-1/2}U = 0$. Due to the non-singularity of Z^TZ or U, it follows that $Z^TY = 0$, and

$$Y\Lambda_2 = \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}Y = -\mathcal{D}^{-1}JY,$$

that is, the columns of Y are eigenvectors of $(-\mathcal{D}^{-1}J)$. Further, since $\mathcal{D}^{-1}J$ is diagonal, the eigenvalues in Λ_2 are a subset of the entries of $(-\mathcal{D}^{-1}J)$. Thus, to ensure that $\mathcal{X}$ are the leading eigenvectors, one needs to ensure $\min_i(\Lambda_1)_{ii} > \max_i(\Lambda_2)_{ii}$, and hence, one may define $\widetilde{\delta}$ as the eigen-gap,

$$\widetilde{\delta} = \min_{1\le i\le k}(\Lambda_1)_{ii} - \max_{1\le i\le(n-k)}(\Lambda_2)_{ii}. \tag{5.33}$$

Hence, the condition $\widetilde{\delta} > 0$ ensures that columns of $\mathcal{X}$ are leading eigenvectors of $\mathcal{L}$. Though the above definition of $\widetilde{\delta}$ suffices, it cannot be easily verified for a given model. Below, we show

that $\widetilde{\delta} \geq \delta$, where the latter is as defined in (5.15). Note that

$$\max_{1 \leq i \leq (n-k)} (\Lambda_2)_{ii} \leq \max_{1 \leq i \leq n} \left(-\frac{J_{ii}}{\mathcal{D}_{ii}} \right) = \min_{1 \leq i \leq n} \frac{J_{ii}}{\mathcal{D}_{ii}}$$

On the other hand, using Weyl's inequality, we have

$$\min_{1 \leq i \leq k} (\Lambda_1)_{ii} = \lambda_{\min}(\mathcal{G}) \geq \lambda_{\min}((\widetilde{\mathcal{D}}^{-1} Z^T Z)^{1/2} G (Z^T Z \widetilde{\mathcal{D}}^{-1})^{1/2}) - \|\widetilde{J} \widetilde{\mathcal{D}}^{-1}\|_2 \, ,$$

where $\lambda_{\min}(\mathcal{G})$ denotes the minimum eigenvalue of $\mathcal{G}$. The inequality follows by viewing $\mathcal{G}$ as the matrix $(\widetilde{\mathcal{D}}^{-1} Z^T Z)^{1/2} G (Z^T Z \widetilde{\mathcal{D}}^{-1})^{1/2}$ perturbed by $-\widetilde{\mathcal{D}}^{-1} \widetilde{J}$. To simplify further, we note

$$\|\widetilde{J} \widetilde{\mathcal{D}}^{-1}\|_2 = \max_{1 \leq i \leq k} \frac{\widetilde{J}_{ii}}{\widetilde{\mathcal{D}}_{ii}} = \max_{1 \leq i \leq n} \frac{J_{ii}}{\mathcal{D}_{ii}},$$

and using Rayleigh's principle, one can show that

$$\lambda_{\min}((\widetilde{\mathcal{D}}^{-1} Z^T Z)^{1/2} G (Z^T Z \widetilde{\mathcal{D}}^{-1})^{1/2}) \geq \lambda_{\min}(G) \min_{1 \leq i \leq k} \frac{(Z^T Z)_{ii}}{\widetilde{\mathcal{D}}_{ii}}.$$

Combining the above bounds, we conclude that $\widetilde{\delta} \geq \delta$. Here, we use the observation that $(Z^T Z)_{jj}$ equals the size of the j^{th} community. Thus, $\delta > 0$ is a sufficient condition for the claim of the lemma.

Proof of Lemma 5.11

Define $\widehat{L} = I - \mathcal{D}^{-1/2} A \mathcal{D}^{-1/2}$. Note that

$$\|L - \mathcal{L}\|_2 \leq \|L - \widehat{L}\|_2 + \|\mathcal{L} - \widehat{L}\|_2. \tag{5.34}$$

We deal with the two terms separately. First, we show that if $d > 9 \ln n$, then

$$\mathsf{P}\left(\|\mathcal{L} - \widehat{L}\|_2 \geq 3 \sqrt{\frac{\ln n}{d}} \right) \leq \frac{2}{n^2} \, . \tag{5.35}$$

To prove (5.35), we note that

$$\begin{aligned}\mathcal{L} - \widehat{L} &= \mathcal{D}^{-1/2}(A - \mathcal{A})\mathcal{D}^{-1/2} \\ &= \sum_{\ell}(h_\ell - \mathsf{E}[h_\ell])\frac{1}{|\xi_\ell|}\mathcal{D}^{-1/2}a_{\xi_\ell}a_{\xi_\ell}^T\mathcal{D}^{-1/2} \ .\end{aligned}$$

Denoting, each matrix in the sum as Y_ℓ, it is easy to see that $\{Y_\ell\}_\ell$ are independent with $\mathsf{E}[Y_\ell] = 0$. Hence, we can apply matrix Bernstein inequality (Theorem 2.11) to obtain

$$\begin{aligned}\mathsf{P}\left(\|\mathcal{L} - \widehat{L}\|_2 \geq 3\sqrt{\frac{\ln n}{d}}\right) &= \mathsf{P}\left(\left\|\sum_\ell Y_\ell\right\|_2 \geq 3\sqrt{\frac{\ln n}{d}}\right) \\ &\leq 2n\exp\left(\frac{-\frac{9\ln n}{d}}{2\left\|\sum_\ell \mathsf{Var}(Y_\ell)\right\|_2 + \frac{2}{3}\sqrt{\frac{9\ln n}{d}}\max_\ell\|Y_\ell\|_2}\right) \ , \end{aligned} \tag{5.36}$$

where

$$\mathsf{Var}(Y_\ell) = \mathsf{E}[Y_\ell^2] = \mathsf{Var}(h_\ell)\frac{(a_{\xi_\ell}^T\mathcal{D}^{-1}a_{\xi_\ell})}{|\xi_\ell|^2}\mathcal{D}^{-1/2}a_{\xi_\ell}a_{\xi_\ell}^T\mathcal{D}^{-1/2} \ .$$

Thus,

$$\sum_\ell \mathsf{Var}(Y_\ell) = \mathcal{D}^{-1/2}\left(\sum_\ell \mathsf{Var}(h_\ell)\frac{(a_{\xi_\ell}^T\mathcal{D}^{-1}a_{\xi_\ell})}{|\xi_\ell|^2}a_{\xi_\ell}a_{\xi_\ell}^T\right)\mathcal{D}^{-1/2} \ .$$

Note that for any matrix B, $\mathcal{D}^{-1/2}B\mathcal{D}^{-1/2}$ and $\mathcal{D}^{-1}B$ have same eigenvalues, and hence, using Gerschgorin's theorem (Stewart and Sun, 1990), one has

$$\begin{aligned}\left\|\sum_\ell \mathsf{Var}(Y_\ell)\right\|_2 &\leq \max_{1\leq i\leq n}\frac{1}{\mathcal{D}_{ii}}\sum_{j=1}^n\left(\sum_\ell \mathsf{Var}(h_\ell)\frac{(a_{\xi_\ell}^T\mathcal{D}^{-1}a_{\xi_\ell})}{|\xi_\ell|^2}a_{\xi_\ell}a_{\xi_\ell}^T\right)_{ij} \\ &= \max_{1\leq i\leq n}\frac{1}{\mathcal{D}_{ii}}\sum_\ell \mathsf{Var}(h_\ell)\frac{(a_{\xi_\ell}^T\mathcal{D}^{-1}a_{\xi_\ell})}{|\xi_\ell|^2}(a_{\xi_\ell})_i\sum_{j=1}^n(a_{\xi_\ell})_j \ .\end{aligned}$$

Observing that $a_{\xi_\ell}^T \mathcal{D}^{-1} a_{\xi_\ell} \leq \frac{a_{\xi_\ell}^T a_{\xi_\ell}}{d}$ and $|\xi_\ell| = \sum_j (a_{\xi_\ell})_j = a_{\xi_\ell}^T a_{\xi_\ell}$, we have

$$\left\| \sum_\ell \mathsf{Var}(Y_\ell) \right\|_2 \leq \frac{1}{d} \max_{1 \leq i \leq n} \frac{1}{\mathcal{D}_{ii}} \sum_\ell \mathsf{Var}(h_\ell)(a_{\xi_\ell})_i \leq \frac{1}{d}$$

since $\mathsf{Var}(h_\ell) = \mathsf{E}[h_\ell](1 - \mathsf{E}[h_\ell]) \leq \mathsf{E}[h_\ell]$. Similarly, one can also compute

$$\|Y_\ell\|_2 \leq |h_\ell - \mathsf{E}[h_\ell]| \frac{1}{|\xi_\ell|} \|\mathcal{D}^{-1/2} a_{\xi_\ell} a_{\xi_\ell}^T \mathcal{D}^{-1/2}\|_2 \leq \frac{|a_{\xi_\ell}^T \mathcal{D}^{-1} a_{\xi_\ell}|}{|\xi_\ell|} \leq \frac{1}{d},$$

where second inequality holds since $h_\ell \in \{0, 1\}$ and $\mathcal{D}^{-1/2} a_{\xi_\ell} a_{\xi_\ell}^T \mathcal{D}^{-1/2}$ is a rank-1 matrix. Substituting above bounds in (5.36) and noting that $\frac{9 \ln n}{d} < 1$, we have

$$\mathsf{P}\left(\|\mathcal{L} - \widehat{L}\|_2 \geq 3\sqrt{\frac{\ln n}{d}} \right) \leq 2n \exp\left(- \frac{9\frac{\ln n}{d}}{\frac{2}{d} + \frac{1}{d}} \right) = \frac{2}{n^2},$$

which proves (5.35). To bound the other term in (5.34), we note that

$$\begin{aligned} \|L - \widehat{L}\|_2 &\leq \|\mathcal{D}^{-1/2} A \mathcal{D}^{-1/2} - D^{-1/2} A D^{-1/2}\|_2 \\ &\leq \|(\mathcal{D}^{-1/2} - D^{-1/2}) A \mathcal{D}^{-1/2} + D^{-1/2} A (\mathcal{D}^{-1/2} - D^{-1/2})\|_2 \\ &\leq \|(\mathcal{D}^{-1} D)^{1/2} - I\|_2 \|(D\mathcal{D}^{-1})^{1/2}\|_2 + \|(\mathcal{D}^{-1} D)^{1/2} - I\|_2. \end{aligned}$$

In above, we use the fact that $D_{ii} = \sum_j A_{ij}$ to conclude that $\|D^{-1/2} A D^{-1/2}\|_2 = 1$. Note that $\mathcal{D}^{-1} D$ is a diagonal matrix with non-negative diagonal entries, and hence,

$$\|(\mathcal{D}^{-1} D)^{1/2} - I\|_2 = \max_{1 \leq i \leq n} \left| \sqrt{\frac{D_{ii}}{\mathcal{D}_{ii}}} - 1 \right| \leq \max_{1 \leq i \leq n} \left| \frac{D_{ii}}{\mathcal{D}_{ii}} - 1 \right|,$$

where the inequality follows from the fact that $|\sqrt{x} - 1| \leq |x - 1|$ for all $x \geq 0$. We now claim that for all $i = 1, \ldots, n$,

$$\mathsf{P}\left(|D_{ii} - \mathcal{D}_{ii}| > 3\mathcal{D}_{ii} \sqrt{\frac{\ln n}{d}} \right) \leq \frac{2}{n^3}. \tag{5.37}$$

Hence, with probability at least $(1 - \frac{2}{n^2})$,

$$\max_{1 \leq i \leq n} \left| \frac{D_{ii}}{\mathcal{D}_{ii}} - 1 \right| \leq 3\sqrt{\frac{\ln n}{d}}.$$

From above and the relation $\|(D\mathcal{D}^{-1})^{1/2}\|_2 \leq 1 + \|(D\mathcal{D}^{-1})^{1/2} - I\|_2$, we have

$$\|L - \widehat{L}\|_2 \leq \frac{9\ln n}{d} + 6\sqrt{\frac{\ln n}{d}} \leq 9\sqrt{\frac{\ln n}{d}}$$

where the last inequality holds since $3\sqrt{\frac{\ln n}{d}} < 1$. The lemma follows by combining above bound with (5.35).

Finally, we prove (5.37). Since, $D_{ii} = \sum_{\ell} h_\ell(a_{\xi_\ell})_i = \sum_{\ell: i \in \xi_\ell} h_\ell$, we use Bernstein inequality to write

$$\begin{aligned}
\mathsf{P}\left(|D_{ii} - \mathcal{D}_{ii}| > 3\mathcal{D}_{ii}\sqrt{\frac{\ln n}{d}}\right) &= \mathsf{P}\left(\left|\sum_{\ell: i\in\xi_\ell} (h_\ell - \mathsf{E}[h_\ell])\right| > 3\mathcal{D}_{ii}\sqrt{\frac{\ln n}{d}}\right) \\
&\leq 2\exp\left(\frac{-\frac{9\mathcal{D}_{ii}^2 \ln n}{d}}{2\sum_{\ell: i\in\xi_\ell} \mathsf{Var}(h_\ell) + 2\mathcal{D}_{ii}\sqrt{\frac{\ln n}{d}}}\right) \\
&\leq 2\exp\left(-\frac{3\mathcal{D}_{ii}\ln n}{d}\right)
\end{aligned}$$

for $d > 9\ln n$. Since, $\mathcal{D}_{ii} \geq d$, we obtain (5.37).

5.A.3 Proofs for results in Section 5.4

Proof of Corollary 5.12

We observe that for the specified model, the matrix $G \in \mathbb{R}^{k\times k}$, as defined in Lemma 5.5, is given by

$$G_{ij} = \begin{cases} \frac{p\alpha_m}{m}\binom{\frac{n}{k}-2}{m-2} + \frac{q\alpha_m}{m}\binom{n-2}{m-2} & \text{if } i = j, \\ \frac{q\alpha_m}{m}\binom{n-2}{m-2} & \text{if } i \neq j. \end{cases}$$

Thus, G is of the form $G = aI + b\mathbf{1}$, where $\mathbf{1}$ is constant matrix of ones. It is easy to verify that for such a matrix, the minimum eigenvalue is a. Hence, we have $\lambda_{\min}(G) = \frac{p\alpha_m}{m}\binom{\frac{n}{k}-2}{m-2}$. Also,

we can compute

$$d = \left(p\alpha_m \binom{\frac{n}{k}-1}{m-1} + q\alpha_m \binom{n-1}{m-1}\right) > q\alpha_m \binom{n-1}{m-1} . \tag{5.38}$$

Note that since the vertex classes are balanced and vertex degrees behave identically, the second term in (5.15) is zero, and we can compute δ as

$$\delta = \frac{np\alpha_m}{kdm}\binom{\frac{n}{k}-2}{m-2} \geq \frac{n}{km}\frac{p\alpha_m\binom{\frac{n}{k}-2}{m-2}}{(p+q)\alpha_m\binom{n-1}{m-1}} \geq \frac{p(r-1)}{m(p+q)}\frac{k\binom{n/k}{m}}{\binom{n}{m}} , \tag{5.39}$$

where the first inequality follows by observing that $d \leq (p+q)\alpha_m\binom{n-1}{m-1}$. Now, observe that under the given condition on α_m, we have

$$\delta^2 d \geq C'\frac{\alpha_m}{n}\binom{n}{m}\left(\frac{k\binom{n/k}{m}}{\binom{n}{m}}\right)^2 \geq C''k^{2m-1}(\ln n)^2\left(\frac{k\binom{n}{k}^m}{n^m}\right)^2 \geq C''k(\ln n)^2,$$

where C'' is a constant depending only on C, p, q and m, that is obtained from (5.38) and (5.39). The last inequality uses the relation $\frac{a^b}{4(b!)} \leq \binom{a}{b} \leq \frac{a^b}{b!}$. Choosing C sufficiently large, the condition of Theorem 5.6 is satisfied, and hence, we can conclude from Theorem 5.6 that $\text{Error}_{\text{NH-Cut}}(\psi, \psi') = O\left(\frac{n\ln n}{\delta^2 d}\right)$, which simplifies to the stated claim.

Proof of Corollary 5.13

The proof is quite similar to that of Corollary 5.12 since the matrices G and $\mathcal{D}$ are linear combinations of the corresponding matrices for the m-uniform hypergraphs. We compute

$$\lambda_{\min}(G) = \sum_{m=2}^{M}\frac{p\alpha_m}{m}\binom{\frac{n}{k}-2}{m-2} \geq \frac{p\alpha_r}{r}\binom{\frac{n}{k}-2}{r-2} ,$$

where $\theta_r > 0$ and we ignore all terms for $m > r$. Substituting the relation for α_m, we can write

$$\lambda_{\min}(G) \geq C_1\frac{\theta_r n^{a-2}(\ln n)^b}{k^{r-2}}$$

for some constant C_1 depending on p, q and r. Also,

$$\begin{aligned} d &= \sum_{m=2}^{M} \left(p\alpha_m \binom{\frac{n}{k}-1}{m-1} + q\alpha_m \binom{n-1}{m-1} \right) \\ &> q \sum_{m=2}^{M} \frac{m}{n} \alpha_m \binom{n}{m} \geq n^{a-1} (\ln n)^b q \sum_{m=2}^{M} m\theta_m \ . \end{aligned}$$

Similarly, one can verify that $d \leq (p+q) n^{a-1} (\ln n)^b \sum_{m=2}^{M} m\theta_m$. Thus,

$$\delta^2 d \geq \frac{n^2 (\lambda_{\min}(G))^2}{k^2 d} \geq \frac{C_1^2 \theta_r^2}{(p+q)} \frac{n^{a-1} (\ln n)^b}{k^{2r-2} \sum_{m=2}^{M} m\theta_m} \ , \tag{5.40}$$

where the inequality follows from above bounds on $\lambda_{\min}(G)$ and d. Under the condition (5.26) with large enough C, the condition of Theorem 5.6 holds and the claim follows.

Proofs for Lemmas 5.14 and 5.15

As in the previous corollaries, we can say that, for both cases, the second term in (5.15) is zero. Hence, $\delta > 0$ if and only if $\lambda_{\min}(G) > 0$. For the setting of Lemma 5.14, one can compute

$$G_{ij} = \begin{cases} \dfrac{\alpha_3}{3} \left[p_1 \left(\dfrac{n}{k} - 2 \right) + p_2 \left(n - \dfrac{n}{k} \right) \right] & \text{if } i = j, \\ \dfrac{\alpha_3}{3} \left[p_2 \left(\dfrac{2n}{k} - 2 \right) + p_3 \left(n - \dfrac{2n}{k} \right) \right] & \text{if } i \neq j. \end{cases}$$

Using an observation made in the proof of Corollary 5.12, we have $\lambda_{\min}(G) = G_{11} - G_{12}$, and (5.27) is equivalent to stating $G_{11} > G_{12}$.

The same arguments are valid for Lemma 5.15, where G is of the form

$$G_{ij} = \begin{cases} \dfrac{1}{2} + \dfrac{\alpha_3}{3} \left[p_1 \left(\dfrac{n}{k} - 2 \right) + p_2 \left(n - \dfrac{n}{k} \right) \right] & \text{if } i = j, \\ \dfrac{\alpha_3}{3} \left[p_2 \left(\dfrac{2n}{k} - 2 \right) + p_3 \left(n - \dfrac{2n}{k} \right) \right] & \text{if } i \neq j. \end{cases}$$

Proof of Lemma 5.16

As mentioned in Lemma 5.5, one can write $\mathcal{A}$ as $\mathcal{A} = ZGZ^T - J$. In the present case, $G \in \mathbb{R}^{2\times 2}$ is given by

$$G_{ij} = \begin{cases} \frac{1}{2m}\binom{s-2}{m-2} + \frac{1}{2m}\binom{n-2}{m-2} & \text{if } i = j = 1, \\ \\ \frac{1}{2m}\binom{n-2}{m-2} & \text{otherwise.} \end{cases}$$

One can also verify that the diagonal matrices $\mathcal{D}$ and J are given by

$$\mathcal{D}_{ii} = \begin{cases} \frac{1}{2}\binom{s-1}{m-1} + \frac{1}{2}\binom{n-1}{m-1} & \text{if } i \in s\text{-clique}, \\ \\ \frac{1}{2}\binom{n-1}{m-1} & \text{otherwise.} \end{cases}$$

and

$$J_{ii} = \begin{cases} -\frac{1}{2m}\binom{s-2}{m-1} - \frac{1}{2m}\binom{n-2}{m-1} & \text{if } i \in s\text{-clique}, \\ \\ -\frac{1}{2m}\binom{n-2}{m-1} & \text{otherwise.} \end{cases}$$

Hence, the claim follows by substituting above relations in (5.15). It is more convenient to write (5.15) using the notations $\widetilde{\mathcal{D}}$ and $\widetilde{J}$ defined in the proof of Lemma 5.10, and it can be written as

$$\delta = \frac{s\lambda_{\min}(G)}{\widetilde{\mathcal{D}}_{11}} - \left|\frac{\widetilde{J}_{11}}{\widetilde{\mathcal{D}}_{11}} - \frac{\widetilde{J}_{22}}{\widetilde{\mathcal{D}}_{22}}\right|, \tag{5.41}$$

where we use the fact $s < (n-s)$. Note that G has non-negative eigenvalues, and we can use the following relation for 2×2 matrices

$$\lambda_{\min}(G) \geq \frac{\det(G)}{\text{Trace}(G)} = \frac{\frac{1}{2m}\binom{s-2}{m-2}\binom{n-2}{m-2}}{\binom{s-2}{m-2} + 2\binom{n-2}{m-2}} \geq \frac{1}{6m}\binom{s-2}{m-2}.$$

Combining this with the observation $\widetilde{\mathcal{D}}_{11} \leq \binom{n-1}{m-1}$, we can argue that the first term in (5.41) is at least

$$\frac{s\binom{s-2}{m-2}}{6m\binom{n-1}{m-1}} \geq C_1 \left(\frac{s}{n}\right)^{m-1}$$

for some $C_1 > 0$. On the hand, the second term in (5.41) can be computed as

$$\begin{aligned}
\left|\frac{\widetilde{J}_{11}}{\widetilde{\mathcal{D}}_{11}} - \frac{\widetilde{J}_{22}}{\widetilde{\mathcal{D}}_{22}}\right| &= \left|\frac{\binom{s-1}{m-1}\binom{n-2}{m-1} - \binom{n-1}{m-1}\binom{s-2}{m-1}}{\binom{n-1}{m-1}\left[\binom{n-1}{m-1} + \binom{s-1}{m-1}\right]}\right| \\
&\leq \frac{\binom{s-1}{m-1}}{\binom{n-1}{m-1}} \left(\frac{n-m}{n-1} - \frac{s-m}{s-1}\right) \\
&\leq \frac{\binom{s-1}{m-1}}{\binom{n-1}{m-1}} \left(\frac{m-1}{s-1}\right) \leq \frac{C_2}{s} \left(\frac{s}{n}\right)^{m-1}
\end{aligned}$$

for some $C_2 > 0$. From above, one can conclude that $\delta > 0$ when $C_1 > \frac{C_2}{s}$, where both C_1 and C_2 depend only on m. Hence, the claim.

Efficiency is doing the thing right. Effectiveness is doing the right thing.

Peter Drucker

Chapter 6

Edge Sampling for Hypergraphs

In this chapter, we focus on the complexity of hypergraph partitioning algorithms, and study efficient modifications. We develop our studies based on Algorithm TTM, but similar discussions also for the other approaches as well. Section 6.1 presents a consistency result for TTM, when few edges are sampled. The result is proved in Appendix 6.A. The intuition gathered from our analysis of sampled TTM is then used in Section 6.2 to present an efficient spectral algorithm for the subspace clustering problem. Numerical results demonstrating the merits of this method can be found in Chapter 8.

We begin this chapter with the computational complexity of Algorithm TTM. Note that the k-means approach of (Ostrovsky et al., 2012) has a complexity of $O(k^2n + k^4)$ since the data is embedded in a k-dimensional space. Furthermore, Steps 2 to 4 involve only matrix operations with the eigenvector computation being the most expensive operation. One may compute the k dominant eigenvectors using power iterations, which can be done provably in $O(kn^2 \ln(kn))$ runtime (Boutsidis et al., 2015). However, the computational bottleneck of the algorithm is Step 1, which owing to the representation in (4.20) has complexity of $O(m^2|\mathcal{E}|)$. This is particularly challenging in computer vision applications, such as subspace clustering or matching, where all edges have non-zero weights, no matter how small these weights may be. Thus, in such applications, $|\mathcal{E}| = \binom{n}{m} = \Theta(n^m)$. This exponential dependence on m makes *exact* hypergraph partitioning algorithms impractical, particularly when one considers higher order relations, for instance $m = 8$ used in (Govindu, 2005; Jain and Govindu, 2013). The natural alternative is to sample few edges of the hypergraph, or equivalently few entries of the adjacency tensor $\mathbf{A}$ (Govindu, 2005; Chen and Lerman, 2009; Duchenne et al., 2011).

In the context of TTM, we may formally express this approach as sampling edges from $\mathcal{E}$ according to some probability distribution $(p_e)_{e\in\mathcal{E}}$, and computing a sample estimate $\widehat{A}$ instead of A. A requirement of the estimator should be its unbiasedness, *i.e*, $\mathsf{E}[\widehat{A}] = A$, where the

expectation is with respect to the sampling distribution given an instance of the hypergraph. Based on (4.20), we propose to use an unbiased estimator of the form

$$\widehat{A} = \frac{(m-2)!}{|\mathcal{I}|} \sum_{e \in \mathcal{I}} \frac{w_e}{p_e} R_e \ , \tag{6.1}$$

where $\mathcal{I}$ is a sub-collection of edges from $\mathcal{E}$ sampled with replacement. Subsequently. one can define $\widehat{D}$ as an unbiased estimate of D, *i.e*, $\widehat{D}_{ii} = \sum_j \widehat{A}_{ij}$. The remaining steps of Algorithm TTM may be carried using the estimated normalized adjacency matrix $\widehat{D}^{-1/2}\widehat{A}\widehat{D}^{-1/2}$. Such an approach has a runtime of $O(m^2 N + kn^2 \ln(kn) + k^2 n + k^4)$, where $N = |\mathcal{I}|$.

6.1 Consistency of hypergraph partitioning with edge sampling

The key question addressed in this section is the minimum number (N) of edges required to be sampled to achieve consistency of such a variant of Algorithm TTM. The following result provides a precise answer to this question.

Theorem 6.1. *Let $(\mathcal{V}, \mathcal{E}, w)$ be a m-uniform hypergraph on $|\mathcal{V}| = n$ vertices generated from a random model with k planted classes, where n is sufficiently large and the cluster sizes are $n_1 \geq n_2 \geq \ldots \geq n_k$. Assume that the approximate k-means method is used. Define $d = \min_{1 \leq i \leq n} \mathsf{E}[\deg(i)]$, and let δ be as defined in (4.18).*

Let N edges be sampled with replacement according to probability distribution $(p_e)_{e \in \mathcal{E}}$, and let $\beta > 0$ be such that $\max_{e \in \mathcal{E}} \frac{w_e}{p_e} \leq \beta$ with probability $(1 - o(1))$. There exist absolute constants $C, C' > 0$, such that, if $\delta > 0$,

$$d > C \frac{k n_1 (\ln n)^2}{n_k \delta^2} \quad \text{and} \quad N > C' \left(1 + \frac{2\beta}{d}\right) \frac{k n_1 (\ln n)^2}{n_k \delta^2} \ , \tag{6.2}$$

then with probability $(1 - o(1))$, the sampled variant of TTM algorithm achieves an error

$$\mathrm{Error}_{\mathrm{TTM}}(\psi, \psi') = O\left(\frac{k n_1 \ln n}{\delta^2} \left(\frac{1}{d} + \frac{1}{N} + \frac{2\beta}{Nd}\right)\right) = o(n). \tag{6.3}$$

The above result is similar to Theorem 4.3 except for an additional condition associated with the number of edges to be sampled. However, we note that the proof of Theorem 6.1 shows that the constant C in (6.2) need to be larger than the corresponding term in Theorem 4.3 by

a certain factor.

The interesting observation in the above result is that if a particular sampling strategy is used, one can note that relatively smaller number of samples N would be required if the hypergraph is more dense (larger d). This fact suggests that there is a high correlation among the information provided by different edges, and hence, using only a small subset of them suffices. On the other hand, for a sparse hypergraph, most edges have zero or negligibly small values, and one can hardly gather information about the true partition from few edges.

Obviously, the bound also depends on the sampling technique, and (6.2) suggests that a better sampling distribution is one for which β is smaller. To clarify this observation, we state the result for two particular sampling distributions: (i) uniform sampling, and (ii) sampling each edge e with probability proportional to its weight, *i.e*,

$$p_e = \frac{w_e}{\sum\limits_{e' \in \mathcal{E}} w_{e'}} \qquad \text{for all } e \in \mathcal{E}. \tag{6.4}$$

For ease of exposition, we elaborate on the effect of these sampling techniques for the particular model described in Section 4.3.2.

Corollary 6.2. *Consider the setting described in Section 4.3.2. Define quantity s such that $\xi = 1$ for uniform sampling, and $\xi = \alpha_m$ for the weighted sampling of (6.4). If there exist constants $C, C' > 0$ such that*

$$\alpha_m > C\frac{k^{2m-1}(\ln n)^2}{n^{m-1}} \qquad \text{and} \qquad N > C'\frac{\xi n k^{2m-1}(\ln n)^2}{\alpha_m}\,, \tag{6.5}$$

then $\text{Error}_{\text{TTM}}(\psi, \psi') = o(n)$ *with probability* $(1 - o(1))$.

The above result shows that in the most dense regime ($\alpha_m = 1$), both sampling techniques have a similar performance. For instance, if $k = O\left(\frac{n^{1/4}}{\ln n}\right)$, as considered in Corollary 4.4, one needs to sample $N = \Omega\left(n^{0.5m+0.75}(\ln n)^{3-2m}\right)$ edges to achieve weak consistency. However, in the case of sparse hypergraphs, Corollary 6.2 clearly indicates that the sampling distribution of (6.4) achieves a runtime that is smaller than that of uniform sampling by a factor of α_m. In fact, in the setting of Corollary 4.5, the above result shows that using the weighted sampling strategy, one needs to sample only $N = \Omega\left(n(\ln n)^{2m+1}\right)$ edges for consistent partitioning. On the other hand, consistency is achieved with uniform sampling only if one uses $N = \Omega(n^m)$ edges, *i.e*, at least a constant fraction of all the edges[1]. On the whole, we can conclude that the weighted sampling

[1] Note that this observation is only true for weighted hypergraphs, where a small α_m implies that most edges have very small, but positive, weights. On the other hand, unweighted hypergraphs with small α_m implies

described in (6.4) helps to reduce the complexity of Step 1 of Algorithm TTM by a factor of $\frac{k^{2m-1}(\ln n)^2}{n^{m-1}}$. A related weighted sampling has been suggested in the matrix literature (Drineas et al., 2006) in the context of column sampling for matrix operations, where the authors show that one should sample columns of a matrix with probability proportional to their norm.

It is obvious that specifying the distribution (p_e) with $p_e \propto w_e$ involves computing all edge weights, and hence, it is an impractical solution for the problem at hand. However, this result leads to an important conclusion – *sample edges with larger weights more frequently*. This is essentially the idea commonly used in most tensor based algorithms. In the case of matching algorithms, one uses an efficient k nearest neighbor search to sample the larger tensor entries (Duchenne et al., 2011). On the other hand, the subspace clustering literature has acknowledged the idea of iterative sampling (Chen and Lerman, 2009; Jain and Govindu, 2013), where one uses an alternating strategy of finding clusters using a sampled set of edges, and then re-sampling edges for which at least $(m-1)$ vertices belong to a cluster. It is clear that both sampling techniques give higher preference to larger edge weights, and hence, as a consequence of Corollary 6.2, both methods are expected to perform better than uniform sampling. Thus Corollary 6.2 provides a theoretical justification for why such heuristics work, thereby answering an open question posed by Chen and Lerman (2009).

6.2 Efficient uniform hypergraph partitioning algorithm

In Corollary 4.6, we observed that for dense hypergraphs, TTM achieves a smaller error bound compared to a a spectral technique (Govindu, 2005; Chen and Lerman, 2009) that relies on higher order singular value decomposition of tensors (HOSVD). Later, we also validate this conclusion in small synthetic problems. However, we empirically observed in (Ghoshdastidar and Dukkipati, 2015a) that for larger problems, naively sampled TTM algorithm is outperformed by the practical variant of HOSVD, which uses an iterative sampling technique. This practical variant of HOSVD is commonly referred to as spectral curvature clustering or SCC (Chen and Lerman, 2009). To address the paradoxical situation, we present an iterative version of Algorithm TTM, for the purpose of subspace clustering. We henceforth refer to this algorithm as *spectral tensor trace maximization with iterative sampling* (Tetris).

We present Algorithm Tetris for the subspace clustering problem described in Section 2.4.3. Recall that in this problem, one is given n r_a-dimensional vectors, each being a noisy perturbation of a vector lying in an union of k subspaces, each of dimension at most $r < r_a$. We fix the

that only few edges are present, and hence, sampling is not required in that case.

order of the tensor as $m = (r+2)$, and and the edge weights are computed using (2.21), where $f_r(\cdot)$ is as the polar curvature of m (see Equations 1-3 of Chen and Lerman, 2009). We also incorporate the convergence criteria and the estimation procedure for σ used in SCC, which are not explicitly stated below. Furthermore, to standardize the approach with SCC, Tetris uses a one-sided degree normalization and computes left singular vectors of Laplacian.

Algorithm Tetris : TTM with iterative sampling for subspace clustering

Input: Data set $Y = [Y_1, \ldots, Y_n]$; k = Number of subspaces;
r = Maximum subspace dimension; and
c = A hyperparameter controlling number of sampled edges (precisely, $N = nc$)

1: Set $m = r + 2$.
2: Uniformly sample c subsets of Y, each containing $(m-1)$ points.
3: Initialize $\widehat{A} \in \mathbb{R}^{n \times n}$ to a zero matrix.
4: **for** $j = 1$ to c **do**
5: Consider j^{th} subset of Y with the points $Y_{j_1}, \ldots, Y_{j_{m-1}}$.
6: **for** $i = 1$ to n **do**
7: Compute the weight w_e for the edge $e = \{Y_i, Y_{j_1}, \ldots, Y_{j_{m-1}}\}$ using (2.21).
8: Update $\widehat{A}_{ij_l} = \widehat{A}_{ij_l} + w_e$ for all $l = 1, \ldots, m-1$.
9: **end for**
10: **end for**
11: Let $\widehat{D} \in \mathbb{R}^{n \times n}$ be diagonal with $\widehat{D}_{ii} = \sum_{j=1}^{n} \widehat{A}_{ij}$.
12: Compute k dominant left singular vectors of $\widehat{D}^{-1}\widehat{A}$, denoted by $\widehat{X} \in \mathbb{R}^{n \times k}$.
13: Normalize rows of $\widehat{X}$ to have unit norm.
14: Run k-means on the rows of the normalized matrix, and partition Y into k clusters.
15: From each obtained cluster, sample c/k subsets, each of size $(m-1)$.
16: Repeat from Step 3, and iterate until convergence.

Output: Clustering of Y into k disjoint clusters.

6.A Proofs for results in this chapter

The discussions in this chapter consider two sources of randomness – the random model for hypergraph, and random sampling of edges (or tensor entries). Hence, we make a distinction in the notation for expectation, variance and probability by specifying the underlying measure.

Remark 6.3. In this section, we use $\mathsf{E}_H[\cdot]$ and $\mathsf{E}_S[\cdot]$ to denote the expectation with respect to distribution of the planted model, and the expectation with respect to sampling distribution, respectively. We also use the conditional expectation, $\mathsf{E}_{S|H}[\cdot]$, over sampling distribution given

a random hypergraph. Similar subscripted notations have been used for probability, $\mathsf{P}(\cdot)$, and variance, $\mathsf{Var}(\cdot)$. Note that this notation is not used in the rest of the thesis.

Proof of Theorem 6.1

Following the arguments in the proof of Theorem 4.3, one can see that it suffices to modify the statement of Lemma 4.8 only, where instead of $\|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2$, we now need to compute a bound on $\|\widehat{D}^{-1/2}\widehat{A}\widehat{D}^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2$. Observe that

$$\begin{aligned}\|\widehat{D}^{-1/2}\widehat{A}\widehat{D}^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2 \\ \leq \|\widehat{D}^{-1/2}\widehat{A}\widehat{D}^{-1/2} - D^{-1/2}AD^{-1/2}\|_2 + \|D^{-1/2}AD^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2 \,,\end{aligned}$$

where the second term is bounded due to Lemma 4.8. Thus, the purpose of this proof is to derive a bound on $\|\widehat{D}^{-1/2}\widehat{A}\widehat{D}^{-1/2} - D^{-1/2}AD^{-1/2}\|_2$ and the associated sufficient condition. To this end, we claim the following: Let β be defined as in Theorem 6.1 and $D_{\min} = \min_{1\leq i\leq n} D_{ii}$. Assume that

$$\mathcal{D}_{\min} > 36(m-1)!\ln n \quad \text{and} \quad N > 9\left(1 + \frac{2\beta(m-1)!}{\mathcal{D}_{\min}}\right)\ln n\,. \tag{6.6}$$

Also let Γ denote the event

$$\Gamma = \left\{D_{\min} > \frac{\mathcal{D}_{\min}}{2}\right\} \bigcap \left\{\max_{e\in\mathcal{E}} \frac{w_e}{p_e} \leq \beta\right\}.$$

Then, conditioned on a given random hypergraph and the event Γ, the following bounds hold with probability $(1 - \frac{2}{n^2})$,

$$\max_{1\leq i\leq n}\left|\frac{\widehat{D}_{ii}}{D_{ii}} - 1\right| \leq 3\sqrt{\frac{\ln n}{N}\left(1 + \frac{2\beta(m-1)!}{\mathcal{D}_{\min}}\right)}\,. \tag{6.7}$$

and

$$\|D^{-1/2}(\widehat{A} - A)D^{-1/2}\|_2 \leq 3\sqrt{\frac{\ln n}{N}\left(1 + \frac{2\beta(m-1)!}{\mathcal{D}_{\min}}\right)}\,. \tag{6.8}$$

Assuming that the above hold, we now derive a bound on $\|\widehat{D}^{-1/2}\widehat{A}\widehat{D}^{-1/2} - L\|_2$ in the following way. First, note that the bounds in (6.7) and (6.8) are with respect to a conditional probability measure, and need to be converted into a bound with respect to the joint probability measure

$\mathsf{P}_{S,H}$. This is not hard to derive as one can see in the case of (6.8), where one can write

$$\begin{aligned}
&\mathsf{P}_{S,H}\left(\|D^{-1/2}(\widehat{A}-A)D^{-1/2}\|_2 > 3\sqrt{\frac{\ln n}{N}\left(1+\frac{2\beta(m-1)!}{\mathcal{D}_{\min}}\right)}\right)\\
&= \mathsf{E}_H\left[\mathsf{P}_{S|H}\left(\|D^{-1/2}(\widehat{A}-A)D^{-1/2}\|_2 > 3\sqrt{\frac{\ln n}{N}\left(1+\frac{2\beta(m-1)!}{\mathcal{D}_{\min}}\right)}\right)\right]\\
&\le \mathsf{E}_{H|\Gamma}\left[\mathsf{P}_{S|H,\Gamma}\left(\|D^{-1/2}(\widehat{A}-A)D^{-1/2}\|_2 > 3\sqrt{\frac{\ln n}{N}\left(1+\frac{2\beta(m-1)!}{\mathcal{D}_{\min}}\right)}\right)\right]\mathsf{P}_H(\Gamma)+\mathsf{P}_H(\Gamma^c)\\
&= O\left(\frac{1}{n^2}\right)+\mathsf{P}_H(\Gamma^c),
\end{aligned} \tag{6.9}$$

where the inequalities follow by observing that all the quantities are smaller than one, and the first term is bounded due to (6.8). From (4.24), it follows that with probability $(1-O(n^{-2}))$,

$$D_{ii} > \mathcal{D}_{ii}\left(1-3\sqrt{\frac{(m-1)!\ln n}{\mathcal{D}_{\min}}}\right)$$

for all $i=1,\ldots,n$. Hence, if $\mathcal{D}_{\min} > 36(m-1)!\ln n$, then

$$D_{\min} > \mathcal{D}_{\min}\left(1-3\sqrt{\frac{(m-1)!\ln n}{\mathcal{D}_{\min}}}\right) > \frac{\mathcal{D}_{\min}}{2}.$$

This fact, along with the assumption on β, shows that $\mathsf{P}_H(\Gamma^c)=o(1))$, and so, the upper bound on $\|D^{-1/2}(\widehat{A}-A)D^{-1/2}\|_2$ holds with probability $(1-o(1))$ even with respect to joint probability measure. Similar result also holds for (6.7). Subsequently, we follow the arguments leading to (4.26) to conclude that

$$\begin{aligned}
&\|\widehat{D}^{-1/2}\widehat{A}\widehat{D}^{-1/2}-D^{-1/2}AD^{-1/2}\|_2\\
&\le \max_{1\le i\le n}\left|\frac{\widehat{D}_{ii}}{D_{ii}}-1\right|\left(2+\max_{1\le i\le n}\left|\frac{\widehat{D}_{ii}}{D_{ii}}-1\right|\right)+\|D^{-1/2}(\widehat{A}-A)D^{-1/2}\|_2\\
&\le 12\sqrt{\frac{\ln n}{N}\left(1+\frac{2\beta(m-1)!}{\mathcal{D}_{\min}}\right)},
\end{aligned} \tag{6.10}$$

where the last inequality holds with probability $(1-o(1))$ under the conditions stated in (6.6). This bound combined with Lemma 4.8 implies

$$\|\widehat{D}^{-1/2}\widehat{A}\widehat{D}^{-1/2} - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}\|_2 \leq 12\sqrt{\frac{(m-1)!\ln n}{\mathcal{D}_{\min}}} + 12\sqrt{\frac{\ln n}{N}\left(1+\frac{2\beta(m-1)!}{\mathcal{D}_{\min}}\right)}\,. \quad (6.11)$$

Subsequently, one can follow the analysis of TTM to arrive at the claim.

To complete the proof, we derive the bounds (6.7) and (6.8), which again relies on the use of Bernstein inequality. For this, observe that

$$\widehat{D}_{ii} = \frac{(m-2)!}{N}\sum_{j=1}^{n}\sum_{e\in\mathcal{J}}\frac{w_e}{p_e}(R_e)_{ij} = \frac{(m-1)!}{N}\sum_{e\in\mathcal{J}}\frac{w_e}{p_e}\mathbb{1}\{i\in e\}\,,$$

where for each $e\in\mathcal{J}$,

$$\begin{aligned}
\mathsf{E}_{S|H,\Gamma}\left[\frac{w_e}{p_e}\mathbb{1}\{i\in e\}\right] &= \sum_{e'\in\mathcal{E}:e'\ni i} p_{e'}\frac{w_{e'}}{p_{e'}} = \frac{D_{ii}}{(m-1)!}\,,\\
\mathsf{Var}_{S|H,\Gamma}\left[\frac{w_e}{p_e}\mathbb{1}\{i\in e\}\right] &= \sum_{e'\in\mathcal{E}:e'\ni i}\frac{w_{e'}^2}{p_{e'}} - \left(\frac{D_{ii}}{(m-1)!}\right)^2 \leq \left(\beta - \frac{D_{ii}}{(m-1)!}\right)\frac{D_{ii}}{(m-1)!}\,,
\end{aligned}$$

and almost surely with respect to $\mathsf{P}_{S|H,\Gamma}$,

$$\left|\frac{w_e}{p_e}\mathbb{1}\{i\in e\} - \frac{D_{ii}}{(m-1)!}\right| \leq \left(\beta + \frac{D_{ii}}{(m-1)!}\right)\,.$$

Define $t = 3\sqrt{\frac{\ln n}{N}\left(1+\frac{2\beta(m-1)!}{\mathcal{D}_{\min}}\right)}$. Since the samples $e\in\mathcal{J}$ are independent and identically distributed, we can use Bernstein inequality to write

$$\begin{aligned}
\mathsf{P}_{S|H,\Gamma}\left(|\widehat{D}_{ii}-D_{ii}| > tD_{ii}\right) &= \mathsf{P}_{S|H,\Gamma}\left(\left|\sum_{e\in\mathcal{J}}\frac{w_e}{p_e}\mathbb{1}\{i\in e\} - \frac{D_{ii}}{(m-1)!}\right| > \frac{NtD_{ii}}{(m-1)!}\right)\\
&\leq 2\exp\left(\frac{-\frac{N^2t^2D_{ii}^2}{(m-1)!}}{2N\left(\beta-\frac{D_{ii}}{(m-1)!}\right)\frac{D_{ii}}{(m-1)!} + \frac{2}{3}\frac{NtD_{ii}}{(m-1)!}\left(\beta+\frac{D_{ii}}{(m-1)!}\right)}\right)\\
&\leq 2\exp\left(\frac{-\frac{ND_{ii}t^2}{(m-1)!}}{\frac{2}{3}\left(4\beta - 2\frac{D_{ii}}{(m-1)!}\right)}\right) \leq \frac{2}{n^3}\,.
\end{aligned}$$

The inequalities are derived using above relations, and the defintion of Γ. From above, (6.7)

follows from union bound. To prove (6.8), observe from (6.1) that

$$D^{-1/2}\widehat{A}D^{-1/2} = \frac{1}{N}\sum_{e\in\mathcal{J}}(m-2)!\frac{w_e}{p_e}D^{-1/2}R_eD^{-1/2}$$

is a sum of independent random matrices with

$$\mathsf{E}_{S|H,\Gamma}\left[(m-2)!\frac{w_e}{p_e}D^{-1/2}R_eD^{-1/2}\right] = D^{-1/2}AD^{-1/2}$$

and

$$\begin{aligned}\left\|(m-2)!\frac{w_e}{p_e}D^{-1/2}R_eD^{-1/2} - D^{-1/2}AD^{-1/2}\right\|_2 &\leq (m-2)!\beta\|D^{-1/2}R_eD^{-1/2}\|_2 + 1\\ &\leq \left(\frac{2\beta(m-1)!}{\mathcal{D}_{\min}} + 1\right).\end{aligned}$$

The first bound uses the fact $\|D^{-1/2}AD^{-1/2}\|_2 = 1$ and the secong follows since $D_{\min} > \frac{1}{2}\mathcal{D}_{\min}$ and $\|R_e\|_2 \leq (m-1)$. We can also bound the norm of the variance term as

$$\begin{aligned}&\left\|\mathsf{E}_{S|H,\Gamma}\left[\left((m-2)!\frac{w_e}{p_e}D^{-1/2}R_eD^{-1/2} - D^{-1/2}AD^{-1/2}\right)^2\right]\right\|_2\\ &= \left\|-\left(D^{-1/2}AD^{-1/2}\right)^2 + ((m-2)!)^2\sum_{e\in\mathcal{E}}\frac{w_e^2}{p_e}D^{-1/2}R_eD^{-1}R_eD^{-1/2}\right\|_2\\ &\leq 1 + \frac{((m-2)!)^2\beta}{D_{\min}}\left\|\sum_{e\in\mathcal{E}}w_eD^{-1}(R_e)^2\right\|_2 \leq \left(1 + \frac{2\beta(m-1)!}{\mathcal{D}_{\min}}\right).\end{aligned}$$

Using these relations and the matrix Bernstein inequality, the bound in (6.8) can be derived quite similar to the derivation of (4.25).

Proof of Corollary 6.2

Note that $\beta \geq \max_e \frac{w_e}{p_e}$. Since, $|\mathcal{E}| = \binom{n}{m}$, it follows that for uniform sampling $p_e = \binom{n}{m}^{-1}$ for all e, and hence, an appropriate choice of $\beta = \binom{n}{m}$. On the other hand, for the sampling in (6.4), $\max_e \frac{w_e}{p_e} = \sum_e w_e$. Using Bernstein inequality, one may easily bound this term from above by $2\sum_e \mathsf{E}_H[w_e] \leq 2\alpha_m\binom{n}{m}$, where the bound holds with probability $(1-n^{-2})$. Thus, ignoring constants factors, one may set $\beta = \xi n^m$, where $\xi = 1$ for uniform sampling and α_m for weighted sampling. The conditions in (6.5) follow directly from (6.2) and d, δ computed in the proof of Corollary 4.4. Here, we set $\epsilon_n = (\ln n)^{-1/2}$. The weak consistency is proved by simply substituting the lower bounds of α_m and N in the error bound of Theorem 6.1.

Why do two colors, put one next to the other, sing?

Pablo Picasso

Chapter 7

Coloring Bipartite Hypergraphs

We briefly discussed the hypergraph weak coloring problem in Section 2.4.4. The present chapter is dedicated to this problem for the case of two colors. We begin with a description of the problem at hand, and a review the existing literature.

7.1 Weak 2-coloring of hypergraphs

A hypergraph $(\mathcal{V}, \mathcal{E})$ is said to be bipartite or weakly 2-colorable if the vertex set $\mathcal{V}$ can be partitioned into two disjoint sets $\mathcal{V}_1$ and $\mathcal{V}_2$ such that every edge $e \in \mathcal{E}$ has non-empty intersections with both sets. In the case of graphs, where each edge is of size two, one can easily find the two sets by breadth first search. However, the problem turns out to be notoriously hard if edges of size more than two are present. In fact, in the case of bipartite 3-uniform and 4-uniform hypergraphs, it is well known that the problem is NP-hard (Dinur et al., 2005; Khot and Saket, 2014).

In general, finding a 2-coloring is relatively easy if a hypergraph consists of very few edges. In an answer to a question asked by Erdös (1963) on 2-colorability of uniform hypergraphs, it is now known that for large m, any m-uniform hypergraph on n vertices with at most $2^m 0.7\sqrt{\frac{m}{\ln m}}$ edges is 2-colorable (Radhakrishnan and Srinivasan, 1998). As pointed by Radhakrishnan and Srinivasan (1998), the result can be extended to non-uniform hypergraphs with minimum edge size m. However, it is much worse if the restriction on the minimum edge size and the number of edges is not imposed. Even when a hypergraph is 2-colorable, the best known algorithms (Alon et al., 1996; Chen and Frieze, 1996) require $O\left((n \ln n)^{1-1/M}\right)$ colors to properly color the hypergraph in polynomial time, where M is the maximum edge size, also called dimension, of the hypergraph. In recent years, 2-colorability of random hypergraphs has also

received considerable attention. Through a series of works (Achlioptas and Coja-Oghlan, 2008; Coja-Oghlan and Zdeborová, 2012; Panagiotou and Coja-Oghlan, 2012), it is now established that random uniform hypergraphs are 2-colorable only when the number of edges are at most Cn, for some constant $C > 0$. Thus, it is evident that coloring relatively dense hypergraphs is difficult unless the hypergraph admits a "nice" structure.

In spite of the hardness of the problem, there are a number of applications that require hypergraph coloring algorithms. For instance, such algorithms have been used for approximate DNF counting (Lu, 2004), as well as in various resource allocation and scheduling problems (Capitanio et al., 1995; Ahuja and Srivastava, 2002). The connection between "Not-All-Equal" (NAE) SAT and hypergraph 2-coloring also demonstrate its significance in context of satisfiability problems. Among the various approaches studied in the literature, perhaps the only known non-probabilistic instances of efficient 2-coloring are in the cases where the hypergraph is α-dense, 3-uniform and bipartite (Chen and Frieze, 1996), or where the hypergraph is m-uniform and its every edge has equal number of vertices of either colors (McDiarmid, 1993).

In this chapter, we consider the problem of coloring random non-uniform hypergraphs of range M, that have an underlying planted bipartite structure. We address the following question.

> **Question 4.** Does there exist a polynomial time algorithm that can color a bipartite random hypergraph with only two colors?

We answer the above question affirmatively by presenting a polynomial time spectral algorithm in Section 7.2 that can properly 2-color instances of the random hypergraph with high probability whenever the expected number of edges is at least $Cn \ln n$ for an absolute constant $C > 0$. This result is formally stated in Section 7.3, and proved using certain lemmas. The proofs of the lemmas can be found in Appendix 7.A.

To the best of our knowledge, a similar model has been only considered by Chen and Frieze (1996), who extended a graph coloring approach of Alon and Kahale (1997) to present an algorithm for 2-coloring of 3-uniform bipartite hypergraphs with Cn number of edges. To this end, our work generalizes the results of (Chen and Frieze, 1996) to non-uniform hypergraphs, and it is the first algorithm that is guaranteed to properly color non-uniform bipartite hypergraphs using only two colors. However, the model can also be derived from the one recently studied by Florescu and Perkins (2016).

7.2 Spectral algorithm for hypergraph coloring

The coloring algorithm presented below is similar, in spirit, to the spectral methods of (Alon and Kahale, 1997; Chen and Frieze, 1996), but certain key differences exist, which are essential to deal with non-uniform hypergraphs. Given a hypergraph $(\mathcal{V}, \mathcal{E})$, an initial guess of the color classes is formed by exploiting the spectral properties of a certain matrix $A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ defined as

$$A_{ij} = \begin{cases} \sum_{e\in\mathcal{E}:e\ni i,j} \frac{1}{|e|} & \text{if } i \neq j, \text{ and} \\ \sum_{e\in\mathcal{E}:e\ni i} \frac{1}{|e|} & \text{if } i = j. \end{cases} \tag{7.1}$$

The above matrix was previously used in Algorithm NH-Cut described in Chapter 5, and is also known to be related to the affinity matrix of the star expansion of hypergraph (Agarwal et al., 2006). The use of matrix A is in contrast to the adjacency based graph construction of (Chen and Frieze, 1996) that is likely to result in a complete graph if the hypergraph is dense.

Algorithm COLOR – Colors a non-uniform hypergraph H:

1: Define the matrix A as in (7.1).
2: Compute $x^A = \arg\min_{\|x\|_2=1} x^T A x$.
3: Let $T = \lceil \log_2 n \rceil$, $\mathcal{V}_1^{(0)} = \{i \in \mathcal{V} : x_i^A \geq 0\}$ and $\mathcal{V}_2^{(0)} = \{i \in \mathcal{V} : x_i^A < 0\}$.
4: **for** $t = 1, 2, \ldots, T$ **do**
5: Let $\mathcal{V}_1^{(t)} = \left\{ i \in \mathcal{V} : \sum_{j\in\mathcal{V}_1^{(t-1)}\setminus\{i\}} A_{ij} < \sum_{j\in\mathcal{V}_2^{(t-1)}\setminus\{i\}} A_{ij} \right\}$, and $\mathcal{V}_2^{(t)} = \mathcal{V}\setminus\mathcal{V}_1^{(t)}$.
6: **end for**
7: **if** $\exists e \in \mathcal{E}$ such that $e \subset \mathcal{V}_1^{(T)}$ or $e \subset \mathcal{V}_2^{(T)}$ **then**
8: Algorithm FAILS.
9: **else**
10: 2-Color $\mathcal{V}$ according to the sets $\mathcal{V}_1^{(T)}, \mathcal{V}_2^{(T)}$.
11: **end if**

The later stage of the algorithm considers an iterative procedure that is similar to (Alon and Kahale, 1997; Chen and Frieze, 1996), but uses a weighted summation of neighbors. Such weighting is crucial while dealing with edges of different sizes.

7.3 Analysis of coloring algorithm

Before stating the main result of this chapter, we present the planted model under consideration, which is based on the model described in Chapter 5. A random hypergraph $(\mathcal{V}, \mathcal{E})$ is generated on $2n$ vertices. The set of vertices is $\mathcal{V} = \{1, 2, \ldots, 2n\}$, which is arbitrarily split into two sets, each of size n, and the sets are colored with two different colors. Given an integer M, and $\alpha_2, \ldots, \alpha_M \in [0, 1]$, the edges of the hypergraph are randomly added in the following way. All the edges of size at most M are added independently, and for any $e \subset \mathcal{V}$,

$$\mathsf{P}(e \in E) = \begin{cases} \alpha_m & \text{if } e \text{ is not monochromatic and } |e| = m, \\ 0 & \text{otherwise.} \end{cases}$$

Note that M and α_m are allowed to vary with n. With respect to the model in Chapter 5, one may note here that we fix the tensor $\mathbf{B}^{(m)}$ to be binary valued, and hence, the edge probabilities are controlled by the sparsity factors α_m, $m = 2, \ldots, M$. We prove the following result.

Theorem 7.1. *Assume $M = O(1)$, and let a bipartite hypergraph $(\mathcal{V}, \mathcal{E})$ of range M be generated from the above model. There is a constant $C > 0$ such that if*

$$\sum_{m=2}^{M} \alpha_m \binom{2n}{m} \geq Cn \ln n, \tag{7.2}$$

then Algorithm COLOR finds a proper 2-coloring of the hypergraph with probability $(1 - o(1))$.

It is easy to see that the expected number of edges in the hypergraph grows as $\sum_{m=2}^{M} \alpha_m \binom{2n}{m}$, and so the condition may be stated in terms of expected number of edges.

7.3.1 A note on the assumptions in Theorem 7.1

The key assumptions made in this chapter are the following:

1. $M = O(1)$, and
2. $\alpha_2, \ldots, \alpha_M$ are such that the expected number of edges is larger than $Cn \ln n$, where $C > 0$ is a large constant.

The assumption $M = O(1)$ is crucial, and helps to ensure that C can be chosen to be a constant. This can be avoided if C is allowed to increase with n appropriately. We note that in Chapter 5, we allow M to grow with n, but impose an additional restriction so that the number of edges of larger size decay rapidly.

The second assumption is stronger than the one in (Chen and Frieze, 1996), where it was shown that a random bipartite 3-uniform hypergraph can be properly 2-colored with high probability if the expected number of edges is Cn. This is due to the use of matrix Bernstein inequality (Tropp, 2012) in our proof that does not provide useful bounds in the most sparse case. On the other hand, Chen and Frieze (1996) use the techniques of Friedman et al. (1989) that allows them to work in the most sparse regime. However, it is not clear how the same techniques can be extended even to uniform hypergraphs of higher order. Thus, it remains an open problem whether a similar result can be proved when the number of edges in the hypergraph grows linearly with n.

7.3.2 Proof of Theorem 7.1

We now prove Theorem 7.1. Without loss of generality, assume that the true color classes in $\mathcal{V}$ are $\{1, 2, \ldots, n\}$ and $\{n+1, \ldots, 2n\}$. Also, let $\mathcal{V}_{err}^{(t)}$, $t = 0, 1, \ldots, T$, denote the incorrectly colored vertices after iteration t, with $\mathcal{V}_{err}^{(0)}$ being the incorrectly colored nodes after initial spectral step. We prove Theorem 7.1 by showing with probability $(1-o(1))$, the size of $\mathcal{V}_{err}^{(T)} < 1$, which implies that all nodes are correctly colored, and hence, the hypergraph must be properly colored. The lemmas stated below are proved in Appendix 7.A. The first lemma bounds the size of $\mathcal{V}_{err}^{(0)}$, *i.e*, the error incurred at the initial spectral step.

Lemma 7.2. *If C in (7.2) is sufficiently large, then with probability $(1 - o(1))$,*

$$|\mathcal{V}_{err}^{(0)}| \leq \frac{n}{M^2 2^{2M+4}} \,. \tag{7.3}$$

Next, we analyze the iterative stage of the algorithm to characterize the vertices that are correctly colored after iteration t.

Lemma 7.3. *Define $\eta = \frac{1}{2^{M+2}} \sum_{m=2}^{M} \frac{\alpha_m (n-1)}{m} \binom{n-2}{m-2}$. For any $t \in \{1, \ldots, T\}$, if any vertex $i \in \mathcal{V}$ satisfies $\sum_{j \in \mathcal{V}_{err}^{(t-1)} \setminus \{i\}} A_{ij} < \eta$, then $P(i \in \mathcal{V}_{err}^{(t)}) \leq n^{-\Omega(C)}$.*

Note that there are only $T = \lceil \log_2 n \rceil$ iterations, and $|\mathcal{V}| = 2n$. Combining the result of Lemma 7.3 with union bound, we can conclude that if C is a large constant, then with probability $(1 - o(1))$, for all iterations $t = 1, 2, \ldots, T$, there does not exist any $i \in \mathcal{V}$ such that $\sum_{j \in \mathcal{V}_{err}^{(t-1)} \setminus \{i\}} A_{ij} < \eta$. We also make the following observation, where η is defined in Lemma 7.3.

Lemma 7.4. *With probability* $(1 - o(1))$*, there does not exist* $S_1, S_2 \subset \mathcal{V}$ *such that* $|S_1| \leq \frac{n}{M^2 2^{2M+4}}$, $|S_2| = \frac{1}{2}|S_1|$ *and for all* $i \in S_2$, $\sum_{j \in S_1 \setminus \{i\}} A_{ij} \geq \eta$.

We now use the above lemmas to proceed with the proof of Theorem 7.1. Lemma 7.2 shows that $|\mathcal{V}_{err}^{(0)}| \leq \frac{n}{M^2 2^{2M+4}}$ with probability $(1 - o(1))$. Conditioned on this event, and due to the conclusion of Lemma 7.3, one can argue that Lemma 7.4 is violated unless $|\mathcal{V}_{err}^{(t)}| < \frac{1}{2}|\mathcal{V}_{err}^{(t-1)}|$ for all iteration t with probability $(1 - o(1))$. Thus, in each iteration, the number of incorrectly colored vertices are reduced by at least half. Hence, after $T = \lceil \log_2 n \rceil$ iterations, $|\mathcal{V}_{err}^{(T)}| < 1$, which implies that all vertices are correctly colored.

7.A Proofs for lemmas in this chapter

Proof of Lemma 7.2

We view the random matrix $A \in \mathbb{R}^{2n \times 2n}$, as a perturbation of the matrix $\mathcal{A} = \mathsf{E}[A]$. Let $\mathcal{E}_0$ denote the collection of all the non-monochromatic subsets of $\mathcal{V}$ of size at most M. One can verify that for any $i, j \in \mathcal{V}$, $i \neq j$

$$\mathcal{A}_{ij} = \sum_{e \in \mathcal{E}_0 : e \ni i,j} \frac{\alpha_{|e|}}{|e|} \qquad \text{and} \qquad \mathcal{A}_{ii} = \sum_{e \in \mathcal{E}_0 : e \ni i} \frac{\alpha_{|e|}}{|e|} \, .$$

Counting the number of possible edges of each size, one can see that

$$\mathcal{A}_{ij} = \begin{cases} \beta_1 - \beta_2 & \text{if } i \neq j, \text{ and } i, j \text{ belong to same color class,} \\ \beta_1 & \text{if } i \neq j, \text{ and } i, j \text{ belong to different color class,} \\ \beta_1 - \beta_2 + \beta_3 & \text{if } i = j, \end{cases} \tag{7.4}$$

where

$$\beta_1 = \sum_{m=2}^{M} \frac{\alpha_m}{m} \binom{2n-2}{m-2}, \qquad \beta_2 = \sum_{m=2}^{M} \frac{\alpha_m}{m} \binom{n-2}{m-2},$$
$$\text{and } \beta_3 = \sum_{m=2}^{M} \frac{\alpha_m}{m} \left(\binom{2n-2}{m-1} - \binom{n-2}{m-1} \right).$$

Hence, we can write $\mathcal{A}$ as

$$\mathcal{A} = \beta_1 1_{2n \times 2n} - \beta_2 \begin{pmatrix} 1_{n \times n} & 0_{n \times n} \\ 0_{n \times n} & 1_{n \times n} \end{pmatrix} + \beta_3 I_{2n}, \tag{7.5}$$

where I_{2n} is the $2n$-dimensional identity matrix, and $1_{n\times n}$ is a $n \times n$ matrix of all 1's. One can verify that the smallest eigenvalue of $\mathcal{A}$ is $(\beta_3 - n\beta_2)$, which has multiplicity 1, and is separated from the other eigenvalues by an eigen-gap of $n\beta_2$. Moreover, the corresponding unit norm eigenvector $x^{\mathcal{A}}$ is such that $x_i^{\mathcal{A}} = \frac{1}{\sqrt{2n}}$ for all $i \leq n$, and $x_i^{\mathcal{A}} = -\frac{1}{\sqrt{2n}}$ for all $i > n$, up to a possible change of sign.

At this stage, we refer Theorem 2.8. By viewing A as a perturbation of $\mathcal{A}$ and noting that the eigen-gap $\delta = n\beta_2$, one can use the above result to conclude that if $\beta_2 > \frac{2}{n}\|A - \mathcal{A}\|_2$, then

$$\|x^A - x^{\mathcal{A}}\|_2 \leq \frac{2\sqrt{2}\|A - \mathcal{A}\|_2}{n\beta_2} \,. \tag{7.6}$$

One can write A as $A = \sum_{e\in\mathcal{E}_0} \frac{h_e}{|e|} a_e a_e^T$, where, for each set $e \in \mathcal{E}_0$, h_e is a Bernoulli$(\alpha_{|e|})$ random variable, and $a_e \in \{0,1\}^{2n}$ is such that $(a_e)_i = 1$ only when $i \in e$. Hence, one may view A as a sum of independent random matrices. Matrix concentration bounds are quite useful to derive a bound on the perturbation $\|A - \mathcal{A}\|_2$. In particular, Theorem 2.11 directly implies that

$$\mathsf{P}(\|A - \mathcal{A}\|_2 > 4\sqrt{n\beta_1 \ln n}) \leq 4n \exp\left(-\frac{16n\beta_1 \ln n}{2\|\mathsf{Var}(A)\|_2 + \frac{8}{3}\sqrt{n\beta_1 \ln n}}\right). \tag{7.7}$$

We note that choosing C in (7.2) large enough, one can satisfy $n\beta_1 > \ln n$. Also, observe that

$$\|\mathsf{Var}(A)\|_2 \leq \max_i \sum_{j=1}^{2n} (\mathsf{Var}(A))_{ij} \leq \max_i \sum_{j=1}^{2n} \mathcal{A}_{ij} \leq 4n\beta_1.$$

Substituting these in (7.7), we have

$$\begin{aligned} \mathsf{P}(\|A - \mathcal{A}\|_2 > 4\sqrt{n\beta_1 \ln n}) &\leq 4n \exp\left(-\frac{16n\beta_1 \ln n}{8n\beta_1 + \frac{8}{3}n\beta_1}\right) \\ &= \frac{4}{\sqrt{n}} = o(1). \end{aligned} \tag{7.8}$$

Thus, with probability $(1 - o(1))$ we have $\|A - \mathcal{A}\|_2 \leq 4\sqrt{n\beta_1 \ln n}$. Due to this bound, one can argue that if $\delta = n\beta_2 > 8\sqrt{\beta_1 n \ln n}$, *i.e*, $\frac{\beta_1}{\beta_2^2} < \frac{n}{64 \ln n}$, then the perturbation bound (7.6) holds.

We can compute that

$$\frac{\beta_1}{\beta_2^2} = \frac{\sum_{m=2}^{M} \frac{\alpha_m}{m}\binom{2n-2}{m-2}}{\left(\sum_{m=2}^{M} \frac{\alpha_m}{m}\binom{n-2}{m-2}\right)^2} \leq \frac{n^2 2^{2M+2}}{\sum_{m=2}^{M} \alpha_m (m-1)\binom{2n}{m}} \leq \frac{n 2^{2M+2}}{C \ln n} .$$

Hence, choosing C sufficiently large, the above bound in smaller than $\frac{n}{64 \ln n}$, and one can claim from (7.6) that

$$\|x^A - x^{\mathcal{A}}\|_2 \leq \frac{8\sqrt{2n\beta_1 \ln n}}{n\beta_2} \leq \frac{2^{M+4.5}}{\sqrt{C}} .$$

Now, we define the set $\widehat{\mathcal{V}}_{err} \subset \mathcal{V}$ as $\widehat{\mathcal{V}}_{err} = \{i \in V : |x_i^A - x_i^{\mathcal{A}}| \geq \frac{1}{\sqrt{2n}}\}$. From the definition of the color classes $\mathcal{V}_1^{(0)}, \mathcal{V}_2^{(0)}$, it directly follows that any vertex not in $\widehat{\mathcal{V}}_{err}$ must be correctly colored. Hence,

$$\begin{aligned} |\mathcal{V}_{err}^{(0)}| \leq |\widehat{\mathcal{V}}_{err}| &\leq \sum_{i \in \widehat{\mathcal{V}}_{err}} 2n |x_i^A - x_i^{\mathcal{A}}|^2 \\ &\leq 2n \|x^A - x^{\mathcal{A}}\|_2^2 = O\left(\frac{n}{C}\right), \end{aligned}$$

where the bound holds with probability $(1 - o(1))$. Thus, choosing C sufficiently large, one obtains that $|\mathcal{V}_{err}^{(0)}| \leq \frac{n}{M^2 2^{2M+4}}$.

Proof of Lemma 7.3

Consider any $i \leq n$. Note that i is correctly colored in iteration t if

$$\sum_{j \in \mathcal{V}_1^{(t-1)} \setminus \{i\}} A_{ij} < \sum_{j \in \mathcal{V}_2^{(t-1)} \setminus \{i\}} A_{ij},$$

or equivalently,

$$\sum_{j \in \mathcal{V}_1^{(t-1)} \setminus \{i\}} A_{ij} < \frac{1}{2} \sum_{j \neq i} A_{ij}. \tag{7.9}$$

Hence, it suffices to show that (7.9) holds under the condition stated in the lemma. A similar condition can be stated for $i > n$.
We note that $\sum_{j\neq i} A_{ij} = \sum_{e\in\mathcal{E}_0: e\ni i} h_e \frac{(|e|-1)}{|e|}$, and so, from Bernstein inequality, we have

$$\begin{aligned}
&\mathsf{P}\left(\sum_{j\neq i} A_{ij} \leq \left(1-\frac{1}{2^{M+2}}\right)\sum_{j\neq i}\mathcal{A}_{ij}\right)\\
&\leq \exp\left(-\frac{\frac{1}{2^{2M+4}}\left(\sum_{j\neq i}\mathcal{A}_{ij}\right)^2}{2\sum_{e\in\mathcal{E}_0: e\ni i}\frac{(|e|-1)^2}{|e|^2}\mathsf{Var}(h_e) + \frac{2}{3.2^{M+2}}\sum_{j\neq i}\mathcal{A}_{ij}}\right)\\
&\leq \exp\left(-\Omega\left(\sum_{j\neq i}\mathcal{A}_{ij}\right)\right) \leq n^{-\Omega(C)}.
\end{aligned}$$

The second inequality holds since for any e, $\frac{(|e|-1)^2}{|e|^2}\mathsf{Var}(h_e) \leq \frac{(|e|-1)}{|e|}\mathsf{E}h_e$, and the last inequality is true under the condition of Theorem 7.1 since

$$\begin{aligned}
\sum_{j\neq i}\mathcal{A}_{ij} &= (2n-1)\beta_1 + (n-1)\beta_2\\
&= \sum_{m=2}^{M}\frac{\alpha_m(m-1)}{2n}\left[\binom{2n}{m} - 2\binom{n}{m}\right] = \Omega(C\ln n).
\end{aligned}$$

Denoting $[n-i] = \{1,\ldots,n\}\backslash i$, *i.e*, the first color class excluding vertex i, we have $\sum_{j\in[n-i]} A_{ij} = \sum_{e\in\mathcal{E}_0: e\ni i} h_e\frac{|e\cap[n-i]|}{|e|}$, and one can bound

$$\begin{aligned}
&\mathsf{P}\left(\sum_{j\in[n-i]} A_{ij} \geq \left(1+\frac{1}{2^{M+2}}\right)\sum_{j\in[n-i]}\mathcal{A}_{ij}\right)\\
&\leq \exp\left(-\frac{\frac{1}{2^{2M+4}}\left(\sum_{j\in[n-i]}\mathcal{A}_{ij}\right)^2}{2\sum_{e\in\mathcal{E}_0: e\ni i}\mathsf{Var}(h_e)\frac{|e\cap U|^2}{|e|^2} + \frac{2}{3.2^{M+2}}\sum_{j\in[n-i]}\mathcal{A}_{ij}}\right)\\
&\leq n^{-\Omega(C)}.
\end{aligned}$$

Thus, with probability $(1 - n^{-\Omega(C)})$, we have

$$\sum_{j\in[n-i]} A_{ij} < \left(1 + \frac{1}{2^{M+2}}\right) \sum_{j\in[n-i]} \mathcal{A}_{ij}$$
$$= \sum_{m=2}^{M} \frac{\alpha_m(n-1)}{m} \left(1 + \frac{1}{2^{M+2}}\right) \left(\binom{2n-2}{m-2} - \binom{n-2}{m-2}\right),$$

and

$$\sum_{j\neq i} A_{ij} > \left(1 - \frac{1}{2^{M+2}}\right) \sum_{j\neq i} \mathcal{A}_{ij}$$
$$= \sum_{m=2}^{M} \frac{\alpha_m}{m} \left(1 - \frac{1}{2^{M+2}}\right) \left((2n-1)\binom{2n-2}{m-2} - (n-1)\binom{n-2}{m-2}\right).$$

Using above relation, we can derive (7.9) since

$$\sum_{j\in\mathcal{V}_1^{(t-1)}\setminus\{i\}} A_{ij} = \sum_{j\in\mathcal{V}_{err}^{(t-1)}\cap\mathcal{V}_1^{(t-1)}\setminus\{i\}} A_{ij} + \sum_{j\in\mathcal{V}_1^{(t-1)}\setminus(\mathcal{V}_{err}^{(t-1)}\cap\{i\})} A_{ij}$$
$$\leq \sum_{j\in\mathcal{V}_{err}^{(t-1)}\setminus\{i\}} A_{ij} + \sum_{j\in[n-i]} A_{ij}$$
$$< \eta + \left(1 + \frac{1}{2^{M+2}}\right) \sum_{j\in[n-i]} \mathcal{A}_{ij}$$

The first inequality uses the fact $\mathcal{V}_1^{(t-1)}\setminus\mathcal{V}_{err}^{(t-1)}$ is the set of correctly colored nodes, with true color same as i. Hence, $\mathcal{V}_1^{(t-1)}\setminus(\mathcal{V}_{err}^{(t-1)}\cap\{i\}) \subset [n-i]$. From definition of η, we have

$$\sum_{j\in\mathcal{V}_1^{(t-1)}\setminus\{i\}} A_{ij}$$
$$\leq \sum_{m=2}^{M} \frac{\alpha_m(n-1)}{m} \left[\frac{1}{2^{M+2}}\binom{n-2}{m-2} + \left(1 + \frac{1}{2^{M+2}}\right)\left(\binom{2n-2}{m-2} - \binom{n-2}{m-2}\right)\right]$$
$$= \sum_{m=2}^{M} \frac{\alpha_m(n-1)}{m} \left(1 - \frac{1}{2^{M+2}}\right) \left[\binom{2n-2}{m-2} - \frac{1}{2}\binom{n-2}{m-2}\right]$$
$$+ \sum_{m=2}^{M} \frac{\alpha_m(n-1)}{2m} \left[\frac{1}{2^M}\binom{2n-2}{m-2} - \binom{n-2}{m-2}\right] - \sum_{m=2}^{M} \frac{\alpha_m(n-1)}{m2^{M+3}}\binom{n-2}{m-2}.$$

One can see that the first term is at most $\frac{1}{2}\left(1-\frac{1}{2^{M+2}}\right)\sum_{j\neq i}\mathcal{A}_{ij} < \frac{1}{2}\sum_{j\neq i}A_{ij}$. On the other hand, we note that

$$\frac{\binom{2n-2}{m-2}}{\binom{n-2}{m-2}} \leq \frac{1}{4}\frac{\binom{2n}{m}}{\binom{n}{m}} \leq \frac{1}{4}\frac{\frac{(2n)^m}{m!}}{\frac{n^m}{4.m!}} = 2^m \leq 2^M \ .$$

So the second term is negative, which proves (7.9), and the claim follows.

Proof of Lemma 7.4

Let $S_1, S_2 \subset \mathcal{V}$ be arbitrary such that $|S_2| = b$, and $\mathcal{E}_{S_1S_2}$ be the set of all non-monochromatic subsets of $\mathcal{V}$ of size at most M that have non-empty intersection with both S_1 and S_2. Then

$$\begin{aligned}\sum_{e\in\mathcal{E}_{S_1S_2}} h_e &\geq \frac{1}{M}\sum_{e\in\mathcal{E}_{S_1S_2}} h_e\frac{|e\cap S_1||e\cap S_2|}{|e|}\\ &\geq \frac{1}{M}\sum_{i\in S_2}\sum_{j\in S_1\setminus\{i\}} A_{ij} \geq \frac{b\eta}{M},\end{aligned}$$

where the last inequality holds under the condition stated in the lemma. Now we bound the probability

$$\begin{aligned}&\mathsf{P}\left(\exists S_1,S_2\subset\mathcal{V}, |S_2| = \frac{1}{2}|S_1| \leq \frac{n}{M^2 2^{2M+5}}, \sum_{j\in S_1\setminus\{i\}} A_{ij} \geq \eta\ \forall i\in S_2\right) \qquad (7.10)\\ &\leq \sum_{b=1}^{\frac{n}{M^2 2^{2M+5}}} \mathsf{P}\left(\exists S_1,S_2\subset\mathcal{V}, |S_2| = \frac{1}{2}|S_1| = b, \text{ and } \sum_{e\in\mathcal{E}_{S_1S_2}} h_e \geq \frac{b\eta}{M}\right)\\ &\leq \sum_{b=1}^{\frac{n}{M^2 2^{2M+5}}} \sum_{S_2:|S_2|=b}\sum_{S_1:|S_1|=2b} \mathsf{P}\left(\sum_{e\in\mathcal{E}_{S_1S_2}} h_e \geq \frac{b\eta}{M}\right)\end{aligned}$$

We observe that

$$\begin{aligned}\sum_{e\in\mathcal{E}_{S_1S_2}}\mathsf{E}[h_e] &= \sum_{m=2}^{M}\sum_{e\in\mathcal{E}_{S_1S_2},|e|=m}\alpha_m \\ &\le 2b^2\sum_{m=2}^{M}\alpha_m\binom{2n-2}{m-2} \\ &\le b^2 2^{M+1}\sum_{m=2}^{M}\alpha_m\binom{n-2}{m-2} \le \frac{b^2\eta M 2^{2M+4}}{n},\end{aligned}$$

and the above bound is smaller than $\frac{b\eta}{2M}$ for $b \le \frac{n}{M^2 2^{2M+5}}$. Hence, we can write

$$\begin{aligned}&\mathsf{P}\left(\sum_{e\in\mathcal{E}_{S_1S_2}} h_e \ge \frac{b\eta}{M}\right) \\ &\le \exp\left(\frac{-\left(\frac{b\eta}{M}-\sum_{e\in\mathcal{E}_{S_1S_2}}\mathsf{E}[h_e]\right)^2}{2\sum_{e\in\mathcal{E}_{S_1S_2}}\mathsf{Var}(h_e)+\frac{2}{3}\left(\frac{b\eta}{M}-\sum_{e\in\mathcal{E}_{S_1S_2}}\mathsf{E}[h_e]\right)}\right) \\ &\le \exp\left(-\frac{3b\eta}{16M}\right).\end{aligned}$$

Substituting in (7.10), we have the probability of the existence of S_1, S_2 with mentioned conditions is at most

$$\sum_{b=1}^{\frac{n}{M^2 2^{2M+5}}}\binom{2n}{b}\binom{2n}{2b}\exp\left(-\frac{3b\eta}{16M}\right) \le \sum_{b=1}^{\infty}\left(2n\exp\left(1-\frac{\eta}{16M}\right)\right)^{3b}.$$

Under the assumption of Theorem 7.1, one can verify that $\eta \ge \frac{C\ln n}{2^{2M+4}}$. So for large C, the above geometric series converges, and is at most $n^{-\Omega(C)} = o(1)$.

It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.

Sir Arthur Conan Doyle, *Scandal in Bohemia*

Chapter 8

Numerical Studies

This chapter consists of numerical evidence that validate our analysis, and also demonstrate the performance of the studied algorithms in practical problems[1].

Before presenting the detailed numerical studies, we provide some illustrations in Figure 8.1 that demonstrate the use of spectral hypergraph partitioning in subspace clustering, motion segmentation, geometric grouping and hypergraph matching.

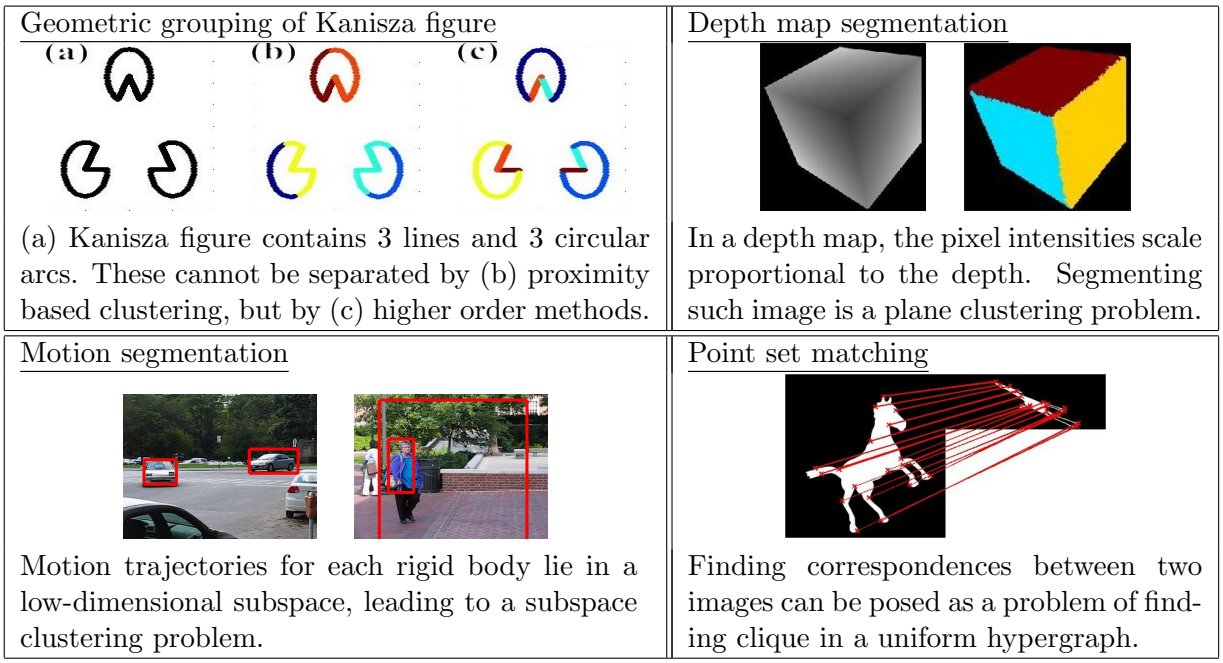

Figure 8.1: Examples illustrating the applications of hypergraph partitioning. These results have been obtained using the Algorithm HOSVD.

[1] The implementations of the various studies in this chapter are available at: http://sml.iisc.ernet.in/publications.html

The numerical results in this chapter are organized in six sections. In Section 8.1, we study the nature of hypergraphs that arise in practice, and comment on the applicability of our assumptions and consistency results. Subsequently, we empirically validate our theoretical findings about the spectral methods, HOSVD, TTM and NH-Cut, in Section 8.2. We then compare these methods with popular hypergraph partitioning algorithms in Section 8.3. Specific problems of categorical data clustering and subspace clustering are considered in Sections 8.4 and 8.5, respectively, where we also compare the above spectral methods with the state-of-the-art algorithms for each problem. In addition, Section 8.5 also contains numerical studies on large problems such as motion segmentation, where efficient partitioning algorithms such as Tetris are essential. Finally, we expand the use of the presented spectral algorithms to any clustering application by constructing similarity hypergraphs based on general multi-point similarity measures. As an example, in Section 8.6, we present a class of similarities arising from the notion of multi-distribution information theoretic divergences, and study the performance of HOSVD when a weighted hypergraph is constructed using this similarity. We compare the performance of this approach to data clustering with standard methods used in the literature.

8.1 Nature of real-world hypergraphs

Our theoretical studies were based on a statistical model for hypergraphs with a planted solution. We now focus our attention on hypergraphs or networks that have been studied in practical problems (Ghoshal et al., 2009; Alpert, 1998). Recall that the consistency results in this thesis are applicable only under certain restrictions on the hypergraph to be partitioned. To be precise, we have established that spectral methods are consistent when the sparsity parameter (α_m) of the hypergraph is above a certain threshold. We study the practicability of such conditions in the case of real-world hypergraphs. We consider two types of applications – *folksonomy*, where the underlying model is a 3-uniform hypergraph, and *circuit design*, which involves non-uniform hypergraph partitioning.

To study the nature of hypergraphs in folksonomy, we consider 11 networks from KONECT, HetRec'2011 and MovieLens[1]. The networks under consideration contain folksonomy data related to the sites – Bibsonomy, CiteULike, MovieLens, vi.sualize.us, Last.fm and Delicious. Six different versions of the MovieLens dataset are available. Each network is a tri-partite 3-uniform hypergraph containing three types of vertices – user, resource and annotation. Each

[1] The HetRec'2011 and MovieLens datasets are maintained by the GroupLens research group, and are available at: http://grouplens.org/
KONECT refers to the Koblenz network collection: http://konect.uni-koblenz.de/

edge is an entry in the database that occurs when an user describes a certain resource by a particular tag or rating. The number of vertices vary between 2630 to 9.8×10^5. Assuming that $k = O(1)$, the sufficient condition in Corollary 5.12 requires that the number of edges in a 3-uniform hypergraph grows as $\Omega\left(n(\ln n)^2\right)$. In Figure 8.2, we compare the number of edges $|\mathcal{E}|$ with $n(\ln n)^2$ for above networks. We observe that in few cases (last four in Figure 8.2), these quantities are similar, whereas for the remaining networks, $|\mathcal{E}|$ is smaller by a nearly constant factor.

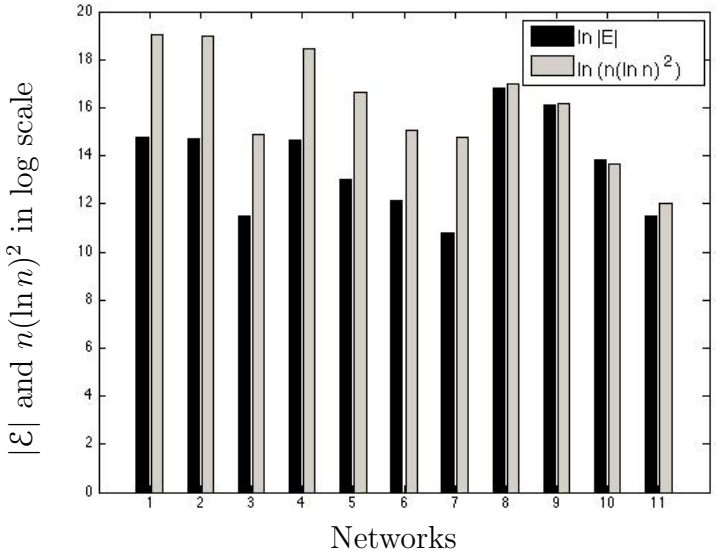

Figure 8.2: Bar plot for $|\mathcal{E}|$ and $n(\ln n)^2$ in logarithmic scale for 11 folksonomy networks.

The next study is related to non-uniform hypergraphs that are encountered in circuit partitioning. We consider 18 circuits from the ISPD98 circuit benchmark suite (Alpert, 1998). From a hypergraph view, the components of the circuit are the vertices of the hypergraph, while the multi-way connections among them are the edges. These networks are also sparse as the number of vertices vary from 1.27×10^4 to 2.1×10^5, while the number of edges range between 1.4×10^4 to 2×10^5. Moreover, these networks contain relatively large number of edges of sizes 2 or 3, and the number of edges of size m gradually decreases with m. We assume $a = 1$, $b = 2$, and ignoring constant factors, we estimate θ_m as $\theta_m = \frac{|\mathcal{E}_m|}{n(\ln n)^2}$, where $\mathcal{E}_m$ is the set of edges of size m in the network. Figure 8.3 shows a plot of this quantity as a function of m for different networks. We find that the estimate of θ_m is bounded by exponentially decaying functions, and hence, one can argue that $\sum_m m\theta_m < \infty$.

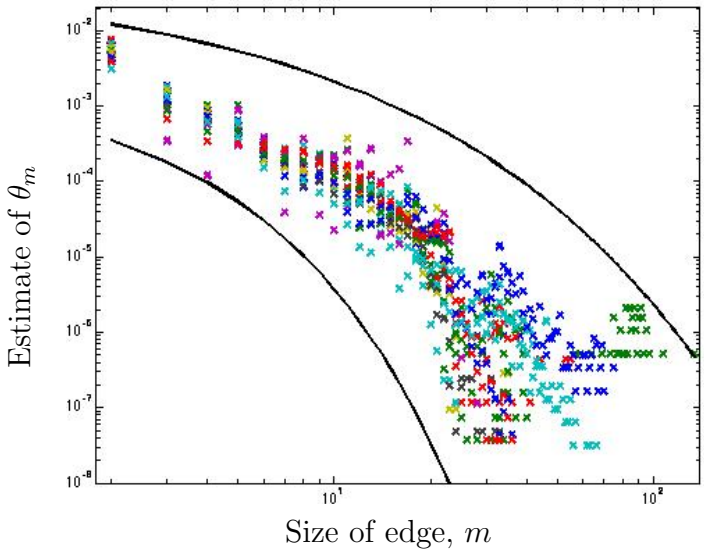

Figure 8.3: Scatter plot for estimated $\theta_m = \frac{|\mathcal{E}_m|}{n(\ln n)^2}$ versus m for the 18 circuits. Plot for each circuit is shown in a different color. The bounding curves correspond to the functions $0.05\exp(-m^{0.5})$ from above and $0.002\exp(-m^{0.8})$ from below.

8.2 Validation of the consistency results

We support the consistency results stated in the previous chapters through numerical simulations. For this, we demonstrate the performance of TTM, HOSVD and NH-Cut when random uniform hypergraphs are generated from a planted model.

Consider the following setting for a m-uniform hypergraph on n vertices. We assume here that $\alpha_m = 1$, $k = 2$, and the true clusters are of equal size. The edges occur with following probabilities. If all vertices in an edge do not belong to the same cluster, then the edge probability is $q = 0.2$, else it is $(p + q)$ for some $p \in (0, 1 - q)$.

Figure 8.4 shows results for three examples, where p is fixed at $p = 0.1$, m is varied over $m = 2, 3, 4$, and the total number of vertices n grows. For each case, 50 planted hypergraphs are generated, and subsequently partitioned by TTH, HOSVD and NH-Cut. The mean error, $\text{Error}(\psi, \psi')$, is reported for each algorithm as a function of n. Figure 8.4 shows that the performance of TTM and NH-Cut are similar, and the errors incurred by these methods are significantly smaller than that of HOSVD. This observation validates our observations made in Corollaries 4.6 and 5.12, and subsequent discussions. It can also be seen that all three methods incur a sub-linear error rate for $m = 2$, *i.e*, they are weakly consistent, whereas, the error reduces, $\text{Error}(\psi, \psi') = o(1)$, for $m \geq 3$.

We consider another example on bi-partitioning 3-uniform hypergraphs, where we fix $q = 0.2$

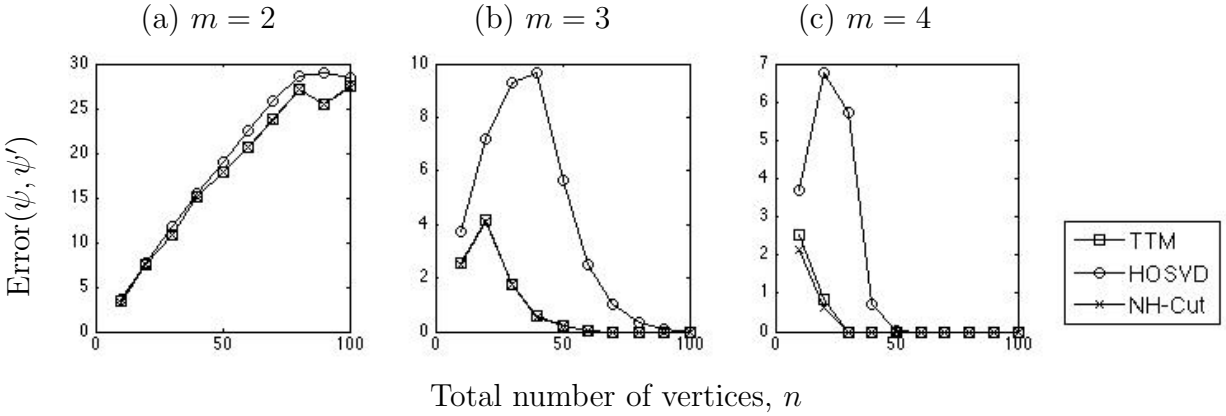

Figure 8.4: Number of vertices mis-clustered by TTM, HOSVD and NH-Cut as n increases. The figures from left to right correspond to cases with $m = 2$, 3 and 4, respectively.

but the density gap p is decreased as 0.1, 0.05 and 0.025. Figure 8.5 shows the errors, averaged over 50 runs, incurred by the three methods as the hypergraph grows. Note that the problem becomes harder as p reduces, and the performance of HOSVD is highly affected. But, the effect is much less in case of TTM and NH-Cut. This follows from the consistency theorems, where one can observe that, in the present context the clustering error varies as $1/p^2$. Same holds for NH-Cut, but in the case of HOSVD, the error varies as $1/p^4$ making the algorithm more sensitive to reduction in probability gap.

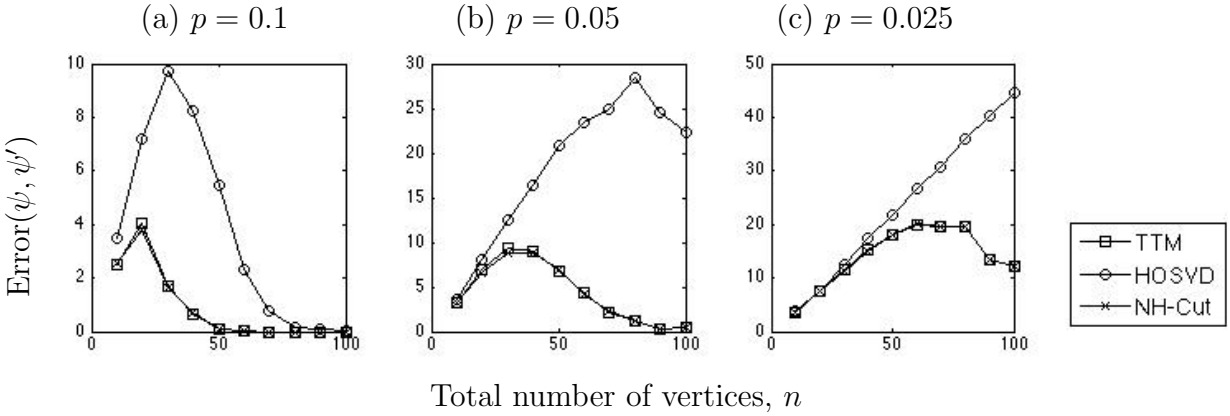

Figure 8.5: Number of vertices mis-clustered by TTM, HOSVD and NH-Cut as n increases. The figures from left to right correspond to cases with $p = 0.1$, 0.05 and 0.025, respectively.

8.3 Finding communities in planted hypergraphs

While the above simulations validate the conclusions of our theoretical analysis of the spectral algorithms, a similar study can be conducted to empirically compare the performance of spectral methods with other hypergraph partitioning algorithms. In particular, we compare TTM, HOSVD and NH-Cut with the following methods:

- hypergraph partitioning by symmetric non-negative tensor factorization (SNTF) of the adjacency tensor (Shashua et al., 2006),
- higher order game theoretic clustering (HGT) (Rota Bulo and Pelillo, 2013), which formulates the partitioning problem as an evolutionary game, and
- the hMETIS tool (Karypis and Kumar, 2000), a multi-level approach that is widely used in VLSI community.

We compare the different algorithms under a planted model for 3-uniform hypergraphs with $k = 3$ planted clusters of equal size. As before, we assume the hypergraph to be dense, $\alpha_3 = 1$, and the inter-cluster edges occur with probability $q = 0.2$. We study the performance of the methods as the the number of vertices n, and the probability gap p varies. The fractional clustering error, $\frac{1}{n}\text{Error}(\psi, \psi')$, averaged over 50 runs, is reported in Figure 8.6. The color bar (on the right) indicates the shade corresponding to different levels of error, with darker shade representing larger error. The figure shows the previously observed trend about relatively performance of TTM, NH-Cut and HOSVD. In addition, it is observed that SNTF and hMETIS provide nearly similar, but marginally worse results than TTM. However, HGT uses a greedy strategy for extracting individual clusters, and hence, often identifies a majority of the vertices as outliers, thereby resulting in poor performance.

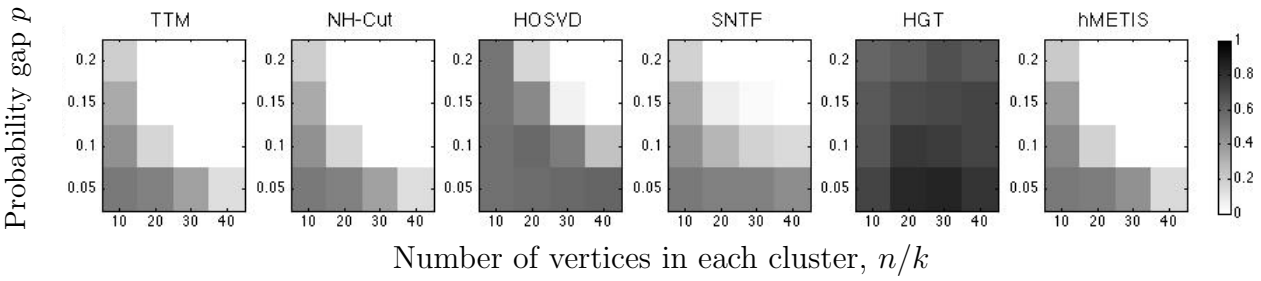

Figure 8.6: Fractional error incurred by hypergraph partitioning algorithms under a planted model. The cluster size, (n/k), and the probability gap p are varied.

8.4 Categorical data clustering with non-uniform hypergraphs

We now shift our attention to the study of spectral methods in practical applications. Partitioning the networks discussed in Section 8.1 is an interesting problem. However, for such networks, the underlying partition is not known, and hence, for these networks, the accuracy of a solution cannot be measured in terms of the number of incorrectly assigned vertices.

Our first practical study is based on benchmark categorical data clustering problems, where the true partition is known. Here, one needs to group instances of a database, each described by a number of categorical attributes. Two such benchmark databases include the *1984 US Congressional Voting Records* and the *Mushroom Database* available at the UCI repository (Lichman, 2013). The first set contains votes of 435 Congress men on 16 issues. The task is to group the Congress men into Democrats and Republicans based on whether they voted for or against each of the issues, or abstained their votes. The mushroom database contains information about 22 features of 8124 varieties of mushrooms. Based on the categorical features, one needs to separate the edible varieties from the poisonous ones. Thus, both databases have two well-defined classes.

A standard approach (Gibson et al., 2000) is to consider a m-uniform m-partite hypergraph, where m is the number of attributes for each data instance. The vertices are the possible values of all attributes, and each instance is an edge of size m. This representation is similar to the folksonomy networks. One then partitions this hypergraph to group the possible attribute values. An instance is then labeled as group-i if a majority of its attributes belong to group-i (Han et al., 1997). In mushroom database, there are some missing entries. We consider such instances as an edge of size $< m$. Also, in the final stage, we break ties randomly.

Alternatively, one may directly consider the instances of the database as the vertices of the hypergraph. For each possible value of each attribute, an edge is considered among all instances that take the particular value of the attribute. This generates a sparse non-uniform hypergraph that can be partitioned to obtain the clusters.

Table 8.1 compares the performance of the non-uniform hypergraph methods, NH-Cut and TTM-ext, with some popular categorical clustering algorithms.

- ROCK (Guha et al., 2000),
- COOLCAT (Barbara et al., 2002),
- LIMBO (Andritsos et al., 2004), and

- hMETIS (Han et al., 1997; Karypis and Kumar, 2000).

For the spectral algorithms, we consider both the aforementioned approaches, which we denote as "Direct" and "Indirect" to specify that the instances are directly partitioned (second method) or instances are grouped using the partition of attribute values. The error is measured as $\frac{1}{n}\text{Error}(\psi, \psi')$. The results for ROCK, COOLCAT and LIMBO are taken from (Andritsos et al., 2004). Table 8.1 shows that both TTM-ext and NH-Cut perform quite well compared to other categorical data clustering methods. Spectral partitioning of clique expansion (Rodríguez, 2002) also performs similar to TTM-ext (see Ghoshdastidar and Dukkipati, 2016).

Table 8.1: Fraction of data mis-clustered by different algorithms in categorical data clustering.

Database	ROCK	COOL-CAT	LIMBO	hMETIS	TTM-ext		NH-Cut	
					Direct	Indirect	Direct	Indirect
Voting	0.16	0.15	0.13	0.24	0.12	0.12	0.12	0.12
Mushroom	0.43	0.27	0.11	0.48	0.11	0.35	0.11	0.35

8.5 Subspace clustering with uniform hypergraphs

Our next application is in the problem of subspace clustering described in Section 2.4.3. We compare the performance of the various hypergraph partitioning methods in the case of subspace clustering. In particular, we consider the line clustering problem in an ambient space of dimension 3. We randomly generate three one-dimensional subspaces, and sampled n/k random points from each subspace. As mentioned earlier, subspace clustering problems typically involve noisy perturbations of the points. To simulate this behavior, we add a mean zero Gaussian noise vector to each point. The covariance of the noise vectors is given as $\sigma_a I$, where we vary σ_a to control the difficulty of the problem. We construct a weighted 3-uniform similarity hypergraph based on polar curvature of triplet of points, which is partitioned by the different methods. The fractional clustering errors are illustrated in Figure 8.7. As is expected, all the methods can identify the exact subspace in the absence of noise, and the errors increase for larger σ_a. Apart from HGT, good performance is observed from all the methods, and the spectral methods are quite robust to the presence of noise.

One can observe that the above comparisons were based on very small problems, where the hypergraph consists of at most 120 vertices. This restriction was imposed since specification of the entire weighted adjacency tensor is computationally infeasible for large hypergraphs. We now compare spectral partitioning methods against the state of the art subspace clustering

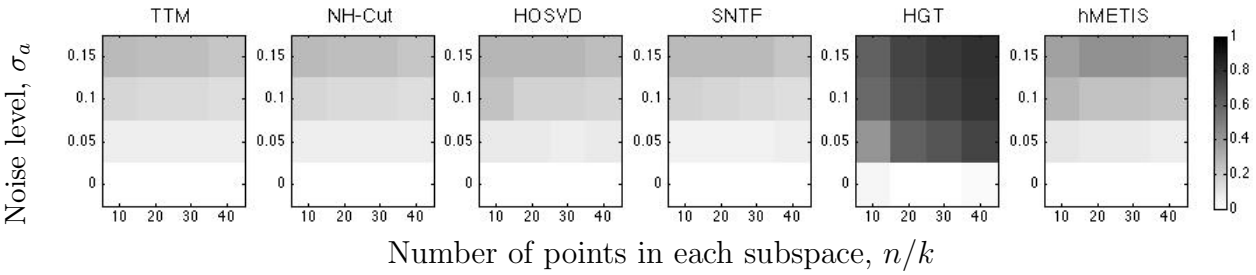

Figure 8.7: Fractional error incurred by hypergraph partitioning algorithms in clustering noisy points from three intersecting lines. The cluster size, (n/k), and the noise level σ_a are varied.

algorithms. Considering the computational complexity of large clustering problems, we use sampled variants of the TTM algorithm, *i.e*, TTM with uniform sampling and TTM with iterative sampling (Tetris). The subspace clustering algorithms under consideration include:

- k-means algorithm based on Euclidean distance,
- k-flats (Bradley and Mangasarian, 2000) which generalizes k-means to subspace clustering,
- sparse subspace clustering (SSC) (Elhamifar and Vidal, 2013), which finds clusters by estimating the subspaces,
- subspace clustering using low-rank representation (LRR) (Liu et al., 2010),
- thresholding based subspace clustering (TSC) (Heckel and Bölcskei, 2013),
- faster variant of SSC using orthogonal matching pursuit (SSC-OMP) (Dyer et al., 2013),
- greedy subspace clustering using nearest subspace neighbor search and spectral clustering (NSN+Spectral) (Park et al., 2014),
- spectral curvature clustering (SCC) (Chen and Lerman, 2009), which is an iterative variant of HOSVD,
- sparse Grassmann clustering (SGC) (Jain and Govindu, 2013)[1], another iterative modification of HOSVD where the eigenvectors are updated at each iteration,
- Algorithm Tetris, and
- Algorithm TTM with uniform sampling, which is derived by performing a single iteration of Steps 1–14 of Algorithm Tetris[2].

[1] For this method, we have used our own implementation.

[2] Here, the edge sampling is not exactly uniform since we only select the c subsets of size $(m-1)$ uniformly.

8.5.1 Comparison of subspace clustering algorithms on synthetic data

We first focus on the problem of clustering randomly generated subspaces[1]. In an ambient space of dimension $r_a = 5$, we randomly generate $k = 5$ subspaces each of dimension $r = 3$. From each subspace, we randomly sample n/k points and perturb every point with a 5-dimensional Gaussian noise vector with mean zero and covariance $\sigma_a I$. In Figure 8.8, we report the fractional error, $\frac{1}{n}\text{Error}(\psi, \psi')$, incurred by various subspace clustering algorithms when (n/k) and σ_a are varied. The results are averaged over 50 independent trials. We note that for existing methods, we fix the parameters as mentioned in (Park et al., 2014). For Tetris, the parameters are set to the same values as SCC, where $c = 100k$ and σ as in (2.21) is determined by the algorithm. In case of uniformly sampled TTM, we fix σ to be same as the value determined by Tetris. To demonstrate that sampling more edges lead to error reduction, we consider uniform sampling for two values $c = 100k$ and $200k$.

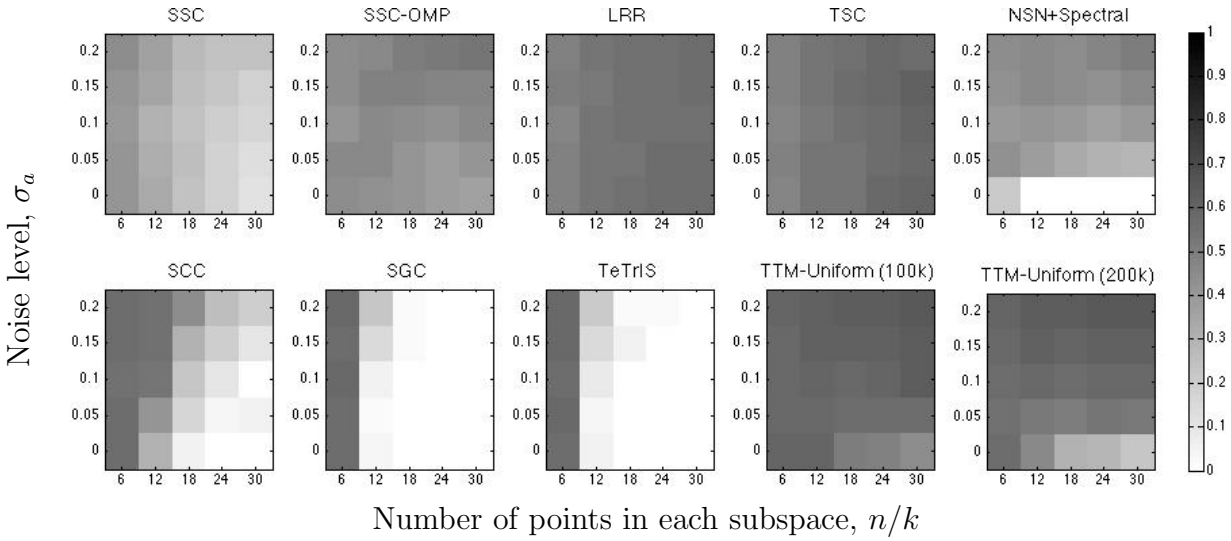

Figure 8.8: Fractional error incurred by subspace clustering algorithms for synthetic data. The number of points in each subspace, (n/k), and the variance of the noise vector σ_a is varied.

Figure 8.8 shows that Tetris and SGC clearly outperform other methods over a wide range of settings. In particular, it can be seen that greedy methods like NSN is accurate in the absence of noise, but a drastic increase in error occurs when the data is noisy. The effect of noise is much less in hypergraph based methods like SCC, SGC or Tetris. One can also observe that these methods do not work well when there are very few points in each cluster (for example,

[1]The experimental setup has been adapted from (Park et al., 2014), and the codes are available at: http://sml.csa.iisc.ernet.in/SML/code/Feb16_TensorTraceMax.zip

6). This is expected since, by definition, these algorithms construct 5-uniform hypergraphs ($m = r + 2$) in this case, and hence, there are very few edges ($\binom{6}{5} = 6$) with large weight for each cluster. However, with increase in number of points, there is a rapid increase in accuracy of the algorithm. This also shows the consistency of these methods empirically. To this end, it seems that NSN or SSC should be recommended for small scale problems (smaller n/k), whereas Tetris or SGC should be the algorithm of choice for larger n and possible presence of noise. Finally, we also observe that TTM with uniform sampling, even with twice the number of samples, performs quite poorly as compared to Tetris or SGC. However, with increase in the number of sampled edges, some extent of error reduction is observed.

8.5.2 Comparison of subspace clustering algorithms on motion segmentation benchmark

The Hopkins 155 database (Tron and Vidal, 2007) contains a number of videos capturing motion of multiple objects or rigid bodies. In each video, few features are tracked along the frames, each giving rise to a motion trajectory that resides in a space of dimension twice the number of frames. One can show that under particular camera models, all trajectories corresponding to a particular rigid body motion span a subspace of dimension at most four (Tomasi and Kanade, 1992). Thus, the problem of segmenting different motions in a video can be posed as a subspace clustering problem.

The Hopkins database contains 120 sequences, each containing two motions, and 35 three motion sequences. We run above mentioned subspace clustering algorithms for purpose of motion segmentation. For existing approaches, the parameters specified in (Park et al., 2014) have been used, and for Tetris and SGC, we fix $l = 3$, which is the value commonly used for SCC. TTM with uniform sampling is not considered due to its higher error rate. Table 8.2 reports the mean and median of the percentage errors incurred by different algorithms, where these statistics are computed over all 2-motion and 3-motion sequences. In order to remove the effect of randomization due to sampling (for SCC, SGC, Tetris) or initialization (for k-means, k-flats, NSN), we average the results over 20 independent trials. The mean computational time (in seconds) of each algorithm is also reported[1].

Table 8.2 shows that Tetris performs quite well in comparison with state of the art subspace clustering algorithms. In particular, Tetris achieves least mean error for two cluster problem. The computational time for Tetris is also much smaller than other accurate methods like SSC

[1] The reported times are based on the fact that we have used Matlab implementations of the algorithms, run on a Mac OS X operating system with 2.2 GHz Intel Core i7 processor and 16 GB memory.

and LRR. The mean error achieved by Tetris is also smaller than SCC in either cases. We note here that the best known results for Hopkins 155 database is achieved by the algorithm in (Jung et al., 2014), which uses techniques based on epipolar geometry, and hence, it is not a subspace clustering algorithm. Smaller errors have also been reported in the literature when one construct larger tensors, $m = 8$ (Jain and Govindu, 2013), or uses manual tuning of parameters (Ghoshdastidar and Dukkipati, 2015b). However, in either cases, computational times increases considerably.

Table 8.2: Mean and median of clustering error and computational time for different subspace clustering algorithms on Hopkins 155 database.

Algorithm	2 motion (120 sequences)			3 motion (35 sequences)		
	Mean (%)	Median (%)	Time (s)	Mean (%)	Median (%)	Time (s)
k-means	19.58	17.92	0.03	26.13	20.48	0.05
k-flats	13.19	10.01	0.38	15.45	14.88	0.76
SSC	1.53	0.00	0.80	4.40	0.56	1.51
LRR	2.13	0.00	0.94	4.03	1.43	1.29
SSC-OMP	16.93	13.28	0.72	27.61	23.79	1.23
TSC	18.44	16.92	0.19	28.58	29.67	0.51
NSN+Spec	3.62	0.00	0.08	8.28	2.76	0.17
SCC	2.53	0.03	0.45	6.40	1.46	0.76
SGC	3.50	0.41	0.54	9.08	5.05	0.89
Tetris	1.31	0.02	0.50	5.71	1.19	0.90

8.6 Data clustering with similarity hypergraphs

The hypergraph partitioning approach to the subspace clustering problem shows that one can possibly represent any multi-point similarity in terms of a tensor. Subsequently, data analysis using multi-point similarities can be formulated as a hypergraph problem. This observation lies at the heart of higher-order learning. However, till date, the use of such approaches have been limited to problems such as subspace clustering, geometric grouping or point set matching. The previous section was dedicated to subspace clustering and its application in motion segmentation, while other illustrative examples can be found in Figure 8.1.

Here, we discuss an extension of higher order learning by using general similarity measure defined over multiple data instances. A standard approach to quantify similarity is by means of the Euclidean distance. However, it often turns out that standard Euclidean distance metrics do not suffice in a wide range of situations. For instance, when one restricts the space to that of probability distributions, information theoretic divergences often prove to be useful

distance measures in the context of learning (Garcia-Garcia and Williamson, 2012; Nielsen and Nock, 2013). To this end, Csiszár's f-divergences (Csiszár, 1967) and the Jensen-type divergences (Sibson, 1969; Lin, 1991), are quite special. This is primarily because these divergences are multi-distribution divergences, and hence, provide a measure of dissimilarity among more than two probability distributions.

8.6.1 Jensen-Tsallis kernels and multi-point extensions

Similarities or kernel functions based on Jensen divergences have been often studied in the literature. This is primarily motivated by the fact that the square-root of the Jensen-Shannon (JS) divergence is a Hilbertian metric (Endres and Schindelin, 2003). Subsequently, the works in (Cuturi et al., 2005) proposed new kernels on probability measures based on the JS-divergence. Martins et al. (2009) further extend the idea to nonextensive extensions of the JS divergence, that arise from applications in statistical physics (Tsallis, 1988). The so-called Jensen-Tsallis (JT) kernel proposed by Martins et al. (2009) is defined as

$$k_q(x,y) = \begin{cases} \dfrac{1}{(q-1)} \displaystyle\sum_{j=1}^{d} \left(\left(x^{(j)} + y^{(j)}\right)^q - \left(x^{(j)}\right)^q - \left(y^{(j)}\right)^q \right) & \text{for } q \neq 1, \\ \displaystyle\sum_{j=1}^{d} \left(\left(x^{(j)} + y^{(j)}\right) \ln \left(x^{(j)} + y^{(j)}\right) - x^{(j)} \ln \left(x^{(j)}\right) - y^{(j)} \ln \left(y^{(j)}\right) \right) & \text{for } q = 1, \end{cases} \tag{8.1}$$

where x, y are d-dimensional probability vectors and $x^{(j)}$ denotes the j^{th} coordinate of x. The quantity q has interesting interpretations in the physics literature, but in the present context, it acts as a parameter for the JT kernel, and the kernel is known to be positive definite for all $q \in [0, 2]$ with the case $q = 1$ corresponding to the JS kernel (Cuturi et al., 2005).

A related similarity measure is the exponential JT (expJT) kernel, defined as

$$k_q^{(e)}(x,y) = \exp\left(t k_q(x,y)\right), \tag{8.2}$$

which is governed by parameters $q \in [0, 2]$ and $t > 0$. The significance of the JT and expJT kernels have been empirically established in the literature (Martins et al., 2009; Bicego et al., 2010), and it is also known that theoretical properties of these kernels do not change if one defines them on an Euclidean space, more precisely, the unit cube in $\mathbb{R}^d$ (see Ghoshdastidar et al., 2014, 2016).

Owing to the possibility of extending Jensen type divergence over multiple probability distributions (Sibson, 1969), one can also define multi-point extensions of the JT-kernel (8.1) and the expJT kernel (8.2). For any integer $m \geq 2$, we define the m-point JT kernel (JTm) as

$$k_{q,m}\left(x_1,\ldots,x_m\right) = \begin{cases} \dfrac{1}{(q-1)}\displaystyle\sum_{j=1}^{d}\left[\left(\sum_{i=1}^{m} x_i^{(j)}\right)^q - \sum_{i=1}^{m}\left(x_i^{(j)}\right)^q\right] & \text{for } q \neq 1 \\ \displaystyle\sum_{j=1}^{d}\left[\left(\sum_{i=1}^{m} x_i^{(j)}\right)\ln\left(\sum_{i=1}^{m} x_i^{(j)}\right) - \sum_{i=1}^{m} x_i^{(j)}\ln x_i^{(j)}\right] & \text{for } q = 1, \end{cases} \tag{8.3}$$

where the arguments $x_1,\ldots,x_m \in [0,1]^d$. Similarly, one can also define the m-point expJT kernel (expJTm) as

$$k_{q,m}^{(e)}(x_1,...,x_m) = \exp\left(tk_{q,m}(x_1,...,x_m)\right) \qquad \text{for } t > 0. \tag{8.4}$$

8.6.2 Similarity hypergraph and clustering

The similarity measures defined in (8.3) and (8.4) provides us the opportunity to formulate a problem of clustering n data instances into a m-uniform hypergraph partitioning problem, where each edge on m vertices has a weight given by the similarity functions in (8.3) or (8.4). In the experiments, we consider $m = 3$ and use HOSVD to partition the similarity hypergraph.

We compare this approach with spectral clustering (Ng et al., 2002) based on a wide variety of pairwise similarities or kernel functions, including the standard JT (8.1) and expJT (8.2) kernels as well as Gaussian and polynomial kernels. We compare the performance of the different similarity functions on UCI datasets (Lichman, 2013) and gene expressions (de Souto et al., 2008). We also compare the results with the performance of some other clustering algorithms such as standard k-means algorithm (KM), spectral clustering with k nearest neighbor based adjacency (SCNN), mean shift algorithm (MS), variants of maximum margin clustering (MMC), and minimal entropy encoding (MEE). The results for KM, SCNN, MS, MMC and MEE have been taken from (Melacci and Gori, 2012), and are reported first.

The UCI datasets considered in our experiments are listed in 8.3. These datasets have previously considered by Melacci and Gori (2012) for comparison of various clustering algorithm. Following the lines of their study, we measure the performance of the different similarity

functions in terms of adjusted Rand index (ARI) of the obtained clusters defined as

$$ARI = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00} + N_{01})(N_{01} + N_{11}) + (N_{00} + N_{10})(N_{10} + N_{11})}$$

where N_{11} denotes the number of pairs which are in the same clusters according to both true labels as well as obtained clusters, and N_{00} is the number of pairs which have different labels and are also in different clusters. On the other hand, N_{01} and N_{10} are the number of pairs for which there is disagreement in the true and obtained clusters, where the former denotes the case of clustering pairs with different labels into the same cluster.

Table 8.3: List of UCI data sets considered for comparison of Jensen type multi-point similarities with standard kernels.

Data set	# instances	# attributes	# classes
Balance	625	4	3
Breast	569	30	2
Diabetes	768	8	2
German	1000	24	2
Heart	270	13	2
Ionosphere	351	34	2
Iris	150	4	2
Wine	178	13	3

In the experiments[1], we tune the parameters of the proposed and existing kernels, and report the best result (maximum ARI) in each case. This way of presenting the results have been adapted from (Melacci and Gori, 2012). For the Jensen-type kernels q is tuned as $q = 0.01$ or in the range $[0.25, 2]$ in steps of 0.25. Both t (for expJT) and parameter for Gaussian is varied from 0.01 to 100 in multiplicative step with a factor of 10, and for Polynomial kernel, we vary the degrees as $1, 2, \ldots, 10$. To account for the randomness in k-means initialization, we average the results over 20 independent runs, as considered in (Melacci and Gori, 2012). Table 8.4 shows that Jensen-type kernels, particularly the exponential variety, perform quite well compared to other methods. Relative merits of the 2-point and 3-point kernels depend mostly on the data.

We also conduct experiments on clustering gene expressions. The cancer gene expression database (de Souto et al., 2008) contains data sets related to two types of gene expressions: cDNA type and Affymetrix data sets. There are 14 cDNA sets and 21 Affymetrix sets, and the number of classes in each set vary between 2 to 14. Further details are available in (de Souto et al., 2008).

[1]Matlab codes are available at: http://sml.csa.iisc.ernet.in/SML/code/TNNLS15_code.zip

Table 8.4: *ARI* obtained from different methods for clustering UCI datasets.

Method	Balance	Breast	Diabetes	German	Heart	Ionosphere	Iris	Wine
KM	0.14	0.73	0.07	0.03	0.29	0.18	0.64	0.36
MS	0.16	0.74	0.02	0.01	0.34	0.00	0.71	0.39
MMC	0.18	0.74	0.10	0.02	0.31	0.30	0.73	0.37
MEE	0.20	0.74	0.08	0.06	0.31	0.58	0.90	0.42
SCNN	0.09	0.80	0.00	0.03	0.33	0.17	0.79	0.38
Gaussian	0.26	0.68	0.10	0.03	0.13	0.17	0.74	0.87
Polynomial	0.26	0.57	0.09	0.03	0.21	0.06	0.58	0.80
JT2	0.43	0.57	0.05	0.03	0.30	0.14	0.65	0.81
expJT2	0.49	0.79	0.10	0.04	0.25	0.19	0.60	0.95
JT3	0.54	0.52	0.05	0.03	0.26	0.16	0.61	0.87
expJT3	0.47	0.68	0.10	0.05	0.27	0.17	0.62	0.93

The performance of a number of clustering algorithms and proximity measures have been compared in (de Souto et al., 2008). The study concluded that best performance is usually obtained from k-means or mixture models, and spectral clustering works well in certain cases. We restrict our comparisons only to spectral clustering, but with various proximity measures or kernels. Standard proximity measures include Pearson's correlation, cosine, Spearman correlation coefficient and Euclidean distance[1]. Following de Souto et al. (2008), the best of 30 independent runs is considered for each dataset. The *ARI* averaged over all datasets of each type is reported in Table 8.5. The results show that the expJT3 kernel surpasses other similarity measures by some margin, thereby establishing the importance of general multi-point similarities.

Table 8.5: *ARI* of spectral clustering and HOSVD for clustering gene expression datasets.

Method	cDNA	Affymetrix
Pearson	0.33	0.39
Cosine	0.32	0.42
Spearman	0.27	0.40
Euclidean	0.10	0.11
Gaussian	0.27	0.27
Polynomial	0.38	0.47
JT2	0.36	0.45
expJT2	0.44	0.54
JT3	0.40	0.47
expJT3	0.47	0.57

[1] For Euclidean distance measure, we report only the result when data is normalized to the unit cube (termed as Z_2 in (de Souto et al., 2008)). We also use this normalization for Jensen type kernels.

When I go from hence let this be my parting word,
that what I have seen is unsurpassable.
I have tasted of the hidden honey of this lotus that
expands on the ocean of light, and thus am I blessed
– let this be my parting word.

Rabindranath Tagore, *Gitanjali*

Chapter 9

Concluding Remarks

The primary focus of this thesis has been to provide a theoretical treatment of hypergraph partitioning algorithms that have been used in several applications over the past two decades. This thesis contributes towards the formalization of the hypergraph partitioning problem by extending the related graph terminology, and, for the first time, expands the stochastic block model literature to analyze the performance of hypergraph partitioning algorithms. In this concluding chapter, we review the tools and techniques that are used to derive the consistency results. We also point out several directions in which the studies in this thesis can be extended in future, and mention some of the open questions that naturally arise in the context of hypergraph partitioning.

We first spend few words on the rationale behind focusing on spectral approaches. Apart from the historical reasons such as spectral methods being the first and the most heavily studied approach in the block model literature, our choice is also influenced by the fact that spectral techniques, particularly spectral clustering, have been the popular choice in practice. This has naturally led to an elevated interest in extending spectral techniques to partition hypergraphs. Discussions in Chapters 3 and 4 make it quite evident that most of the algorithms for uniform hypergraph partitioning or higher order learning that have been proposed in the machine learning literature are closely related to some spectral approach. Even theorists often suggest that different properties of a hypergraph may be gathered from the spectrum of the incidence matrix or the adjacency tensor of a hypergraph (see Chapter 1).

Another important factor that makes spectral hypergraph partitioning more interesting is the wide variety in the spectral algorithms. This is clearly witnessed in the preceding chapters of this thesis. For example, variants of spectral clustering arise from different graph partitioning objectives, but have a similar flavor. On the other hand, direct extensions of these methods to hypergraphs turn out to be quite dissimilar. While HOSVD relies on tensor decompositions and

TTM is based on a trace maximization principle, NH-Cut takes a different route of hypergraph reduction. Yet, an underlying framework binds these approaches, which we have exploited in our analysis of the algorithms. To be precise, the consistency results proved in this thesis depend on the block structure of the adjacency matrix of a reduced graph. While this reduction is explicit in the description of TTM-ext and NH-Cut, the proofs for other methods implicitly exploit this structure. To this end, this thesis reaffirms the observation of Agarwal et al. (2006) that any spectral partitioning method is closely related to a hypergraph reduction based strategy. Subsequently, our consistency results may be easily extended to other spectral methods as well.

While the discussions in this thesis are dedicated to the analysis of spectral methods, it is easy to observe the framework can be used to study the theoretical guarantees of other partitioning approaches. Our extension of the stochastic block model is quite natural, where the essential idea is to generate a set of independent edges with edge probability (or edge weight) governed by the class labels of the participating vertices. Specifically, we extend the sparse stochastic block model presented in (Lei and Rinaldo, 2015), where the edge probability is decomposed as a product of a label dependent $\Theta(1)$ term, and a label independent sparsity factor that is allowed to vary with n. This factorization is particularly useful in our context as we specify the sparsity factor in terms of both the number of nodes n, and the edge size m. This allows one to control the density of edges of different sizes leading to interesting consequences discussed in Chapter 5.

In the case of graphs, the stochastic block model is often extended to account for factors such as degree heterogeneity or overlapping communities (Lei and Rinaldo, 2015; Zhang et al., 2014). Similar extensions of the model for planted hypergraphs are certainly conceivable. However, it also seems possible that some information, such as community overlap, may be lost due to hypergraph reduction. To this end, the following question seems interesting.

> **Question 5** (Alternative methods)**.** Are smaller error rates achievable by algorithms that do not use reduction, and directly exploit the spectrum of the incidence matrix H or the adjacency tensor $\mathbf{A}$?

An affirmative answer seems possible in certain cases since a tensor power iteration based approach is typically known to have exceptional performance in hypergraph matching problems (Nguyen et al., 2015). However, in the problem of detecting a balanced bipartition, it has been recently shown that reduction based approaches are optimal (Florescu and Perkins, 2016).

Another issue we have eluded throughout the thesis is the determination of the number of clusters k. Following the lines of standard spectral clustering, we have made a strong assumption about prior knowledge of k. However, recent results in the block model literature show that a

variety of techniques based on cross validation (Chen and Lei, 2014), likelihood ratio test (Wang and Bickel, 2015) etc. can be used to estimate the number of clusters. It would be interesting to extend such methods to the case of planted hypergraphs, and we feel that positive results can be obtained.

A more crucial question about our consistency results arises at this juncture.

> **Question 6** (Optimality)**.** Consider the planted hypergraph model and the spectral methods presented in this thesis. Are the consistency theorems stated here optimal?

We clarify the meaning of optimality in the present context. This takes us to the block model literature, where sharp error bounds are known for various partitioning approaches. For general approaches such as spectral clustering, weak consistency results are known to hold under an assumption of the sparsity factor as $\alpha_2 = \Omega(\frac{\ln n}{n})$ (Lei and Rinaldo, 2015). On the other hand, for proper 2-coloring of bipartite 3-uniform hypergraphs, one needs to assume $\alpha_2 = \Omega(\frac{1}{n})$ (Chen and Frieze, 1996). From Corollaries 4.5 and 5.12 as well as Theorem 7.1, it appears that our results are not optimal since the sparsity requirements are larger by logarithmic factors. This sub-optimality can be attributed to two factors: analysis of the k-means step, and the matrix concentration bounds used in our results.

Unlike the results in (Rohe et al., 2011; Lei and Rinaldo, 2015), we do not make any assumption on the correctness of the k-means step. This is replaced by a weaker assumption in terms of the minimum sparsity of the hypergraph, which is larger than the standard requirements. Such an assumption is needed to ensure that the rows of the computed eigenvector matrix are ϵ-separable, which is a necessary condition for guaranteeing the correctness of the approximate k-means algorithm (Ostrovsky et al., 2012). This additional analysis addresses a long standing issue in the block model literature, where distance based clustering of eigenvectors using k-means is always assumed to provide near-optimal solutions. We note that the approach taken here is not the only way to tackle the issue. One may also use other methods such as k-balls (Gao et al., 2015) to perform distance based clustering. In such a case, one can rely on alternative versions of our consistency result, for instance Corollary 5.7. The sole reason behind the use and analysis of the k-means step is the widespread use of this technique in practice.

Observe that the analysis of k-means is not required in the case of hypergraph 2-coloring. Yet, the sparsity requirement in this setting (Theorem 7.1) is worse than the condition in (Chen and Frieze, 1996) by a single factor of $\log n$. This factor is also present in the preceding consistency results on hypergraph partitioning, and arises from the use of the matrix Bernstein inequality (Tropp, 2012). Chung and Radcliffe (2011) correctly pointed out that this inequality

is quite useful for graphs with sparsity $\alpha_2 = \Omega(\frac{\log n}{n})$, but fails to provide useful conclusion for sparser graphs. To deal with sparse graphs, Lei and Rinaldo (2015) as well as other authors often rely on sharp concentration bounds for binary adjacency matrices (Friedman et al., 1989). Such a result is not directly applicable in the present context since the adjacency matrices obtained after hypergraph reduction are usually non-binary or weighted (see Chapter 5). Furthermore, the use of matrix Bernstein inequality also allows us extend our results to weighted hypergraphs. It is doubtful whether the tools used by Friedman et al. (1989) can be even extended to deal with weighted graphs. However, if one restricts the discussions to unweighted uniform hypergraphs, the following question seems quite interesting.

> **Question 7** (Tensor concentration)**.** Does a generalization of (Friedman et al., 1989) hold for sparse binary tensors? If so, then what is its implication on the allowable sparsity for community detection in hypergraphs?

In (Ghoshdastidar and Dukkipati, 2015a), we derived a concentration bound for the operator norm or the largest ℓ_2 eigenvalue of dense tensors that is quite similar to the matrix case, but the sparse case has not been studied yet.

Exact recovery of the partition in the sparse regime, $\alpha_2 = \Theta(\frac{1}{n})$, has received considerable attention in recent years. To this end, it suffices to discuss only weak consistency of algorithms since it is now known that one can iteratively refine the solution of a weakly consistent algorithm to exactly recover the partition (Vu, 2014; Lei and Zhu, 2014). We have used this technique in Algorithm COLOR, and believe that the same trick can be extended to more general cases. Typically, achieving consistent spectral methods in the sparse regime is quite challenging, particularly due the disparity of vertex degrees. Typically one uses spectral properties of other alternatives to the adjacency or Laplacian matrices, such as:

- trimmed adjacency matrix (Vu, 2014), where rows of the adjacency matrix with large norms are zeroed out (this is equivalent to removing vertices with large degrees),
- non-backtracking operator defined on the set of edges (Krzakala et al., 2013), which is known to be less sensitive to high degree vertices, or
- regularized adjacency matrix (Le et al., 2015), where a constant factor to the entries to reduce the degree disparity.

It would be quite interesting to study extensions of these quantities to the case of hypergraphs. In this context, we mention that the vertex deletion trick is useful in coloring sparse uniform hypergraphs (Chen and Frieze, 1996).

The question still remains that how sparse a hypergraph can be so that partitioning is possible. Recent works in the block model literature (Decelle et al., 2011; Mossel et al., 2013a; Chen and Xu, 2014) derive this threshold (to exact constants) in the case of graphs. Thus, it is now established below a certain sparsity level, no algorithm can identify the underlying partition, whereas above this threshold, belief propagation (Mossel et al., 2013a) can exactly recover the partition. On the hand, spectral methods (Krzakala et al., 2013; Le et al., 2015) achieve this threshold up to constant factors. One can immediately deduce that this phenomenon is related to a phase transition of planted graphs. Sharp results in more general stochastic block models have also been proved recently (Abbe and Sandon, 2016). In the context of hypergraph partitioning, a question immediately arises.

> **Question 8** (Phase transition)**.** What is the algorithmic barrier for community detection in hypergraphs?

Phase transitions in uniform hypergraphs have been studied in the literature (Achlioptas and Coja-Oghlan, 2008; Panagiotou and Coja-Oghlan, 2012), and thresholds for 2-colorability and boolean satisfiability are known up to constant factors. Recently, Florescu and Perkins (2016) considered the special case of planted bisection and derived the boundary of community detection in this setting. However, the general case of partitioning uniform or non-uniform hypergraphs still remains unexplored.

We now switch gears from the theoretical questions about planted models to practical aspects of hypergraph partitioning. The computational complexity of algorithms has been the primary concern in the modern era of big data and high performance computing. While Chapter 8 shows that practical variants of spectral partitioning algorithms such as Tetris or SCC (Chen and Lerman, 2009) are quite efficient, Chapter 6 reaches to the heart of this problem and inquires into the efficiency of sampling schemes.

Sampling is a well known strategy in both matrix theory and graph theory, particularly in the context of reducing time complexity. A variety of sampling techniques have been developed in the matrix literature, and have been translated to the domain of graph partitioning. Surprisingly, sampling schemes have never surfaced in the stochastic block model literature. As we pointed in Chapter 6, the problem becomes more significant in the case of weighted hypergraphs, where one has arbitrarily large number of edges. Theorem 6.1 resolves this issue by showing that an appropriate sampling strategy helps to drastically reduce the computational complexity of hypergraph partitioning without compromising theoretical merits.

It is needless to say here that while this result provides the perfect starting point, it does not provide a complete solution for tackling large hypergraphs. Indeed, Tetris has a complexity

of $O(n^2)$, which prohibits its use in partitioning hypergraphs with millions of vertices. In such a scenario, it becomes important to study vertex sampling or down-sampling of hypergraphs. While Nyström method has been a popular choice in the case of graphs (Fowlkes et al., 2004), a multi-level paradigm is more common for hypergraph partitioning (Karypis and Kumar, 2000). In (Ghoshdastidar and Dukkipati, 2015b), we empirically observed that a Nyström approximation for tensors may not be the appropriate tool for edge sampling, but its merits in vertex sampling has not been studied yet. Thus, it still remains to be seen whether vertex sampling graphs or hypergraphs can be as efficient and effective as edge sampling.

Perhaps, it would not be an exaggeration to claim that this thesis lays the foundations for a statistical study of hypergraph partitioning. We feel that further studies in this direction will come to light, and will make the literature of planted hypergraphs as rich as that of graphs. At the same time, it will be encouraging to see alternative methods for analyzing hypergraph partitioning algorithms. In particular, the graph literature is enriched with studies on minimax error rates for planted model (Gao et al., 2015), and approximation guarantees on well-clustered graphs (Peng et al., 2015) among others. Extensions of such analysis to hypergraphs will be an useful contribution in the hypergraph partitioning literature.

References

Abbe, E. and C. Sandon (2016). Achieving the ks threshold in the general stochastic block model with linearized acyclic belief propagation. In *Advances in Neural Information Processing Systems*. 150

Achlioptas, D. and A. Coja-Oghlan (2008). Algorithmic barriers from phase transitions. In *Proceedings of 49th Annual Symposium on Foundations of Computer Science*. 6, 11, 35, 119, 150

Agarwal, S., K. Branson, and S. Belongie (2006). Higher order learning with graphs. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 17–24. 12, 31, 32, 33, 66, 69, 83, 120, 147

Agarwal, S., J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie (2005). Beyond pairwise clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 838–845. 2, 3, 32, 34, 66

Ahuja, N. and A. Srivastava (2002). On constrained hypergraph coloring and scheduling. In *Approximation Algorithms for Combinatorial Optimization*, pp. 14–25. Springer Berlin Heidelberg. 36, 119

Aizenman, M. and D. J. Barsky (1987). Sharpness of the phase transition in percolation models. *Communications in Mathematical Physics 108*(3), 489–526. 6

Alon, N. and N. Kahale (1997). A spectral technique for coloring random 3-colorable graphs. *SIAM Journal of Computing 26*, 1733–1748. 30, 87, 95, 119, 120

Alon, N., P. Kelsen, S. Mahajan, and R. Hariharan (1996). Coloring 2-colorable hypergraphs with a sublinear number of colors. *Nordic Journal of Computing 3*, 425–439. 36, 118

Alon, N., M. Krivelevich, and B. Sudakov (1998). Finding a large hidden clique in a random graph. *Random Structures & Algorithms 13*(3-4), 457–466. 30

Alon, N., M. Krivelvich, and B. Sudakov (1998). Finding a large hidden clique in a random graph. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pp. 594–598. 96

Alpert, C. J. (1998). The ISPD98 circuit benchmark suite. In *ISPD '98 Proceedings of the 1998 International Symposium on Physical Design*, pp. 80–85. 131, 132

Alpert, C. J. and A. B. Kahng (1995). Recent directions in netlist partitioning. *Integration, the VLSI Journal 19*(1–2), 1–81. 31, 33

Amini, A. A. and E. Levina (2014). On semi-definite relaxations for the block model. *arXiv preprint arXiv:1406.5647*. 7, 31

Anandkumar, A., R. Ge, D. Hsu, and S. M. Kakade (2014). A tensor approach to learning mixed membership community models. *Journal of Machine Learning Research 15*, 2239–2312. 20

Anandkumar, A., R. Ge, and M. Janzamin (2014). Guaranteed non-orthogonal tensor decomposition via alternating rank-1 updates. *arXiv Preprint arXiv:1402.5180v3*. 20

Andritsos, P., P. Tsaparas, R. J. Miller, and K. C. Sevcik (2004). LIMBO: scalable clustering of categorical data. In *International Confenrence on Extending Database Technology*, pp. 123–146. 136, 137

Arias-Castro, E., G. Chen, and G. Lerman (2011). Spectral clustering based on local linear approximations. *Electronic Journal of Statistics 5*, 1537–1587. 3, 7, 35, 46, 66

Arora, S., S. Rao, and U. V. Vazirani (2004). Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pp. 222–231. 2, 22, 24

Barbara, D., J. Couto, and Y. Li (2002). Coolcat: An entropy-based algorithm for categorical clustering. In *International Conference on Information Knowledge Management*, pp. 582–589. 136

Berge, C. (1984). *Hypergraphs: combinatorics of finite sets*, Volume 45. Elsevier. 2

Berners-Lee, T. and M. Fischett (2000). *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. HarperInformation. 1

Bernstein, F. (1908). Zur theorie der trigonometrischen Reihen. *Leipz. Bet. 60*, 325–338. 2

Bicego, M., A. F. T. Martins, V. Murino, P. M. Q. Aguiar, and M. A. T. Figueiredo (2010, August). 2D shape recognition using information theoretic kernels. In *IEEE International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey, pp. 25–28. 142

Bickel, P. J. and A. Chen (2009). A nonparametric view of network models and newman-girvan and other modularities. *Proceedings of the National Academy of Sciences 106*, 21068–21073. 2, 7

Boley, H. (1977). Directed recursive labelnode hypergraphs: A new representation-language. *Artificial Intelligence 9*(1), 49–85. 2, 3

Bolla, M. (1993). Spectra, euclidean representations and clusterings of hypergraphs. *Discrete Mathematics 117*(1), 19–39. 5, 63, 69, 82

Boutsidis, C., A. Gittens, and P. Kambadur (2015). Spectral clustering via the power method - provably. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*. 109

Bradley, P. S. and O. L. Mangasarian (2000). k-plane clustering. *Journal of Global Optimization 16*, 23–32. 138

Bühler, T. and M. Hein (2009). Spectral clustering based on the graph p-Laplacian. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 81–88. 5

Capitanio, A., A. Nicolau, and N. Dutt (1995). A hypergraph-based model for port allocation on multiple-register-file vliw architectures. *International Journal of Parallel Programming 23*(6), 499–513. 3, 36, 119

Carroll, J. D. and J. Chang (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart?Young decomposition. *Psychometrika 35*, 283–319. 20

Catalyurek, U. V. and C. Aykanat (1999). Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on Parallel and Distributed Systems 10*(7), 673–693. 3

Chaitin, G. J. (1982). Register allocation & spilling via graph coloring. *ACM SIGPLAN Notices - Proceedings of the 1982 SIGPLAN symposium on Compiler construction 17*(6), 98–101. 2

Chen, G. and G. Lerman (2009). Foundations of a multi-way spectral clustering framework for hybrid linear modeling. *Foundations of Computational Mathematics 9*, 517–558. 3, 7, 35, 43, 44, 45, 46

Chen, G. and G. Lerman (2009). Spectral curvature clustering. *International Journal of Computer Vision 81*(3), 317–330. 13, 34, 43, 109, 112, 113, 138, 150

Chen, H. and A. Frieze (1996). Coloring bipartite hypergraphs. In *Integer Programming and Combinatorial Optimization*, pp. 345–358. 4, 7, 8, 36, 118, 119, 120, 122, 148, 149

Chen, K. and J. Lei (2014). Network cross-validation for determining the number of communities in network data. *arXiv preprint arXiv:1411.1715*. 148

Chen, Y., S. Sanghavi, and H. Xu (2014). Improved graph clustering. *IEEE Transactions on Information Theory 60*(10), 6440–6455. 2, 7

Chen, Y. and J. Xu (2014). Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *arXiv preprint arXiv:1402.1267*. 30, 31, 150

Chertok, M. and Y. Keller (2010). Efficient high order matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence 32*(12), 2205–2215. 37

Choi, D. S., P. J. Wolfe, and E. M. Airoldi (2012). Stochastic blockmodels with a growing number of classes. *Biometrika 99*(2), 273–284. 7, 28, 92

Chung, F. (1992). The laplacian of a hypergraph. In *Expanding Graphs: Proceedings of a DIMACS Workshop*, pp. 21–36. 69

Chung, F. and M. Radcliffe (2011). On the spectra of general random graphs. *Electronic Journal of Combinatorics 18*(1), 215–229. 39, 148

Chung, F. R. K. (1997). *Spectral graph theory*, Volume 92. American Mathematical Society. 5, 10, 23, 25

Coja-Oghlan, A. and L. Zdeborová (2012). The condensation transition in random hypergraph 2-coloring. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 119

Comon, P. (2001). From source separation to blind equalization: Contrast based approaches. In *International Conference on Image and Signal Processing*. 68

Cooper, J. and A. Dutle (2012). Spectra of uniform hypergraphs. *Linear Algebra and its Applications 436*(9), 3268–3292. 4

Csiszár, I. (1967). Information-type measures of difference of probability distributions and indirect observation. *Studia Scientiarum Mathematicarum Hungarica 2*, 229–318. 142

Cuturi, M., K. Fukumizu, and J. P. Vert (2005). Semigroup kernels on measures. *Journal of Machine Learning Research 6*, 1169–1198. 142

Darling, R. W. R. and J. R. Norris (2005). Structure of large random hypergraphs. *Annals of Applied Probability 15*(1A), 125–152. 11, 94

Davis, C. and W. M. Kahan (1970). The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis 7*(1), 1–46. 13, 38

De Lathauwer, L., B. De Moor, and J. Vandewalle (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications 21*(4), 1253–1278. 20

de Souto, M. C. P., I. G. Costa, D. S. A. de Araujo, T. B. Ludermir, and A. Schliep (2008). Clustering cancer gene expression data: a comparative study. *BMC bioinformatics 9*(1), 497. 143, 144, 145

Decelle, A., F. Krzakala, C. Moore, and L. Zdeborová (2011). Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E 84*(066106). 2, 31, 150

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological) 39*(1), 1–38. 5

Dinur, I., O. Regev, and C. D. Smyth (2005). The hardness of 3-uniform hypergraph coloring. *Combinatorica 25*(1), 519–535. 118

Drineas, P., R. Kannan, and M. W. Mahoney (2006). Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing 36*(1), 132–157. 112

Duchenne, O., F. Bach, I.-S. Kweon, and J. Ponce (2011). A tensor-based algorithm for high-order graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence 33*(12), 2383–2395. 13, 37, 67, 109, 112

Dyer, E. L., A. C. Sankaranarayanan, and R. G. Baraniuk (2013). Greedy feature selection for subspace clustering. *Journal of Machine Learning Research 14*(1), 2487–2517. 138

Elhamifar, E. and R. Vidal (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*(11), 2765–2781. 138

Endres, D. M. and J. E. Schindelin (2003). A new metric for probability distributions. *IEEE Transactions on Information Theory 49*(7), 1858–1860. 142

Erdös, P. (1963). On a combinatorial problem. *Nordisk Mat. Tidskr 11*, 5–10. 118

Erdös, P. and A. Rényi (1959). On random graphs I. *Publicationes Mathematicae 6*, 290–297. 6, 11, 29

Feldman, V., E. Grigorescu, L. Reyzin, S. Vempala, and Y. Xiao (2012). Statistical algorithms and a lower bound for planted clique. *Electronic Colloquium on Computational Complexity 19*(64). 30

Feldman, V., W. Perkins, and S. Vempala (2015). On the complexity of random satisfiability problems with planted solutions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pp. 77–86. 11

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal 23*(2), 298–305. 2

Florescu, L. and W. Perkins (2016). Spectral thresholds in the bipartite stochastic block model. In *Proceedings of the Conference on Learning Theory (COLT)*, pp. 943–959. 119, 147, 150

Fowlkes, C., S. Belongie, F. Chung, and J. Malik (2004). Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*(2), 214–225. 151

Friedman, J., J. Kahn, and E. Szemeredi (1989). On the second eigenvalue of random regular graphs. In *Proceedings of the twenty-first annual ACM Symposium on Theory of Computing*. 122, 149

Friedman, J. and A. Wigderson (1995). On the second eigenvalue of hypergraphs. *Combinatorica 15*(1), 43–65. 5

Gao, C., Z. Ma, A. Y. Zhang, and H. H. Zhou (2015). Achieving optimal misclassification proportion in stochastic block model. *arXiv preprint arXiv:1507.03772*. 13, 27, 28, 31, 89, 148, 151

Garcia-Garcia, D. and R. C. Williamson (2012, June). Divergences and risks for multiclass experiments. In *25th Annual Conference on Learning Theory*, Edinburgh, Scotland. 142

Garey, M. R. and D. S. Johnson (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman & Co. 1

Ghoshal, G., V. Zlatic, G. Caldarelli, and M. E. J. Newman (2009). Random hypergraphs and their applications. *Physical Review E 79*(066118). 2, 131

Ghoshdastidar, D., A. P. Adsul, and A. Dukkipati (2016). Learning with jensen-tsallis kernels. *IEEE Transactions on Neural Networks and Learning Systems (In press)*. 142

Ghoshdastidar, D. and A. Dukkipati (2014). Consistency of spectral partitioning of uniform hypergraphs under planted partition model. In *Advances in Neural Information Processing Systems*. 43

Ghoshdastidar, D. and A. Dukkipati (2015a). A provable generalized tensor spectral method for uniform hypergraph partitioning. In *Proceedings of the International Conference on Machine Learning*. 112, 149

Ghoshdastidar, D. and A. Dukkipati (2015b). Spectral clustering using multilinear svd: Analysis, approximations and applications. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 141, 151

Ghoshdastidar, D. and A. Dukkipati (2016). Consistency of spectral hypergraph partitioning under planted partition model. *The Annals of Statistics (In press, arXiv:1505.01582)*. 137

Ghoshdastidar, D., A. Dukkipati, A. P. Adsul, and A. Vijayan. (2014, June). Spectral clustering with jensen-type kernels and their multi-point extensions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio. IEEE. 142

Gibson, D., J. Kleinberg, and P. Raghavan (2000). Clustering categorical data: An approach based on dynamical systems. *VLDB Journal 8*(3-4), 222–236. 2, 3, 33, 136

Gilbert, E. N. (1959). Random graphs. *Annals of Mathematical Statistics 30*(4), 1141–1144. 6

Girvan, M. and M. E. J. Newman (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences 99*(12), 7821–7826. 2, 22

Govindu, V. M. (2005). A tensor decomposition for geometric grouping and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1150–1157. 3, 4, 12, 14, 34, 35, 42, 43, 109, 112

Guha, S., R. Rastogi, and K. Shim (2000). Rock: A robust clustering algorithm for categorical attributes. *Inform. Syst. 25*(5), 345–366. 2, 136

Guimera, R. and L. A. N. Amaral (2005). Functional cartography of complex metabolic networks. *Nature 433*(7028), 895–900. 2

Hadley, S. W. (1995). Approximation techniques for hypergraph partitioning problems. *Discrete Applied Mathematics 59*(2), 115–127. 4, 32

Han, E. H., G. Karypis, V. Kumar, and B. Mobasher (1997). Clustering based on association rule hypergraphs. In *SIGMOD'97 Workshop on Research Issues on Data Mining and Knowledge Discovery*. 33, 136, 137

Heckel, R. and H. Bölcskei (2013). Subspace clustering via thresholding and spectral clustering. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*. 138

Hitchcock, F. L. (1927). The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics 6*, 164–189. 20

Holland, P. W., K. B. Laskey, and S. Leinhardt (1983). Stochastic blockmodels: First steps. *Social networks 5*(2), 109–137. 6, 29

Horn, R. A. and C. R. Johnson (2013). *Matrix analysis.* (2 ed.). Cambridge University Press. 17

Hu, S. and L. Qi (2012). Algebraic connectivity of even uniform hypergraph. *Journal of Combinatorial Optimization 24*, 564–579. 4, 63, 82

Ihler, E., D. Wagner, and F. Wagner (1993). Modelling hypergraphs by graphs with the same mincut properties. *Information Processing Letters 45*(4), 171–175. 32

Jain, P. and S. Oh (2014). Provable tensor factorization with missing data. In *Advances in Neural Information Processing Systems*, pp. 1431–1439. 20

Jain, S. and V. M. Govindu (2013). Efficient higher-order clustering on the grassmann manifold. In *IEEE International Conference on Computer Vision*. 34, 35, 43, 109, 112, 138, 141

Jerrum, M. (1992). Large cliques elude the metropolis process. *Random Structures & Algorithms 3*(4), 347–360. 30

Jung, H., J. Ju, and J. Kim (2014). Rigid motion segmentation using randomized voting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1210–1217. 141

Kannan, R., S. Vempala, and A. Vetta (2004). On clusterings: Good, bad and spectral. *Journal of the ACM 51*(3), 497–515. 5, 28

Karrer, B. and M. E. J. Newman (2011). Stochastic blockmodels and community structure in networks. *Physical Review E 83*(1), 016107. 31

Karypis, G. and V. Kumar (2000). Multilevel k-way hypergraph partitioning. *VLSI Design 11*(3), 285–300. 2, 3, 4, 31, 32, 135, 137, 151

Kernighan, B. W. and S. Lin (1970). An efficient heuristic procedure for partitioning graphs. *Bell system technical journal 49*(2), 291–307. 2, 32

Khot, S. (2002). Hardness results for coloring 3-colorable 3-uniform hypergraphs. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 23–32. 4

Khot, S. and R. Saket (2014). Hardness of finding independent sets in 2-colorable and almost 2-colorable hypergraphs. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1607–1625. 4, 36, 118

Kolda, T. G. and B. W. Bader (2009). Tensor decompositions and applications. *SIAM Review 51*(3), 455–500. 20

Krzakala, F., C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang (2013). Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences 110*(52), 20935–20940. 7, 31, 149, 150

Kumar, A., Y. Sabharwal, and S. Sen (2004). A simple linear time $(1+\epsilon)$-approximation algorithm for geometric k-means clustering in any dimensions. In *Proceedings-Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 454–462. 13, 27, 40, 89

Le, C. M., E. Levina, and R. Vershynin (2015). Sparse random graphs: Regularization and concentration of the Laplacian. *arXiv preprint arXiv:1502.03049*. 31, 86, 149, 150

Lee, J., M. Cho, and K. M. Lee (2011). Hyper-graph matching via reweighted random walk. In *IEEE Conference on Computer Vision and Pattern Recognition*. 37, 67

Lee, J. R., S. O. Gharan, and L. Trevisan (2012). Multi-way spectral partitioning and higher-order Cheeger inequalities. In *Proceedings of the Forty-Fourth Annual ACM on Symposium on Theory of Computing*, pp. 1117–1130. 25

Lei, J. and A. Rinaldo (2015). Consistency of spectral clustering in stochastic block models. *Annals of Statistics 43*. 7, 13, 24, 29, 30, 31, 38, 71, 86, 88, 91, 92, 147, 148, 149

Lei, J. and L. Zhu (2014). A generic sample splitting approach for refined community recovery in stochastic block models. *arXiv preprint arXiv:1411.1469*. 7, 31, 149

Lettvin, J. Y., H. R. Maturana, W. S. McCulloch, and W. H. Pitts (1959). What the frog's eye tells the frog's brain.s. *Proceedings of the Institute of Radio Engineers 47*, 1940–1951. 1

Lichman, M. (2013). UCI machine learning repository. http://archive.ics.uci.edu/ml. 136, 143

Lim, L.-H. (2005). Singular values and eigenvalues of tensors: a variational approach. In *Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pp. 129–132. 21, 68

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Info. Theory 37*, 145–151. 142

Liu, G., Z. Lin, and Y. Yu (2010). Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*. 138

Liu, H., L. J. Latecki, and S. Yan (2010). Robust clustering as ensembles of affinity relations. In *Advances in Neural Information Processing Systems*, pp. 1414–1422. 35, 37, 67

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory 28*(2), 129–137. 4, 13, 27

Lu, C.-J. (2004). Deterministic hypergraph coloring and its applications. *SIAM Journal on Discrete Mathematics 18*(2), 320–331. 36, 119

Luce, D. R. and A. D. Perry (1949). A method of matrix analysis of group structure. *Psychometrika 14*(2), 95–116. 2

Martins, A. F. T., N. A. Smith, E. P. Xing, P. M. Q. Aguiar, and M. A. T. Figueiredo (2009). Nonextensive information theoretic kernels on measures. *Journal of Machine Learning Research 10*, 935–975. 142

McDiarmid, C. (1993). A random recolouring method for graphs and hypergraphs. *Combinatorics, Probability and Computing 2*(3), 215–229. 119

McSherry, F. (2001). Spectral partitioning of random graphs. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 529–537. 6, 7, 29

Melacci, S. and M. Gori (2012). Unsupervised learning by minimal entropy encoding. *IEEE Transactions on Neural Networks and Learning Systems 23*(12), 1849–1861. 143, 144

Michoel, T. and B. Nachtergaele (2012). Alignment and integration of complex networks by hypergraph-based spectral clustering. *Physical Review E 86*(056111). 2, 3, 67, 68

Mossel, E., J. Neeman, and A. Sly (2013a). Belief propagation, robust reconstruction, and optimal recovery of block model. *arXiv preprint arXiv:1309.1380*. 7, 31, 150

Mossel, E., J. Neeman, and A. Sly (2013b). A proof of the block model threshold conjecture. *arXiv preprint arXiv:1311.4115*. 71

Ng, A., M. Jordan, and Y. Weiss (2002). On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pp. 849–856. 2, 5, 7, 24, 42, 44, 65, 143

Nguyen, Q., A. Gautier, and M. Hein (2015). A flexible tensor block coordinate ascent scheme for hypergraph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5270–5278. 37, 147

Nielsen, F. and R. Nock (2013). Total jensen divergences: Definition, properties and k-means++ clustering. *arXiv preprint arXiv:1309.7109*. 142

Ostrovsky, R., Y. Rabani, L. J. Schulman, and C. Swamy (2012). The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM 59*(6), 28:1–28. 4, 13, 27, 40, 46, 51, 53, 70, 83, 88, 89, 109, 148

Panagiotou, K. and A. Coja-Oghlan (2012). Catching the k-NAESAT threshold. In *ACM Symposium on Theory of Computing (STOC)*. 6, 119, 150

Park, D., C. Caramanis, and S. Sanghavi (2014). Greedy subspace clustering. In *Advances in Neural Information Processing Systems*, Volume 27, pp. 2753–2761. 138, 139, 140

Peng, R., H. Sun, and L. Zanetti (2015). Partitioning well-clustered graphs: Spectral clustering works! In *Proceedings of the Annual Conference on Learning Theory*, pp. 1423–1455. 5, 24, 151

Qi, L. (2005). Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation 40*(6), 1302–1324. 4, 11, 22

Radhakrishnan, J. and A. Srinivasan (1998). Improved bounds and algorithms for hypergraph two-coloring. In *Proceedings of 39th Annual Symposium on Foundations of Computer Science*, pp. 684–693. 118

Raghavendra, P. and T. Schramm (2015). Tight lower bounds for planted clique in the degree-4 SOS program. *arXiv preprint arXiv:1505.05136*. 30

Rangapuram, S. S., P. K. Mudrakarta, and M. Hein (2014). Tight continuous relaxation of the balanced k-cut problem. In *Advances in Neural Information Processing Systems*, Volume 27, pp. 3131–3139. 5

Rodríguez, J. A. (2002). On the laplacian eigenvalues and metric parameters of hypergraphs. *Linear and Multilinear Algebra 50*, 1–14. 4, 82, 137

Rohe, K., S. Chatterjee, and B. Yu (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *Annals of Statistics 39*(4), 1878–1915. 7, 28, 30, 45, 53, 70, 71, 88, 92, 148

Rota Bulo, S. and M. Pelillo (2013). A game-theoretic approach to hypergraph clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence 35*(6), 1312–1327. 4, 35, 67, 135

Schmidt, J. P. (1987). Probabilistic analysis of strong hypergraph coloring algorithms and the strong chromatic number. *Discrete Mathematics 66*(3), 259–277. 36

Schmidt-Pruzan, J. and E. Shamir (1985). Component structure in the evolution of random hypergraphs. *Combinatorica 5*(1), 81–94. 11

Schweikert, G. and B. W. Kernighan (1979). A proper model for the partitioning of electrical circuits. In *Proceedings of 9th Design Automation Workshop*, Dallas, pp. 57–62. 3, 4

Shashua, A., R. Zass, and T. Hazan (2006). Multi-way clustering using super-symmetric non-negative tensor factorization. In *European Conference on Computer Vision*, pp. 595–608. 35, 66, 135

Shi, J. and J. Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(8), 888–905. 2, 5, 24, 64

Shi, T., M. Belkin, and B. Yu (2009). Data spectroscopy: Eigenspaces of convolution operators and clustering. *The Annals of Statistics 37*(6), 3960–3984. 6, 28

Sibson, R. (1969). Information radius. *Probability Theory and Related Fields 14*, 149–160. 142, 143

Simonovits, M. and V. T. Sós (1991). Szemer'edi's partition and quasirandomness. *Random Structures and Algorithms 2*(1). 29

Sperner, E. (1928). Ein satz aber untermengen einer endlichen menge. *Mathematische Zeitschrift 27*(1), 544–548. 2

Spielman, D. A. (2011). Spectral graph theory. In U. Naumann and O. Schenk (Eds.), *Combinatorial Scientific Computing*, Chapter 18, pp. 495–524. CRC Press. 10, 22, 24

Stasi, D., K. Sadeghi, A. Rinaldo, S. Petrović, and S. E. Fienberg (2014). β models for random hypergraphs with a given degree sequence. *arXiv preprint arXiv:1407.1004*. 11

Stewart, G. W. and J. Sun (1990). *Matrix Perturbation Theory*. Academic Press. 13, 38, 61, 90, 102

Tomasi, C. and T. Kanade (1992). Shape and motion from image streams under orthography. *International Journal of Computer Vision 9*(2), 137–154. 140

Tron, R. and R. Vidal (2007). A benchmark for the comparison of 3-D motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*. 140

Tropp, J. A. (2012). User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics 12*(4), 389–434. 12, 39, 73, 91, 122, 148

Tsallis, C. (1988). Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statiscal Physics 52*, 479–87. 142

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika 31*, 279–311. 20

Vapnik, V. N. (1998). *Statistical learning theory*, Volume 1. Wiley. 4, 5

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing 17*(4), 395–416. 5, 22, 23, 59

von Luxburg, U., M. Belkin, and O. Bousquet (2008). Consistency of spectral clustering. *The Annals of Statistics 36*(2), 555–586. 5, 6, 8, 28

Vu, V. (2014). A simple SVD algorithm for finding hidden partitions. *arXiv preprint arXiv:1404.3918*. 7, 28, 31, 149

Wang, Y. X. and P. Bickel (2015). Likelihood-based model selection for stochastic block models. *arXiv preprint arXiv:1502.02069*. 148

Wasserman, S. (1994). *Social network analysis: Methods and applications*. Cambridge university press. 2

Weyl, H. (1912). Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen. *Mathematische Annalen 71*, 441–479. 37

Whitney, H. (1935). On the abstract properties of linear dependence. *American Journal of Mathematics 57*(3), 509–533. 2

Wu, S., S. Wei, Y. Wang, R. Vaidyanathan, and J. Yuan (2015). Partition information and its transmission over partition information and its transmission over boolean multi-access channels. *IEEE Transactions on Information Theory 61*(2), 1010–1027. 3

Zass, R. and A. Shashua (2006). Doubly stochastic normalization for spectral clustering. In *Advances in Neural Information Processing Systems*. 65

Zha, H., X. He, C. Ding, H. Simon, and M. Gu (2001). Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems*. 65

Zhang, Y., E. Levina, and J. Zhu (2014). Detecting overlapping communities in networks with spectral methods. *arXiv preprint* (arXiv:1412.3432v2). 31, 147

Zhou, D., J. Huang, and B. Schölkopf (2007). Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1601–1608. 3, 4, 5, 12, 15, 63, 82, 83