UNIVERSITY OF CALIFORNIA, SAN DIEGO

Learning from Higher Order Relations

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Sameer Agarwal

Committee in charge:

Professor Serge J. Belongie, Chair
Professor Ian Abramson
Professor Fan Chung Graham
Professor Henrik Wann Jensen
Professor David Kriegman

2006

The dissertation of Sameer Agarwal is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2006

To my parents

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The first person I want to express my gratitude towards is my advisor, Professor Serge Belongie. Since I first met him in the fall of 2001, Serge has been a source of constant friendship, ideas, guidance and advice. Each of the two problems that have been considered in this dissertation are direct consequences of conversations with him. Professor David Kriegman has always treated me as one of his advisees. His generosity with his time, advice, ideas and willingness to patiently listen to me as I rambled on about this and that is much appreciated.

Professor Henrik Wann Jensen gave me the chance to apply clustering to a real life problem and taught me much of what I know about computer graphics. Professor Fan Chung Graham and Professor Ian Abramson not only agreed to be on my research exam and dissertation committees, but have also been generous with their time whenever I have had a question. Professor Graham was also responsible for supporting me through the last year of my study. Professor Charles Elkan and Professor Garrison Cottrell are responsible for me choosing UCSD for graduate studies and supported me financially for the first year and a half of my stay at UCSD. Professor Sanjoy Dasgupta is yet to complain about my random and mostly unannounced visits to his office with questions. Fredrik Kahl got me started with convex optimization and Professor Gert Lanckriet has helped me to continue pursuing it. He is an unending source of ideas and information about convex optimization. Even though we did not always agree, Professor Pietro Perona and Lihi Zelnik-Manor helped me clarify a number of my ideas about higher order clustering.

Life in graduate school is rough without friends and collaborators. Fortunately, I have been blessed with the company of friends who have often been excellent collaborators too. Josh Wills has been my office mate for a good part of five years and in that time we have shared much of the ups and down of graduate life. Along the way we worked on a number of problems together, some attempts more successful than others. Kristin Branson has listened to far too many of my cooky ideas, worked with me on more than a few of them and corrected my writings too many times to count. Satya Prakash Mallick has been a dear friend, a close confidant and collaborator. More than anything else, he has taught me the value of sheer determination. Manmohan Chandraker taught me various bits of multiview geometry and autocalibration, and has been my partner in crime in writing some of the hairiest MATLAB code ever written. Given our shared love of eating, both Satya and Manmohan have also been willing and cheerful subjects

for my various cooking experiments. Eric Wiewiora (the man who ate everything) and Lawrence Cayton are the kind of people one wants as one's neighbors in graduate school. People you can go and talk to about the random mist swirling inside your head. Kristin and Lawrence were also kind enough to give Chapters 1 and 3 a thorough read. The credit for these chapters being readable goes to them. The errors are of course mine.

I would also like to thank, Craig Donner, Andrew Rabinovich, Piotr Dollar, Ben Ochoa, Will Chang, Vincent Rabaud, William Beaver and Jongwoo Lim, Matt Dailey, Bianca Zadrozny, Hector Jasso and Greg Hamerly for making life in graduate school and in particular the Pixel Lab and the Artificial Intelligence Lab fun and interesting.

Outside school, Jayne Catlin has been a constant. Over the years she has been a source of much love, support & encouragement. Everyone needs kind, generous and caring friends like Manish and Surabhi Goyal.

My uncle and aunt Shyam and Manjula's house has been my home away from home. Sachin, Richa and Neal have always indulged their older and slightly idiosyncratic brother. And last but not the least, I would like to thank my parents Santosh and Usha. You have taught me the importance of pursuing my dreams. I dedicate this dissertation to you.

Portions of this dissertation are based on papers that I have co-authored with others. Listed below are my contributions to each of these papers.

1. Chapter 2 is in part based on the paper "Higher Order Learning with Graphs" by S. Agarwal, K. Branson, S. Belongie [2]. I developed the primary equivalence between star expansion and clique expansion, did the literature survey for the paper and contributed to the writing of the paper.

2. Chapter 1 and Chapter 2 are in part based on the paper "Beyond Pairwise Clustering," by S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman and S. Belongie [4]. I was responsible for the development and analysis of the Clique Averaging algorithm, literature survey, experiment design. I also contributed to the execution and analysis of the experiments and the writing of the paper.

3. Chapter 1 and Chapter 3 are in part based on the paper "Toward a Perceptual Space for Reflectance" by J. Wills, S. Agarwal, D. Kriegman and S. Belongie [141]. I was responsible for the development of the non-metric multidimensional scaling algorithm, performed the data analysis, helped with the psychophysics experiments and contributed to the writing of the paper.

VITA

1977	Born, Bulandshahar, Uttar Pradesh, India.
2000	M. S. (Integrated), Indian Institute of Technology, Kanpur.
2006	Ph. D., University of California, San Diego.

PUBLICATIONS

A. Rabinovich, S. Agarwal, S. Krajewski, J. Reed, J.H. Price and S. Belongie, “Accuracy of Unsupervised Spectral Decomposition for Densitometry of Histological Sections”, *In Review.*

J. Wills, S. Agarwal, D. Kriegman and S. Belongie, “Toward a Perceptual Space for Reflectance”, *In Review.*

J. Wills, S. Agarwal and S. Belongie, “A Feature-based Approach for Dense Segmentation and Estimation of Large Disparity Motion,” *International Journal of Computer Vision*, **68**(2):125–143, June 2006.

S. Agarwal, K. Branson, S. Belongie, “Higher Order Learning with Graphs”, *To Appear, International Conference on Machine Learning, 2006.*

S. P. Mallick, S. Agarwal, D. Kriegman, S. Belongie, B. Carragher, C. Potter, “Structure and View Estimation for Tomographic Reconstruction: A Bayesian Approach”, *To Appear, Computer Vision and Pattern Recognition, 2006.*

S. Agarwal, M. Chandraker, F. Kahl, D. Kriegman and S. Belongie, “Practical Global Optimization for Multiview Geometry”, *To Appear, European Conference on Computer Vision , 2006.*

S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman and S. Belongie, “Beyond Pairwise Clustering,” *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2005, pp. 838-845.

S. Agarwal, S. P. Mallick, D. Kriegman and S. Belongie, “On Refractive Optical Flow,” *Proceedings of the European Conference on Computer Vision*, 2004, pp. 483-494, vol. 2.

S. Agarwal, R. Ramamoorthi, S. Belongie and H.W. Jensen, “Structured Importance Sampling of Environment Maps,” *ACM Transactions on Graphics – Proceedings of SIGGRAPH*, **22**(3):605-612, July 2003.

A. Rabinovich, S. Agarwal, C. Laris, J. Price, and S. Belongie “Unsupervised Color Decomposition of Histologically Stained Tissue Samples” *Neural Information Processing Systems*, 2003.

J. Wills, S. Agarwal and S. Belongie, “What Went Where,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 37-44, vol. 1.

K. Deb, A.P. Mathur, S. Agarwal and T. Meyrivan, “A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, **6**(2):182-197, April 2002.

S. Agarwal and S. Belongie, “On the Non-Optimality of Four Color Coding of Image Partitions,” *International Conference on Image Processing*, 2002, pp. 677-680, vol. 2.

K. Morikawa, S. Agarwal, C. Elkan and G. Cottrell, “A Taxonomy of Computational and Social Learning”, *Workshop on Developmental Embodied Cognition*, 2001.

H. Kargupta, E.R. Sanseverino, E. Johnson and S. Agarwal, “The Genetic Algorithm, Linkage Learning and Scalable Data Mining”, *Intelligent Data Analysis in Science*, Hugh Cartwright, editor, Oxford University Press, 2000.

K. Deb, S. Agarwal A.P. Mathur and T. Meyrivan, “A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II,” *Proceedings of the Parallel Problem Solving from Nature* **IV** *Conference*, 2000.

K. Deb and S. Agarwal, “Understanding Interactions Between Genetic Algorithm Parameters,” *Foundations of Genetic Algorithms,* 1999.

K. Deb and S. Agarwal, “A Niched-penalty Approach for Constraint Handling in Genetic Algorithms,”*Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, 1999.

ABSTRACT OF THE DISSERTATION

Learning from Higher Order Relations

by

Sameer Agarwal
Doctor of Philosophy in Computer Science

University of California San Diego, 2006

Professor Serge J. Belongie, Chair

In a number of domains in computer vision, machine learning and psychology, it is common to model and analyze data in terms of pairwise interactions between data elements, that is, distances between pairs of points. A large variety of supervised and unsupervised algorithms exist for learning the structure of such data. However, it is not always the case that interactions between entities can be described in terms of pairs; often, higher order relations of size three and higher offer a more natural formulation.

Consider the k-lines clustering problem, where the objective is to group data points in a d-dimensional vector space into k clusters where elements in each cluster are well approximated by a line. As every pair of data points trivially define a line, there does not exist a useful measure of similarity between pairs of points for this problem. However, it is possible to define measures of similarity over triplets of points that indicate how close they are to being collinear. Another example is the task of constructing an embedding obtained from psychometric experiments involving *paired comparisons*. A common paired comparison experiment is where subjects are shown three images – X, Y and Z – and asked to report which of X or Y is more similar to Z. Here each observation is a triadic relation on the set of images. We refer to the problem of learning from such data-sets as higher order learning.

In this dissertation we present solutions to two problems in higher order learning. The first is a new technique for clustering called Clique Averaging, in which we use a novel generative model for hypergraphs to construct a clustering algorithm that has better theoretical as well as empirical performance than existing algorithms. We show applications to illumination invariant clustering and k-subspace learning. The second is a new non-metric multi-dimensional scaling (MDS) algorithm for ordinal data, that is,

data sets in which instead of the magnitude, only the relative or rank order of distances is known. We will show how this algorithm can be used to construct a perceptual space for reflectance.

1

Introduction

> *"Come, Watson, come!" he cried. "The game is afoot. Not a word! Into your clothes and come!"*
>
> *Sherlock Holmes,*
> *The Adventure of the Abbey Grange*

By all accounts we are living in the information age: a time in which the speed at which we can communicate information far exceeds any physical movement that we are capable of. With every passing moment, the amount of information in our stores increases, be it the amount of email in our mailbox, information about the nucleotide structures of animal and plant DNA, video surveillance footage of the incoming patrons at a casino or measurements of how different materials reflect light. The various bits of information we collect, and, the sources that we collect them from, are heterogeneous in structure with complex underlying causes and much of our scientific and technological progress depends on being able to identify and understand these causes.

The subject of this dissertation is the development of data analysis methods that are efficient and effective both in theory and in practice. In particular, we consider two of the most common tasks performed by an investigator when faced with the problem of making sense of a data set about which very little or nothing is known *a priori*—clustering and multidimensional Scaling.

1.1 Clustering

Clustering is the task of partitioning a data set into a number of groups such that elements in the same group or cluster are more similar to each other than elements in different groups. It is a widely used tool with applications in just about every field of study where there is data to be analyzed. Some examples are computer vision, statistics, machine learning, computer graphics, psychometrics, marketing, numerical linear algebra and VLSI CAD. For example, in the case of perceptual grouping, clustering is used to understand the large scale structure of images by ignoring small local variations. In other cases like vector quantization it is used for approximating a large data set with a small one. In numerical linear algebra, clustering algorithms are used to make decisions about how large matrices should be distributed across the processors of a parallel computer so that communication costs are minimized and in VLSI CAD, clustering algorithms decide how the transistors on a silicon wafer should be laid out such that the cost of wiring the circuit is minimized [8, 40, 41, 58, 66, 94, 101, 106, 109, 121].

There are two dominant approaches to clustering. The first approach is based on assuming a generative model for the data, and we will refer to it as *Model Based Clustering*. The second approach is based on a more non-parametric and discriminative view of the clustering process, and we will refer to it as *Pairwise Clustering*.

1.1.1 Model Based Clustering

Model based clustering methods are, as the name suggests based on the idea that there is an underlying stochastic model that was responsible for generating the data, say for example a mixture of Gaussian distributions. The clustering algorithm then is essentially a model fitting algorithm that finds a partitioning of the data that minimizes the fitting error. Let us consider as an example the k-means algorithm [41, 92].

Suppose we are given a data matrix $X = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ of n points in a d dimensional vector space, and a natural number $k > 1$, the number of groups we wish to partition this data set into. Then, perhaps the most common method for clustering the columns of X is to model each partition with a single point, the cluster center. Let $Y = [\mathbf{y}_1, \dots, \mathbf{y}_k] \in \mathbb{R}^{d\times k}$ be the set of cluster centers and let $C = \{C_i\}$ be a disjoint partitioning of the data such that C_i denotes the set of points in the i^{th} cluster, then we define the cost or error of this clustering as

$$e_{C,Y} = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \|\mathbf{y}_i - \mathbf{x}_j\|_2^2 \tag{1.1}$$

The aim is to find that set of cluster centers Y and partition C such that $e_{C,Y}$ is minimized. The minimization of $e_{C,Y}$ is a mixed-integer non-linear programming problem, which is NP-Hard [57]. Thus, in practice an approximate solution based on an alternating minimization procedure is used. The algorithm alternates between optimizing over C given Y, which is a simple nearest neighbor assignment problem, and optimizing over Y given C, which can be shown to be equivalent to averaging the elements in each cluster.

The k-means algorithm is a close relative of the well known Expectation-Maximization (EM) algorithm [37]. The EM algorithm is a widely used method for Maximum Likelihood estimation of model parameters. The k-means algorithm can be thought of as a discrete version of the EM algorithm where the assignment of points to clusters is the expectation step and the estimation of cluster centers is the maximization step. Indeed, the EM algorithm is often a part of model based clustering algorithms.

1.1.2 Pairwise Clustering

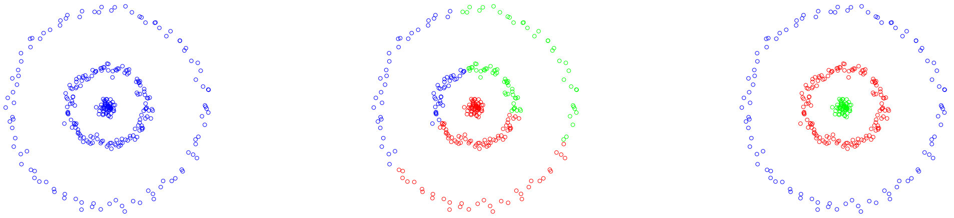
Pairwise clustering methods take a discriminative view of the clustering process, in that they do not have an explicit model for each cluster. Instead, these approaches are characterized by the use of a clustering objective function that tries to maximize the separation between clusters while promoting compactness within a cluster. Pairwise clustering methods can be considered analogs of the discriminative approach to pattern classification where the shape of the boundary between pattern classes is the focus of study instead of the model that gives rise to the pattern classes themselves.

As an example of pairwise clustering, let us consider the *single linkage algorithm* In this algorithm, every element of the data set starts out as its own cluster and at each iteration of the algorithm, the two closest clusters are identified and merged into one. The process continues for $n-k-1$ iterations till k clusters have been identified. A number of different choices are possible for determining the distance between two clusters, a simple choice is to define it as the distance between their two closest points, i.e.

$$d(C, C') = \min_{\mathbf{x} \in C, \mathbf{x}' \in C'} \|\mathbf{x} - \mathbf{x}'\|_2 \tag{1.2}$$

Now, even though the above algorithm was stated in terms of the vectors $\mathbf{x}$, a vector representation for the points is not needed. A matrix $D = [d_{ij}]$ of pairwise distances, dissimilarities or similarities suffices. This is a characteristic of this class of clustering methods and thus the name *Pairwise* clustering. In many applications, for example perceptual grouping, an explicit vector representation of the data may not be available or might not even exist, but pairwise measures of similarity between pixels are easy to construct. In such cases pairwise clustering algorithms have an advantage over model based algorithms which typically require an explicit representation of the data.

A variety of sophisticated pairwise clustering algorithms have been developed recently that have excellent performance in practice [101, 121]. Sometimes, even when the data is available in a vector form, it is converted into a pairwise dissimilarity matrix for use with a pairwise clustering algorithm.



(a) Data set

(b) k-means

(c) Spectral clustering

Figure 1.1 An example of a data set on which k-means fails while a spectral clustering algorithm successfully partitions it into three concentric annuli. (a) The original dataset. (b) Result of running k-means on the dataset. (c) Result of running a Normalized cuts based spectral clustering algorithm. In (b) and (c) the three colors indicate membership in the three partitions reported by the respective clustering algorithms.

Modeling the structure of clusters is a hard task. Assumptions that lead to tractable algorithms are not necessarily true for the problems that need to be solved, e.g., assuming that data in each cluster was generated from a Gaussian distribution centered at the cluster center. The fact that pairwise clustering methods do not make such strong model assumptions is perhaps the single most important reason for their success and popularity. This success of course does not come without a cost. Pairwise clustering algorithms, especially those based on spectral analysis of similarity matrices have a significantly higher computational cost than model based approaches. Further, pairwise clustering algorithms require the specification of a function that measures the similarity between pairs of points. While at first it may appear that the specification of this function is equivalent to choosing a model in model based clustering algorithms, this is not entirely correct. The difference here is that the similarity measure only acts locally, i.e., it operates on just two data points at a time as opposed to model based clustering where all the data in a cluster has to fit the same model. This means that the assumptions this function makes about the structure of the clusters only need to be satisfied locally.

Consider for example the task of partitioning the point set shown in Figure 1.1(a). It is made up of three concentric annuli that we wish to separate into three different clusters. The model assumption made by k-means that each cluster is generated by stochastically perturbing a single point isotropically, i.e., the assumption that all points in a cluster are close to the cluster center is clearly not true here, and this is reflected in the quality of the clustering obtained by running the k-means algorithm as illustrated in Figure 1.1(b). On the other hand, it is reasonable to assume that two points $\mathbf{x}$ and $\mathbf{x}'$ which are close-by have a good chance of belonging to the same cluster, a heuristic that is captured by our use of the similarity measure $e^{-\|\mathbf{x}-\mathbf{x}'\|_2^2/\sigma}$. Here, σ is a scale parameter, which can either be specified by the user or can be selected automatically as was done in this example [143]. Figure 1.1(c) shows the results. A perfect clustering is achieved.

1.1.3 Beyond Pairwise Clustering

It is not always the case, however, that there exists a similarity measure for pairs of data points. For some clustering problems, one may need to consider three or more data points together to determine if they belong to the same cluster.

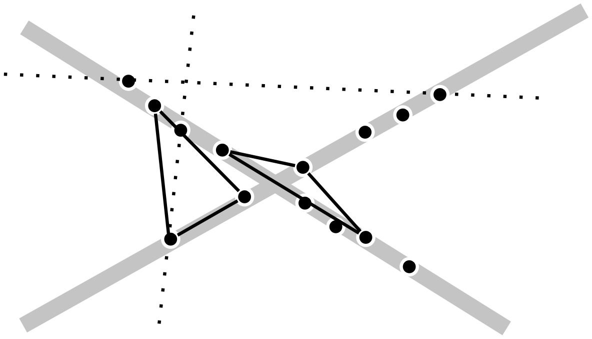


Figure 1.2 The k-lines clustering problem. Points are approximately located on the two gray lines. Notice that while pairs of points trivially define lines (shown dotted), the area of the triangle defined by a triple of points allows one to define a useful measure of similarity.

Consider the k-lines clustering problem, where the objective is to group data points in a d-dimensional vector space into k clusters where elements in each cluster are well approximated by a line. While the particular problem of clustering data into a mixture of lines can be formulated as a model based clustering problem [135]. The algorithm will and does indeed breakdown when this modeling assumption is violated, e.g., the clusters instead of being straight lines are curved with some unknown curvature, though locally they continue to look like segments of a straight line.

As every pair of data points trivially define a line, there does not exist a useful measure of similarity between pairs of points for this problem and thus the direct application of a pairwise clustering method is not possible. However, it is possible to define measures of similarity over triplets of points that indicate how close they are to being collinear. Thus, it makes sense to consider extensions of pairwise clustering methods to situations where similarity measures are defined on more than two points at a time. We refer to similarity measures over tuples of size three and more as *Higher order relations*. See Figure 1.2 for an illustration.

The subject of the first part of this dissertation is the largely neglected but fundamental problem of clustering data on the basis of triadic and higher-order relations. In particular we are interested in higher order clustering problems that occur in computer vision.

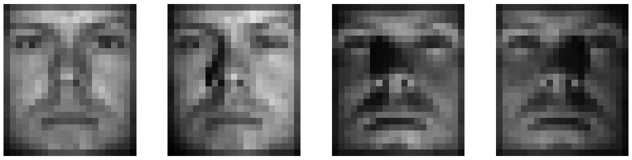


Figure 1.3 Illumination invariant clustering. Shown above are four images of the same individual under varying illumination conditions. Illumination invariant clustering refers to the task of clustering images like this into groups based on identity while ignoring variations due to illumination.

As an example of the kind of problems we are interested in, consider the problem of clustering a collection of images of different objects, each of which is imaged in the same pose, but under different lighting conditions. See Figure 1.3 for an illustration. Jacobs et al. have shown that for any two images, there exists a Lambertian surface with spatially varying albedo and a pair of light source directions that could produce the two images [76]. Hence, there is no function of a pair of images that returns zero when the images depict the same object under differing lighting yet returns a non-zero value when the images are depicting different objects. Furthermore, it is well known that the set of images of a Lambertian surface under arbitrary lighting (without shadowing) lies on a 3-D linear subspace in the image space [17]. As any three images span a 3-D subspace, one needs to consider at least four images at a time to define a measure of affinity.

As another example, consider the problem of partitioning a set of correspondences into clusters that are related by the same motion model. The usual approaches are based on

1. Greedy set covering using RANSAC [128],
2. Hough transform [14],
3. Pose clustering in the space of the model parameters [117].

There are fundamental problems with each of these approaches. RANSAC was designed for detecting a single model in the presence of noise and, as we will show, it does not scale well to the case of multiple overlapping models. Approaches based on a generalized Hough transform require a bounded finite parameterization of the model. Finding such a parametrization is not a trivial problem; even if one is available, the Hough transform for anything but the simplest problems requires a huge amount of memory. Clustering in the space of model parameters, while conceptually attractive, may not be tractable. The problem is that, to perform this clustering one needs to be able to define a measure of similarity between arbitrary pairs of models. Given that most parameter spaces are non-linear manifolds without a global metric, there may not be any easy way of doing this. In contrast, the fitting error of a set of points to a model is a natural and easily available measure of dissimilarity, without any limitations on the geometric structure of the parameter space of the model.

Given a data set, it is common practice to represent the pairwise similarity relations between its elements using a weighted graph. A number of machine learning methods for unsupervised and semi-supervised learning can then be formulated in terms of operations on this graph. For this reason, clustering algorithms are also frequently referred to as graph partitioning algorithms. In some cases like spectral clustering, the relation between the structural and the spectral properties of the graph can be exploited to construct matrix theoretic methods that are also graph theoretic. The most commonly used matrix in these methods is the Laplacian of the graph [32]. The success of graph and matrix theoretic methods in clustering have prompted researchers to extend these representations to the case of higher order relations

One direction of research has focused on the use of tensors for representing and analyzing higher order relations. Tensors are a generalization of matrices to higher dimensional arrays, and they can be analyzed using tools from multilinear algebra [52, 85, 118, 119].

Another line of work has focused on the use of hypergraphs. Hypergraphs are a generalization of ordinary graphs in which the edges are arbitrary non-empty subsets of the vertex set. Instead of having edges connecting pairs of vertices, hypergraphs have edges that connect sets of two or more vertices [20]. Weighted hypergraphs are then a natural way of representing higher order relations [4, 22, 50, 110, 111, 145, 148]. In this dissertation we focus on the use of spectral methods for partitioning hypergraphs.

1.2 Multidimensional Scaling

Consider the following problem. Given a symmetric $n \times n$ matrix $D = [d_{ij}]$ of pairwise squared Euclidean distances, find a matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ such that the following holds true.

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \quad \forall\ i, j = 1, \ldots, n \tag{1.3}$$

That is, given pairwise distances between points, assign coordinates to the points in a d dimensional Euclidean space such that these pairwise distances are preserved. This is the canonical multidimensional scaling problem. The matrix X is an embedding of D.

Multidimensional scaling (MDS) refers to the general task of assigning Euclidean coordinates to a set of objects such that a given set of dissimilarity, similarity or ordinal relations between the points are obeyed. This assignment of coordinates is also known as an embedding. The most well known of the various MDS algorithms is Classical multidimensional scaling (CMDS) [24, 53, 126, 127], where the dissimilarities between points are assumed to be actual Euclidean distances. In general, we maybe given similarities, monotonic functions of the true pairwise distances or just the relative ordering of these distances as input. In each case, the data is expected to be corrupted with noise, and obtaining an embedding which exactly satisfies the input data is usually impossible. In such cases, the problem then is to find an embedding that best satisfies the input data according to some quality measure. Further, the user might impose additional constraints on the embedding; for example a very common constraint is a bound on the dimension, in other words, a bound on the number of rows in X.

Multidimensional scaling is a widely used technique in statistics, machine learning, psychometrics, psychophysics, computer vision, computer graphics, sensor networks and even chemistry [24, 34, 36, 58, 94]. Its most common use is for visualizing a set of objects given measurements indicating their pairwise proximity or distance. In some cases, the similarity/dissimilarity measurements are indirect measurements between objects that live in some vector space, e.g., wireless sensor localization from signal strength measurements. In other cases, the objects are abstract with no immediate or direct way of realizing them except as a result of measuring their relative structure. The most common examples of this are experiments in psychometrics and psychophysics where human or animal responses define the pairwise dissimilarity relations between stimuli.

Some times, an explicit vector space representation is available for the data set, but it is has too high a dimensionality for it to be visualized easily. However, in many cases most or all of this high dimensional data actually lies on a low dimensional manifold. In such cases, it makes sense to measure the pairwise distances in this high dimensional space and then demand a new low dimensional embedding which preserves these distances. This is known as dimensionality reduction or manifold learning. When the underlying manifold is a linear subspace of the original space, the pairwise distance between points is just the ordinary Euclidean distance then the appropriate algorithm is classical multidimensional scaling. This can be shown to be equivalent to Principal Component Analysis (PCA). In the case when the manifold has more complicated structure, more sophisticated distance measurement schemes are used followed by classical multidimensional scaling [124].

In the previous section we considered the problem of clustering data. A number of clustering algorithms that operate on pairwise similarities, e.g., spectral clustering algorithms, can be broken up into two steps. In the first step multidimensional scaling is performed on an appropriately selected Gram matrix that encodes the similarities. The Gram matrix for example could either be the Combinatorial Laplacian or the Normalized Laplacian. In the second step an algorithm like k-means is used to cluster the data points in the embedding space. The expectation is that the correct choice of the Gram matrix will embed points in the same cluster close to each other and those in different clusters far apart from each other.

Multidimensional scaling algorithms can be classified into two distinct classes based on the kind of structure they pay attention to and preserve in a data set. An algorithm that takes as input pairwise distances or dissimilarities and outputs an embedding that attempts to best preserve these distances/dissimilarities is a *Metric* multidimensional scaling algorithm. On the other hand, an algorithm that only pays attention to the ordinal structure of the dissimilarities, i.e., the relative ordering of the dissimilarities between pairs of points is known as a *Non-Metric* multidimensional scaling algorithm. Non-metric multidimensional scaling finds extensive usage in psychometrics and psychophysics where the actual magnitude of the dissimilarity between pairs is not available, too difficult to measure, or not reliable.

As an example, consider an experiment in which multiple human subjects are asked to rate the dissimilarity between pairs of visual stimuli, say on a scale of 1-100. To visualize these ratings, the investigator constructs an embedding that preserves these *perceptual* distances between the various objects. A naïve solution is to either embed the average distance between pairs of objects or to find an embedding that minimizes the average distortion over all subjects. Experiments like this however are fraught with a number of problems. The first is that different subjects may use different internal scales to rate the dissimilarity between pairs of objects [83].

Further, within the span of a single experiment there are drift effects that cause the same subject to rate the same pair of stimuli differently, depending upon the order in which it was shown. Thus, the use of such rating methods using continuous scales requires careful training of subjects before the actual experiment. Even then sophisticated correction procedures may have to be applied to the data before any inferences can be made [83].

An alternative is to pay attention only to the ordinal information in the data, i.e., only consider whether the subject considered two objects to be less or more dissimilar than another pair of objects. This is better because two subjects with internal scales that are monotonic functions of each other will result in the same set of relative comparisons despite having significant differences in the actual magnitudes of dissimilarities that they report.

Non-metric multidimensional scaling is the primary subject of our study in the second part of this dissertation. We will present a novel non-metric multidimensional algorithm which takes as input a set of $\mathcal{S}$ of paired comparisons, i.e.,

$$\mathcal{S} = \left\{(i,j,k,l) | \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 < \|\mathbf{x}_k - \mathbf{x}_l\|_2^2\right\} \tag{1.4}$$

The phrase *paired comparison* refers to the fact that the comparison is between the two pairs of points (i,j) and (k,l).

We will show that this formulation of the non-metric multidimensional scaling problem subsumes the more commonly discussed case of Shepard-Kruskal scaling [24, 86, 87, 120]. Along the way we will also consider the problem of metric multidimensional scaling from the point of view of semidefinite programming and propose new algorithms for minimizing Kruskal's STRESS-1 functional and Shepard-Kruskal scaling.

We will also show how a simple variant of the solution for non-metric multidimensional scaling can be used for learning transformations of a data set such that

Euclidean distances in the transformed space fit the preferences indicated by subjects in a psychophysics test.

1.3 Contributions

My contributions in this dissertation are as follows:

1. I extend pairwise clustering methods to domains where similarity measures require more than two points and model it as a hypergraph partitioning problem.

2. A number of attempts have been made at extending spectral graph theoretic methods for analyzing hypergraphs, including applications to hypergraph paritioning. I show how all these approaches with the notable exception of two are equivalent to analyzing two closely related graphs derived from the hypergraph.

3. I propose a new hypergraph-to-graph construction, *Clique Averaging*, which we show is provably better in how it approximates the hypergraph and demonstrate its superiority over existing hypergraph partitioning algorithms.

4. I survey the literature on metric and non-metric multidimensional scaling from the point of view of modern convex optimization.

5. I propose new algorithms for performing both metric and non-metric multidimensional scaling.

6. I propose a new algorithm for learning transformations of a data set such that Euclidean distances in the transformed space fit a given set of preferences.

7. I demonstrate the performance of our novel non-metric multidimensional scaling algorithm on human subject data and use it to construct a low dimensional perceptual space for reflectance.

The rest of this dissertation is organized into three chapters. Chapter 2 considers the clustering problem with higher order relations and Chapter 3 considers the problem of non-metric multidimensional scaling. Chapter 4 discusses various other problems that I have worked on in my time in graduate schools and my contributions to their study.

2

Clustering

"All things appear and disappear because of the concurrence of causes and conditions. Nothing ever exists entirely alone; everything is in relation to everything else."

Gautam Buddha

In this chapter we consider the problem of clustering in domains with higher order relations, i.e., datasets in which similarity measures are functions of three or more points. We will consider this to be an instance of the hypergraph partitioning problem. We will survey existing methods for analyzing hypergraphs and see how they are related, propose a novel method for partitioning hypergraphs and analyze its relationship to existing methods and why its expected to perform better in theory and then validate this analysis with some experiments.

The chapter is organized as follows. Section 2.1 surveys some of the work that has been done on analyzing triadic and higher arity relations in geometry, psychometrics, economics and VLSI CAD. Section 2.2 sets up the notation that will be used in this chapter. Section 2.3 surveys some of the prominent spectral methods that have been proposed for partitioning graphs and the prominent role of the graph Laplacian in these algorithms. Section 2.4 considers the algebraic generalization of Laplacian to higher order structures and shows why it is not useful for machine learning tasks. Section 2.5 presents a survey of graph constructions and linear operators related to hypergraphs that various studies have used for analyzing the structure of hypergraphs and for unsupervised and semi-supervised learning, we then go on to show how all these constructions can be reduced to two closely graph constructions and their associated Laplacians. In section 2.8 we propose a new graph construction called *Clique Averaging* and analyze its properties.

Section 2.9 applies the various clustering algorithms to two problems in computer vision. Finally, Section 2.10 concludes with a discussion and directions for future work.

2.1 Related Work

The study of distances defined over sets of size greater than two is not new. The literature on n-metrics is devoted to constructing and analyzing distance measures defined over $(n+1)$-tuples. In this notation the usual pairwise metrics are referred to as 1-metrics. The primary focus of this literature is the study of topological and geometrical properties of these generalized measures [38].

While the work on n-metrics is theoretical, a more practical line of work has emerged in the psychometrics community. Starting with the work of Hayashi, who proposed the area of a triangle as the triadic distance between its vertices [59], a number of researchers have developed generalizations of multidimensional scaling to the case of triadic data. In one of the earliest such works, Carroll & Chang developed an algorithm for n-adic MDS using a generalization of the singular value decomposition to the case of n-dimensional matrices [27]. Subsequently, Cox et al. have proposed an MDS algorithm based on a combination of gradient descent and Isotonic Regression [35]. Axiomatic theories of triadic distances have been developed by Joly & LeCalvé and Heiser & Bennani [61, 81]. Both sets of authors use this axiomatic development as the basis of new MDS algorithms. Joly & LeCalvé propose the use of a least squares procedure for recovering pairwise distances and running classical MDS on them, and, Heiser & Bennani introduce provably convergent algorithms based on iterative majorization that directly solve for the Euclidean embedding.

Another line of work that deals with affinities defined on arbitrary subsets of the data is the work on hierarchical clustering based on the theory of partially ordered sets. The first extensive study of generalized affinities in a set theoretic context was done by Hubert, he referred to these n-adic affinities as *n-Clustering Functions* [72]. However work on constructing algorithms for clustering over n-adic relationships can be traced back to the early work on hierarchical clustering by Johnson, Jardine & Sibson and Hubert [70, 71, 79, 80]. This work was subsequently generalized in the work of Hubert, Janowitz, Herden, Bandelt and others [15, 62, 77, 78]. The paper by Herden provides a brief survey of this line of work.

The most extensive and large scale use of hypergraph partitioning algorithms, however, occurs in the field of VLSI design and synthesis. A typical application involves the partitioning of large circuits into k equally sized parts in a manner that minimizes the connectivity between the parts. The circuit elements are the vertices of the hypergraph and the *nets* that connect these circuit elements are the hyperedges [8]. The leading tools for partitioning these hypergraphs are based on two phase multi-level approaches [82]. In the first phase, they construct a hierarchy of hypergraphs by incrementally collapsing the hyperedges of the original hypergraph according to some measure of homogeneity. In the second phase, starting from a partitioning of the hypergraph at the coarsest level, the algorithm works its way down the hierarchy and at each stage the partitioning at the level above serves as an initialization for a vertex swap based heuristic that refines the partitioning greedily [44,84]. The development of these tools is almost entirely heuristic and very little theoretical work exists that analyzes their performance beyond empirical benchmarks.

The set of tools available for partitioning graphs are much better developed than those for hypergraphs. A case in point is the development of algorithms for solving the max-flow min-cut problem on hypergraphs. While efficient algorithms for the case of graphs have been available for sometime now [51], it is only recently that efficient algorithms that operate directly on hypergraphs have been developed [105]. Thus it makes sense to consider methods that construct a graph that approximates the hypergraph and partition it; this partition in turn induces a vertex partitioning on the original hypergraph. The two most commonly used graph approximations are *Clique Expansion* and *Star Expansion*. Clique Expansion, as the name suggests, expands each hyperedge into a clique. Star expansion introduces a dummy vertex for each hyperedge and connects each vertex in the hyperedge to it [68]. As can be expected, the weights on the edges of the clique and the star determine the cut properties of the approximating graph [54,75].

Ihler et al. have shown that for hypergraphs with edges of size greater than three there does not exist a clique expansion that preserves the min-cut [75]. Hadley gave a closed form expression for the least squares solution to the problem of best min-cut preserving clique expansion [54]. However, as the solution involves Sterling numbers it is not numerically feasible to use this solution for even moderate sized graphs. In most applications of clique-expansion a heuristic scaling is used when assigning values to each graph edge [9]. We will have more to say on the subject of clique and star expansion in sections 2.5 and 2.6.

2.2 Notation

Let $G(V, E)$ denote a hypergraph with vertex set V and edge set E. The edges are arbitrary subsets of V with weight $w(e)$ associated with edge e. The degree $d(v)$ of a vertex is $d(v) = \sum_{e \in E | v \in e} w(e)$. The degree of an edge e is denoted by $\delta(e) = |e|$. For k-uniform hypergraphs, the degrees of each edge are the same, $\delta(e) = k$. In particular, for the case of ordinary graphs or "2-graphs," $\delta(e) = 2$. The vertex-edge incidence matrix H is $|V| \times |E|$ where the entry $h(v, e)$ is 1 if $v \in e$ and 0 otherwise. By these definitions, we have:

$$d(v) = \sum_{e \in E} w(e) h(v, e) \quad \text{and} \quad \delta(e) = \sum_{v \in V} h(v, e) \tag{2.1}$$

D_e and D_v are the diagonal matrices consisting of edge and vertex degrees, respectively. W is the diagonal matrix of edge weights, $w(\cdot)$. A number of different symbols have been used in the literature to denote the Laplacian of graph. We follow the convention in [32] and use L for the combinatorial Laplacian and $\mathcal{L}$ for the normalized Laplacian. L is also known as the unnormalized Laplacian of a graph and is usually written as

$$L = D_v - S \tag{2.2}$$

where S is the $|V| \times |V|$ adjacency matrix with entry (u, v) equal to the weight of the edge (u, v) if they are connected, 0 otherwise. An important variant is the normalized Laplacian,

$$\mathcal{L} = I - D_v^{-1/2} S D_v^{-1/2} \tag{2.3}$$

For future reference it is useful to rewrite the above expressions in terms of the vertex-edge incidence relation

$$S = HWH^\top - D_v \tag{2.4}$$

$$L = 2D_v - HWH^\top \tag{2.5}$$

$$\mathcal{L} = I - \frac{1}{2} D_v^{-1/2} HWH^\top D_v^{-1/2} \tag{2.6}$$

2.3 Spectral Clustering

Spectral clustering methods have emerged as one of the most successful class of methods for partitioning graphs. The word spectral refers to the fact that all these methods depend in a key way on the spectral (eigenvector/eigenvalue) structure of various linear operators associated with the graphs. In this section we will take a brief look at some of the more popular spectral clustering algorithms in use today and the role of the graph Laplacian in these algorithms.

Perhaps the simplest spectral clustering algorithm is the *Affinity Factorization Algorithm* proposed by Perona & Freeman [104]. They consider the best rank one approximation to the affinity matrix $A = I + S$, where I is the identity matrix, i.e., the problem:

$$\min_{\mathbf{p}} \|A - \mathbf{p}\mathbf{p}^\top\|_F. \tag{2.7}$$

It can be shown that if $U\Sigma U^\top = A$ is the eigenvalue decomposition of A, then the best rank k approximation to it is the matrix $U_k \Sigma_k U_k^\top$ where, U_k is the matrix of the k eigenvectors associated with the k largest eigenvalues of A. Σ_k is the diagonal matrix of the corresponding eigenvalues [42, 67]. In particular the best rank one approximation is $\mathbf{p} = u_1\sqrt{\lambda_1}$. Further, since A is a non-negative matrix, as a consequence of the Perron-Frobenius theorem, $\mathbf{p}$ is a non-negative vector.

Perona & Freeman consider the problem of partitioning the pixels of an image into background and foreground pixels. The vector $\mathbf{p}$ had as many entries as the number of pixels in the image and the value p_i corresponding to the pixel i was considered to be indicative of its saliency. Once the vector $\mathbf{p}$ was estimated, the clustering was obtained by thresholing the entries of $\mathbf{p}$ at some appropriately chosen threshold.

Unfortunately this approach does not scale to multiple clusters as subsequent eigenvectors of A do not share the nice non-negativity properties of $\mathbf{p}$, and the only option is to consider a recursive process in which the algorithm is applied recursively to partitions obtained at an earlier step till some termination criterion is met. Another approach along the same lines is the one proposed by Sarkar & Boyer [113], in which they consider the eigenvectors of S to cluster regions of the image into distinct clusters. Note that both S and $A = S + I$ have the same eigenvectors, the only difference is in the eigenvalues which differ by 1.

The use of the eigenstructure of the combinatorial Laplacian

$$L = D_v - S, \tag{2.8}$$

for graph parititoning was made popular by the work of Pothen et al. [106] though it had been suggested much earlier by a number of authors [30, 39, 45]. Pothen et al.

considered the problem of clustering the rows of a sparse matrix for the purposes of distributing a computation involving this matrix across a number of processors in a computing system. In [56], Hall showed that the second smallest eigenvector of L, i.e. the eigenvector associated with the second smallest eigenvalue solves the one dimensional *quadratic placement problem.* The quadratic placement problem is

$$\arg\min_{\mathbf{x}} \quad \sum_{i,j}^{n} (x_i - x_j)^2 S_{ij} \tag{2.9}$$

$$\text{subject to} \quad \mathbf{x}^\top \mathbf{x} = 1. \tag{2.10}$$

Hall then used the second and third smallest eigenvectors to perform two dimensional clustering.

In VLSI CAD and other domains, it is sometimes useful to have partitions which are *balanced.* The size of a partition can be measured in different ways, each of which leads to a measure of balance and cluster quality. One simple way is to count the number of vertices in each partition. This leads to the notion of Ratio Cut [55]. For a bi-partitioning of a vertex set $V = A_1 \cup A_2$. The ratio cut is defined as

$$C(A_1, A_2, S) = \frac{\sum_{i \in A_1, j \in A_2} S_{ij}}{|A_1|} + \frac{\sum_{i \in A_1, j \in A_2} S_{ij}}{|A_2|} \tag{2.11}$$

In [55] it was shown that the second smallest eigenvalue of L is a lower bound on the minimum ratio cut of the graph G. The final assignment of vertices to cluster is done via various heuristics involving either thresholding of the eigenvector, or greedy refinement using Kernighan-Lin [84].

A variant of the combinatorial Laplacian is the Normalized Laplacian introduced by Chung [32]:

$$\mathcal{L} = I - D_v^{-1/2} S D_v^{-1/2} \tag{2.12}$$

Since then a number of researchers have considered the use of the eigenvectors of the Normalized Laplacian in clustering data [12, 98, 101, 121, 142, 144].

Let $A = (A_r)_{r \in \{1,\ldots,k\}}$ be a partitioning of V. Then, consider the following cost function

$$C(A,S) = \sum_{r=1}^{k} \frac{\sum_{i \in A_r, j \in V \setminus A_r} S_{ij}}{\sum_{i \in A_r, j \in V} S_{ij}} \tag{2.13}$$

$$= \sum_{r=1}^{k} \frac{\sum_{i \in A_r, j \in V \setminus A_r} S_{ij}}{\sum_{i \in A_r} d(i)} \tag{2.14}$$

The above cost function is a sum of k terms, one for each cluster. Each term is the ratio of the sum of the weights of the edges that have one vertex in that cluster and the other in some other cluster, to the sum of the weights of all edges with at least one vertex in that cluster. More intuitively, it is the ratio of cut or the sum of edge weights that are violated as a result of the particular partitioning to the total association or connectivity of that cluster to the rest of the graph. This cost function is a k-way generalization of the ratio-cut cost function. In the ratio-cut cost function, the size of each cluster was its cardinality, i.e. the number of vertices contained in it. This can be misleading when there are a lot of vertices with small edge weights connecting them to the rest of the graph. A better measure of the size of the cluster is the degree sum of the vertices contained in it. Equation 2.14 formalizes this.

Now let $E = [\mathbf{e}_1, \dots, \mathbf{e}_k]$ be a matrix of column vectors, where the vector $\mathbf{e}_j$ is the $\{0,1\}$ indicator vector for the j^{th} cluster, i.e $e_{ij} = 1$ if the i^{th} point is in cluster j and 0 otherwise. Then the above cost function can be re-written as

$$C(E,S) = \sum_{r=1}^{k} \frac{\mathbf{e}_r^\top (D_v - S)\mathbf{e}_r}{\mathbf{e}_r^\top D_v \mathbf{e}_r} \tag{2.15}$$

Then, Bach & Jordan show that the above objective function can be re-stated as $k - \operatorname{tr} Y^\top D_v^{-1/2} S D_v^{-1/2} Y$ for any matrix Y such that the columns of $D_v^{-1/2} Y$ are piecewise constant with respect to the cluster and $Y^\top Y = I$. Relaxing the piecewise constancy constraint they showed.

Theorem 2.1. *The maximum of* $\operatorname{tr} Y^\top D_v^{-1/2} S D_v^{-1/2} Y$ *over matrix* $Y \in \mathbb{R}^{n \times k}$ *such that* $Y^\top Y = I$ *is the sum of the* k *largest eigenvalues of* $D_v^{-1/2} S D_v^{-1/2}$*. It is attained at all* Y *of the form* $Y = UB$ *where,* $U \in \mathbb{R}^{n \times k}$ *is any orthonormal basis of the* k*-th principal subspace of* $D_v^{-1/2} S D_v^{-1/2}$ *and* B *is an arbitrary rotation matrix in* $\mathbb{R}^{k \times k}$*.*

The above theorem is a generalization of the result in [121] where the authors showed that a similar relaxation for bi-partitioning leads to the second smallest eigenvector of $\mathcal{L}$. The actual assignment of the vertices to clusters is obtained by performing weighted k-means clustering on the k-dimensional embedding obtained using k-th principal subspace of $D_v^{-1/2} S D_v^{-1/2}$.

For large problems, the algorithms as described above are not very practical due to their computational requirements. In [48] the authors introduced the use of the Nÿstrom approximation to speed this computation. It exploits the fact that if there are a small number of clusters in the data, then it is possible to approximate the eigenvectors of the Laplacian by first calculating the eigenvector decomposition of a much smaller matrix obtained by sub-sampling rows and columns and then extrapolating them to the full sized matrix. The reason one is able to do this is because in many cases when the data is well separated in feature space, rows of the Laplacian corresponding to points in the same cluster are very similar and the Laplacian itself has low rank. Spectral methods for clustering in machine vision and learning have been discussed widely, the papers by Weiss [137] and Higham et al. [63] have further discussion on these issues.

Graph Laplacians have found uses beyond clustering too. In particular they have been used extensively for solving semi-supervised learning problems [147]. Semi-supervised learning refers to the problem of learning a classifier from a combination of label and unlabeled data as opposed to supervised learning where all the training data has class labels associated with it. The use of the graph Laplacian is based on the observation that the graph Laplacian is the discrete analog of the Laplace-Beltrami operator on compact Riemannian manifolds [18, 32, 112] and like it can be used as a discrete regularization operator for functions defined on a graph.

In [146], the authors develop a discrete calculus on 2-graphs by treating them as discrete analogs of compact Riemannian manifolds. As one of the consequences of this development they argue that, in analogy to the continuous case, the graph Laplacian be defined as an operator $L : \mathcal{H}(V) \to \mathcal{H}(V)$,

$$Lf := \frac{1}{2} \operatorname{div}(\nabla f) \tag{2.16}$$

where $\mathcal{H}(V)$ is the Hilbert space of all vertex functions on the graph G. More explicitly,

$$(Lf)(v) = \frac{1}{\sqrt{g(v)}} \sum_{u \sim v} \left(\frac{w(u,v)}{\sqrt{g(v)}} f(v) - \frac{w(u,v)}{\sqrt{g(u)}} f(u) \right) \tag{2.17}$$

Here g is some positive vertex function in $\mathcal{H}(V)$. For the particular choice of $g(v) = 1$, the above is the same as the combinatorial Laplacian and for $g(v) = d(v)$ it is equivalent to the normalized Laplacian.

Zhou et al. also argue that there exists a family of regularization operators on the 2-graphs, the Laplacian being one of them, that can be used for transduction, i.e., given a partial labeling of the graph vertices y, use the geometric structure of the graph to induce a labeling f on the unlabeled vertices. The vertex label $y(v)$ is $+1, -1$ for positive and negative valued examples, respectively, and 0 if no information is available about the label. They consider the regularized least squares problem

$$\arg\min_{f} \left(\langle f, Lf \rangle + \mu \|f - y\|_2^2 \right) \tag{2.18}$$

While the discrete version of the problem where $f(v) \in \{+1, -1\}$ is a hard combinatorial problem, relaxing the range of f to the real line $\mathbb{R}$ results in a simple linear least squares problem, solved as $f = \mu(\mu I + L)^{-1}y$. A similar formulation is considered by [18].

In the absence of any labels, the problem reduces to that of clustering, and the eigenmodes of the Laplacian are used to label the vertices. This results in the familiar spectral clustering algorithms discussed above [101, 121].

More generally, in [122], the authors prove that just as the continuous Laplacian is the unique linear second order self adjoint operator invariant under the action of rotation operators, the same is true for the Laplacian and the unnormalized Laplacian with the group of rotations replaced by group of permutations.

A number of successful regularizers in the continuous domain can be written as $\langle f, r(L)f \rangle$ where L is the continuous Laplacian, f is the model and r is non-decreasing scalar function that operates on the spectrum of Δ. Smola and Kondor show that the same can be shown for a variety of regularization operators on graphs.

2.4 Higher Order Laplacians

In light of the previous section, it is interesting to consider generalizations of the Laplacian to higher order structures. We now present a brief look at the algebro-geometric view of the Laplacian, and how it leads to the generalization of the combinatorial Laplacian for hypergraphs. For simplicity of exposition, we will consider the unweighted case. For a more formal presentation of the material in this section, we refer the reader to [31, 32, 47, 99].

Let us assume that a graph represents points in some abstract space with the edges representing lines connecting these points and the weights on the edge having an inverse relation to the length of the line. The Laplacian then measures how smoothly a function defined on these points (vertices) changes with respect to their relative arrangement. As we saw earlier, the quadratic form $f^\top Lf$ does this for the vertex function f. This view of a graph and its Laplacian can be generalized to hypergraphs. A hypergraph represents points in some abstract space where each hyperedge corresponds to a simplex in that space with the vertices of the hyperedge as its corners. The weight on the hyperedge is inversely related to the *size* of the simplex. Now we are not restricted to define functions on just vertices, we can define functions on sets of vertices, corresponding to lines, triangles, etc. Algebraic topologists refer to these functions as p-chains, where p is size of the simplices on which they are defined. Thus vertex functions are 0-chains, edge functions are 1-chains, and so on. In each case one can ask the question, how does one measure the variation in these functions with respect to the geometry of the hypergraph or its corresponding simplex?

Let us take a second look at the graph Laplacian. As the graph Laplacian is a positive semidefinite operator, it can be written as

$$L = BB^\top \tag{2.19}$$

Here, B is a $|V| \times |E|$ matrix such that (u, v)th column contains $+1$ and -1 in rows u and v, respectively. The exact ordering does not matter. B is called the boundary operator ∂_1 that maps on 1-chains (edges) to 0-chains and $B^\top$ is the co-boundary operator that maps 0-chains to 1-chains. Note that B is different from H; although H is also a vertex-edge incidence matrix, all of its entries are non-negative. We can rewrite

$$f^\top Lf = f^\top BB^\top f = \|B^\top f\|_2^2. \tag{2.20}$$

Thus $f^\top Lf$ is the squared norm of a vector of size $|E|$, whose entries are the change in the vertex function or the 0-chain along an edge. This is a particular case of the general definition of the p^{th} Laplacian operator on p-chains, given by

$$L_p = \partial_{p+1}\partial_{p+1}^\top + \partial_p^\top \partial_p \tag{2.21}$$

Symbolically, this is exactly the same as the Laplace operator on p-forms on a Riemannian manifold [112]. For the case of hypergraphs or simplicial complexes, we interpret this as the operator that measures variations on functions defined on p-sized subsets of the vertex set (p-chains). It does so by considering the change in the chain with respect to simplices of size $p + 1$ and $p - 1$. For the case of the ordinary graph, we only consider the first term in the above expression since vertex functions are 0-chains, and there are no -1 sized simplices. It is however possible to consider 1-chains or functions defined on edges of the graphs and measure their variation using the edge Laplacian, given by $L_1 = B^\top B$. In light of this, the usual Laplacian on the graph is the L_0 or vertex Laplacian. In [31] the Laplacian for the particular case of the k-uniform hypergraph is presented. A more elaborate discussion of the construction of various kinds of Laplacians on simplicial complexes and their uses is described in [47].

Unfortunately, while geometrically and algebraically these constructions extend the graph Laplacian to hypergraphs, it is not clear how one can use them in machine learning. The fundamental object we are interested in is a vertex function or 0-chain, thus the linear operator we are looking for should operate on 0-chains. Notice, however, that a p^{th} order Laplacian only considers p-chains, and the structure of the Laplacian depends on the incidence relations between $p-1$, p and $p+1$ simplices. To operate on vertex functions, one needs a vertex Laplacian, which unfortunately only considers the incidence of 0-chains with 1-chains. Thus the vertex Laplacian for a k-uniform hypergraph will not consider any hyperedges, rendering it useless for the purposes of studying vertex functions. Indeed the Laplacian on a 3-uniform graph operates on 2-chains, functions defined on all pairs of vertices [31].

2.5 Hypergraph Learning Algorithms

A number of existing methods for learning from a hypergraph representation of data first construct a graph representation using the structure of the initial hypergraph. Then, they project the data onto the eigenvectors of the combinatorial or normalized graph Laplacian. Other methods define a hypergraph “Laplacian” using analogies from the graph Laplacian. These methods show that the eigenvectors of their Laplacians are useful for learning, and that there is a relationship between their hypergraph Laplacians and the structure of the hypergraph. In this section, we review these methods. In the next section, we compare these methods analytically.

2.5.1 Clique Expansion

The clique expansion algorithm constructs a graph $G^x(V, E^x \subseteq V^2)$ from the original hypergraph $G(V, E)$ by replacing each hyperedge $e = (u_1, ..., u_{\delta(e)}) \in E$ with an edge for each pair of vertices in the hyperedge [148]: $E^x = \{(u, v) : u, v \in e, e \in E\}$.

Note that the vertices in hyperedge e form a clique in the graph G^x. The edge weight $w^x(u,v)$ minimizes the difference between the weight of the graph edge and the weight of each hyperedge e that contains both u and v:

$$w^x(u,v) = \operatorname*{arg\,min}_{w^x(u,v)} \sum_{e \in E : u,v \in e} (w^x(u,v) - w(e))^2 \tag{2.22}$$

Thus, clique expansion uses the discriminative model that every edge in the clique of G^x associated with hyperedge e has weight $w(e)$. The minimizer of this criterion is simply

$$w^x(u,v) = \mu \sum_{e \in E : u,v \in e} w(e) = \mu \sum_{e} h(u,e)h(v,e)w(e). \tag{2.23}$$

Here μ is a fixed scalar. The combinatorial or normalized Laplacian of the constructed graph G^x is then used to partition the vertices.

2.5.2 Star Expansion

The star expansion algorithm constructs a graph $G^*(V^*, E^*)$ from hypergraph $G(V,E)$ by introducing a new vertex for every hyperedge $e \in E$, thus $V^* = V \cup E$ [148]. It connects the new graph vertex e to each vertex in the hyperedge to it, i.e. $E^* = \{(u,e) : u \in e, e \in E\}$.

Note that each hyperedge in E corresponds to a star in the graph G^* and that G^* is a bi-partite graph. Star expansion assigns the scaled hyperedge weight to each corresponding graph edge:

$$w^*(u,e) = w(e)/\delta(e) \tag{2.24}$$

The combinatorial or normalized Laplacian of the constructed graph G^x is then used to partition the vertices.

2.5.3 Bolla's Laplacian

Bolla [22] defines a Laplacian for an unweighted hypergraph in terms of the diagonal vertex degree matrix D_v, the diagonal edge degree matrix D_e, and the incidence matrix H, defined in Section 2.2.

$$L^o := D_v - H D_e^{-1} H^\top. \tag{2.25}$$

The eigenvectors of Bolla's Laplacian L^o define the "best" Euclidean embedding of the hypergraph. Here, the cost for embedding $\phi : V \to \mathbb{R}^k$ of the hypergraph is the total squared distance between pairs of embedded vertices in the same hyperedge

$$\sum_{u,v \in V} \sum_{e \in E : u,v \in e} \|\phi(u) - \phi(v)\|^2 \tag{2.26}$$

Bolla shows a relationship between the spectral properties of L^o and the minimum cut of the hypergraph.

2.5.4 Rodriguez's Laplacian

Rodríguez [110, 111] constructs a weighted graph $G^r(V, E^r = E^x)$ from an unweighted hypergraph $G(V, E)$. Like clique expansion, each hyperedge is replaced by a clique in the graph G^r. The weight $w^r(u, v)$ of an edge is set to the number of edges containing both u and v:

$$w^r(u, v) = |\{e \in E : u, v \in e\}| \tag{2.27}$$

Rodríguez expresses the graph Laplacian applied to G^r in terms of the hypergraph structure:

$$L^r(G^r) = D_v^r - HH^\top \tag{2.28}$$

where D_v^r is the vertex degree matrix of the graph G^r. Like Bolla, Rodriguez shows a relationship between the spectral properties of L^r and the cost of minimum partitions of the hypergraph.

2.5.5 Zhou's Normalized Laplacian

Zhou et al. [145] generalize their earlier work on regularization on graphs and consider the following regularization on a vertex function f.

$$\langle f, L^z f \rangle = \frac{1}{2} \sum_{e \in E} \frac{1}{\delta(e)} \sum_{\{u,v\} \subseteq e} w(e) \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2$$

Note that this regularization term is small if vertices with high affinities have the same label. They show that the operator L^z can then be written as

$$L^z = I - D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2} \tag{2.29}$$

In addition, Zhou et al. define a hypergraph normalized cut criterion for a k-partition of the vertices $P_k = \{V_1, ..., V_k\}$:

$$\mathrm{NCut}(P_k) := \sum_{i=1}^{k} \frac{\sum_{e \in E} w(e)|e \cap V_i||e \cap V_i^c|}{\delta(e) \sum_{v \in V_i} d(v)}. \tag{2.30}$$

This criterion is analogous to the normalized cut criterion for graphs. They then show that if minimizing the normalized cut is relaxed to a real-vealed optimization problem, the second smallest eigenvector of L^z is the optimal classification function f. Finally, they also draw a parallel between their hypergraph normalized cut criterion and random walks over the hypergraph.

2.5.6 Gibson's Dynamical System

In [50] the authors have proposed a dynamical system to cluster categorical data that can be represented using a hypergraph. They consider the following iterative process.

1. $s_{ij}^{n+1} = \sum_{e:i \in e} \sum_{k \neq i \in e} w_e s_{kj}^n$
2. Orthonormalize the vectors s_j^n.

They prove that the above iteration is convergent. We observe that

$$\begin{aligned} s_{ij}^{n+1} &= \sum_e h(i,e) \left(\sum_k h(k,e) w_e s_{kj}^n - w_e s_{ij}^n \right) \\ s_j^{n+1} &= (HWH^\top - D_v) s_j^n \end{aligned} \tag{2.31}$$

Thus, the iterative procedure described above is the power method for calculating the eigenvectors of the adjacency matrix $S = D_v - HWH^\top$.

2.5.7 Li's Adjacency Matrix

Li et al. [91] formally define properties of a regular, unweighted hypergraph $G(V, E)$ in terms of the star expansion of the hypergraph. In particular, they define the $|V| \times |V|$ adjacency matrix of the hypergraph, $HH^\top$. They show a relationship between the spectral properties of the adjacency matrix of the hypergraph $HH^\top$ and the structure of the hypergraph.

2.6 Comparing Hypergraph Learning Algorithms

In this section, we compare the algorithms for learning from a hypergraph representation of data described in Section 2.5. In Section 2.6.1, we compute the normalized Laplacian for the star expansion graph. In Section 2.6.2, we compute the combinatorial and normalized Laplacian of the clique expansion graph. In Section 2.6.3, we show that these Laplacians are nearly equivalent to each other. Finally, in Section 2.7, we show that the various hypergraph Laplacians can be written as the graph Laplacian of the clique expansion graph.

We begin by stating a simple lemma. The proof is trivial.

Lemma 2.2. *Let,*

$$B = \begin{bmatrix} I & -A \\ -A^\top & I \end{bmatrix}$$

be a block matrix with A rectangular. Consider the eigenvalue problem

$$\begin{bmatrix} I & -A \\ -A^\top & I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$

then the following relation holds

$$AA^\top \mathbf{x} = (1-\lambda)^2 \mathbf{x}$$

2.6.1 Star Graph Laplacian

Given a hypergraph $G(V, E)$, consider the star graph $G^*(V^*, E^*)$, i.e. $V^* = V \cup E, \quad E^* = \{(u, e) : u \in e, e \in E\}$. Notice that this is a bipartite graph, with vertices corresponding to E on one side and vertices corresponding to V on the other, since there are no edges from V to V or from E to E. Let us also assume that the vertex set V^* has been ordered such that all elements of V come before elements of E.

Let $w^* : V \times E \to \mathbb{R}^+$ be the (as yet unspecified) graph edge weight function. In addition, let S^* be the $(|V| + |E|) \times (|V| + |E|)$ affinity matrix. We can write the affinity matrix in terms of the hypergraph structure and the weight function w^* as

$$S^* = \begin{bmatrix} 0_{|V|} & HW^* \\ W^*H^\top & 0_{|E|} \end{bmatrix} \tag{2.32}$$

The degrees of vertices in G^* are then

$$d^*(u) = \sum_{e \in E} h(u,e) w^*(u,e) \quad u \in V \tag{2.33}$$

$$d^*(e) = \sum_{u \in V} h(u,e) w^*(u,e) \quad e \in E \tag{2.34}$$

The normalized Laplacian of this graph can now be written in the form

$$\mathcal{L}^* = \begin{bmatrix} I & -A \\ -A^\top & I \end{bmatrix}. \tag{2.35}$$

Here, A is the $|V| \times |E|$ matrix

$$A = D_v^{*-1/2} H W D_v^{*-1/2}$$

with entry (u, e)

$$A_{ue} = \frac{h(u,e) w^*(u,e)}{\sqrt{d^*(u)}\sqrt{d^*(e)}}. \tag{2.36}$$

Any $|V| + |E|$ eigenvector $\mathbf{x}^\top = [\mathbf{x}_v^\top,\ \mathbf{x}_e^\top]$ of $\mathcal{L}^*$ satisfies $\mathcal{L}^* \mathbf{x} = \lambda \mathbf{x}$. Then by Lemma 2.2, we know that

$$AA^\top \mathbf{x}_v = (\lambda - 1)^2 \mathbf{x}_v. \tag{2.37}$$

Thus, the $|V|$ elements of the eigenvectors of the normalized Laplacian $\mathcal{L}^*$ corresponding to vertices $V \subseteq V^*$ are the eigenvectors of the $|V| \times |V|$ matrix $AA^\top$. Element (u, v) of $AA^\top$ is

$$[AA^\top]_{uv} = \sum_{e \in E} \frac{h(u,e) h(v,e) w^*(u,e) w^*(v,e)}{\sqrt{d^*(u)} d^*(e) \sqrt{d^*(v)}}. \tag{2.38}$$

For the standard star expansion weighting function, $w^*(u,e) = w(e)/\delta(e)$, so the vertex degrees are

$$d^*(u) = \sum_{e \in E} h(u,e) w(e)/\delta(e) \quad u \in V \tag{2.39}$$

$$d^*(e) = \sum_{u \in e} w(e)/\delta(e) = w(e) \quad e \in E \tag{2.40}$$

Thus, we can write

$$[AA^\top]^*_{uv} = \sum_{e \in E} \frac{h(u,e) h(v,e) w(e)/\delta(e)^2}{\sqrt{d^*(u)}\sqrt{d^*(v)}} \tag{2.41}$$

2.6.2 Clique Graph Laplacian

Given a hypergraph $G(V, E)$, consider the graph $G^c(V, E^c = E^x)$ with the same structure as the clique expansion graph, i.e. $E^c = \{(u, v) : u, v \in e, e \in E$.

Let $w^c : V \times E \to \mathbb{R}^+$ be the (as yet unspecified) hypergraph edge weight. We can write the normalized Laplacian of G^c in terms of the hypergraph structure and the weight function w^c as $\mathcal{L}^c := I - C$. If there is no hyperedge $e \in E$ such that $u, v \in E$ then $C_{uv} = 0$. Otherwise,

$$[C]_{uv} = \frac{w^c(u, v)}{\sqrt{d^c(u)}\sqrt{d^c(v)}} \tag{2.42}$$

where

$$d^c(u) = \sum_{e \in E} h(u, e) \sum_{v \in e \setminus \{u\}} w^c(u, v) \tag{2.43}$$

is the vertex degree. For the standard clique expansion construction,

$$w^c(u, v) = w^x(u, v) = \sum_{e \in E : u, v \in e} w(e). \tag{2.44}$$

so the vertex degrees are

$$d^c(u) = d^x(u) = \sum_{e \in E} h(u, e)(\delta(e) - 1)w(e) \tag{2.45}$$

2.6.3 Unifying Star and Clique Expansion

To show the relationship between star and clique expansion, consider the star expansion graph $G^*_c(V^*, E^*)$ with weighting function

$$w^*_c(u, e) := w(e)(\delta(e) - 1) \tag{2.46}$$

Note that this is $(\delta(e) - 1)\delta(e)$ times the standard star expansion weighting function $w^*(u, e)$ (Eq. (2.24)). Plugging this value into Equations (2.33) and (2.34), we get that the degrees of vertices in G^* are

$$d^*_c(u) = \sum_{e \in E} h(u, e)w(e)(\delta(e) - 1) = d^x(u) \tag{2.47}$$

$$d^*_c(e) = w(e)\delta(e)(\delta(e) - 1) \tag{2.48}$$

where $d^x(u)$ is the vertex degree for the standard clique expansion graph G^*_x. Thus,

$$[A^*_c A^{*\top}_c]_{uv} = \sum_{e \in E} \left[\frac{\delta(e)-1}{\delta(e)}\right] \frac{h(u,e)h(v,e)w(e)}{\sqrt{d^x(u)}\sqrt{d^x(v)}} \tag{2.49}$$

Similarly, suppose we choose the clique expansion weighting function

$$w^c_*(u,v) := \frac{\sum_{e \in E} h(u,e)h(v,e)w(e)}{\delta(e)(1-\delta(e))} \tag{2.50}$$

Then we can show that the vertex degree is

$$d^c_*(u) = \sum_{e \in E} h(u,e)w(e)/\delta(e) = d^*(u) \tag{2.51}$$

where $d^*(u)$ is the vertex degree function for standard star expansion. We can then write

$$[C_*]_{uv} = \sum_{e \in E} \frac{1}{\delta(e)\delta(e-1)} \frac{h(u,e)h(v,e)w(e)}{\sqrt{d^*(u)}\sqrt{d^*(v)}} \tag{2.52}$$

A commonly occuring case is the k-uniform hypergraph. In this case, each hyperedge has exactly the same number of vertices, i.e. $\delta(e) = k$. Then it is easy to see that the bipartite graph matrix $A^*_c A^{*\top}_c$ is a constant scalar times the clique expansion matrix C. Thus, the eigenvectors of the normalized Laplacian for the bipartite graph G^*_c are exactly the eigenvectors of the normalized Laplacian for the standard clique expansion graph G^x. Similarly, the clique matrix C_* is a constant scalars times the standard star expansion matrix $[AA^\top]^*$. Thus, the eigenvectors of the normalized Laplacian for the clique graph G^c_* are exactly the eigenvectors of the normalized Laplacian for standard star expansion. This is a surprising result, since the two graphs are completely different in the number of vertices and the connectivity between these vertices.

For non-uniform hypergraphs (i.e. the hyperedge cardinality varies), the bipartite graph matrix $A^*_c A^{*\top}_c$ while not the same is close to the clique expansion matrix C^c_*. Each term in the sum in Equation (2.49) has an additional factor $(\delta(e)-1)/\delta(e)$, giving slightly higher weight to hyperedges with a higher degree. This difference however is not large, especially with higher cardinalities. As the bipartite graph matrix $A_c A_c^\top$ is approximately the clique expansion matrix, we conclude that their eigenvectors are similar. A similar relation holds for the clique graph G^c_* and the standard star expansion where the clique graph gives lower weight to larger edges. These observations can be reversed to characterize the behavior of the standard clique expansion and star expansion construction, and we conclude that the clique expansion gives more weight to evidence from larger edges than star expansion.

There is no clear reason why one should give more weight to smaller hyperedges versus larger edges or vice versa. The exact choice will depend on the properties of the affinity function used.

2.7 Unifying Hypergraph Laplacians

In this section we take a second look at the various constructions in Section 2.5 and show that they all correspond to either clique or star expansion of the original hypergraph with the appropriate weighting function.

For an unweighted hypergraph, Bolla's Laplacian L^o corresponds to the unnormalized Laplacian of the associated clique expansion with the weight matrix of the hypergraph the inverse of the degree matrix D_e:

$$W^o = HD_e^{-1}H^\top \tag{2.53}$$

The row sums of this matrix are given by

$$d^o(u) = \sum_v \sum_{e \in E} h(u,e) \frac{1}{\delta(e)} h(v,e) \ = \sum_{e \in E} h(u,e) \tag{2.54}$$

which as a diagonal matrix is exactly the vertex degree matrix D_v for an unweighted hypergraph, giving us the unnormalized Laplacian

$$L^o = D_v - HD_e^{-1}H^\top \tag{2.55}$$

The Rodríguez Laplacian can similarly be shown to be the unnormalized Laplacian of the clique expansion of an unweighted graph with every hyperedge weight set to 1. Similarly, Gibson's algorithm calculates the eigenvectors of the adjacency matrix for the clique expansion graph.

We now turn our attention to the normalized Laplacian of Zhou et al. Consider the star expansion of the hypergraph with the weight function $w^z(u,e) = w(e)$. Then the adjacency matrix for the resulting bi-partite graph can be written as

$$S^z = \begin{bmatrix} 0 & HW \\ WH^\top & 0 \end{bmatrix} \tag{2.56}$$

Table 2.1 This table summarizes the various hypergraph learning algorithms, their underlying graph construction and the associated matrix used for the spectral analysis.

Algorithm	Graph	Matrix
Bolla	Clique	Combinatorial Laplacian
Rodríguez	Clique	Combinatorial Laplacian
Zhou	Star	Normalized Laplacian
Gibson	Clique	Adjacency
Li	Star	Adjacency

It is easy to show that the degree matrix for this graph is the diagonal matrix

$$D^z = \begin{bmatrix} D_v & 0 \\ 0 & WD_e \end{bmatrix} \tag{2.57}$$

Thus the normalized Laplacian for this bi-partite graph is given by the matrix

$$\begin{bmatrix} I & -D_v^{-1/2}HW^{-1/2}D_e^{-1/2} \\ -D_e^{-1/2}W^{-1/2}H^\top WD_v^{-1/2} & I \end{bmatrix}$$

Now if we consider the eigenvalue problem for this matrix, with eigenvectors $\mathbf{x}^\top = [\ \mathbf{x}_v \quad \mathbf{x}_e\ ]$ then by Lemma 2.2, we can show that $\mathbf{x}_v$ is given by the following eigenvalue problem.

$$\begin{aligned} D_v^{-1/2}HWD_e^{-1}H^\top D_v^{-1/2}\mathbf{x}_v &= (1-\lambda)^2\mathbf{x}_v \\ (I - D_v^{-1/2}HWD_e^{-1}H^\top D_v^{-1/2})\mathbf{x}_v &= (1-(1-\lambda)^2)\mathbf{x}_v \end{aligned} \tag{2.58}$$

This is exactly the same eigenvalue problem that Zhou et al. propose for the solution of the clustering problem. Thus Zhou et al.'s Laplacian is equivalent to constructing a star expansion and using the normalized Laplacian defined on it. The following table summarizes this discussion.

Thus we have shown that a variety of methods for analyzing hypergraphs despite their very different formulations, can be reduced to two graph constructions – the star expansion and the clique expansion – and the study of their associated Laplacians. We have also shown that for the commonly occurring case of k-uniform graphs these two constructions are identical. This is a surprising and unexpected result as the two graph constructions are completely different in structure. In the case of non-uniform graphs, we showed that the essential difference between the two constructions is how they weigh the evidence from hyperedges of differing sizes.

2.8 Clique Averaging

In this section, we describe a new algorithm for partitioning hypergraphs. It is a two-step procedure. In the first step we construct a weighted graph that approximates the hypergraph. This approximation is based on a novel algorithm that we call *Clique Averaging*. In the second step we use a spectral clustering algorithm (NCut) based on the normalized Laplacian of the graph to partitioning its vertex set. As the second step of this algorithm is well known, we will focus on the development and properties of Clique Averaging.

We begin with some notation. Since we are interested in constructing a weighted 2-graph from a given weighted undirected hypergraph. We can without loss of generality assume that the 2-graph being constructed is complete, and the only thing to be determined are the edge weights. In the previous sections we have used $w(e)$ to indicate the weight on the hyperedge e. We will use the symbol $w^a(u,v)$ and $w^c(u,v)$ to indicate the edge weighting functions for the clique averaged graph and clique expanded graphs respectively. We make the dependence on the vertices u and v explicit in this notation. For notational simplicity we will assume that the hypergraph H is a k-graph, i.e. all edges have the same degree k.

The two weighting functions w and w^a can then formally be described as

$$w : V^k \to R^+ \quad \text{and} \quad w^a : V^2 \to R^+. \tag{2.59}$$

As the hypergraphs we are dealing with are undirected, the functions w and w^a are symmetric in their arguments, i.e., their value remains the same if the order of the arguments is arbitrarily permuted. Note that given an ordering of the edges, w and w^a can also be represented as vectors.

We are now ready to introduce our hypergraph approximation scheme. Our construction and analysis of the approximation will be based on considering graph approximations of a single hyperedge e. The extension to the whole hypergraph is then a matter of linear superposition. We begin by revisiting the observation that the value of the weighting function $w(e)$ is independent of the order in which we consider the vertices in e. In light of this, when considering the various kinds of graphs that can be associated with the hyperedge e, the only graph structure that satisfies the requirements of symmetry is the k-clique on e. A k-clique is a completely connected graph on k vertices. Thus the task of approximating $w(e)$ boils down to assigning weights to the edges in k-clique associated with e.

As we mentioned earlier, the most widely used such approximation scheme is Clique Expansion, and is based on the assumption that every edge in the clique associated with e has edge weight equal to $w(e)$. Formally

$$w^c(u,v) = w(e), \qquad \forall \quad u,v \in e \tag{2.60}$$

Collecting the above set of equations over all hyperedges results in an over-determined linear system consisting of $\binom{k}{2}\ \binom{n}{k}$ equations. This system has a simple least squares solution given by

$$w^c(u,v) = \frac{1}{\mu(n,k)}\left(\sum_{u,v\in e} w(e)\right). \tag{2.61}$$

Here $\mu(n,k) = \binom{n-2}{k-2}$ is the number of hyperedges that contain a particular pair of vertices. Thus the weight on an edge is the arithmetic mean of the weights of all the hyperedges that contain both of its vertices. Other choices for $\mu(n,k)$ are also possible and will amount to different weighting schemes when working with hyperedges of varying sizes. The optimal choice of weighting in Clique Expansion when combining information across hyperedges is an area of research in itself [54, 75].

The relationship between a hyperedge and the edge weights in its clique in the above approach was the simplest possible, where one assumes that the hyperedge weight and the edge weights are equal to each other. In an attempt to make this relationship richer, we take a generative view of the problem. Let us assume that there exists a $\binom{k}{2}$-ary function F such that, given the edge weights on a k-clique, it returns the corresponding hyperedge weight. Formally

$$w(e) = F\left(w^a\left(v_1,v_2\right),\ldots,w^a\left(v_i,v_j\right),\ldots,w^a\left(v_{k-1},v_k\right)\right). \tag{2.62}$$

Now given a particular generative model F and a hypergraph H, the hypergraph approximation problem can then be stated as the problem of solving for those values of the graph edge weights $w^a(v_i,v_j)$ that satisfy the above equation over all hyperedges simultaneously. Of course how well the graph G captures the structure of hypergraph

H is now a function of F. So what is a good choice of F? We begin our search by demanding some simple properties of F:

1. **Positivity** F should be positive for positive valued arguments.
2. **Symmetry** F should be symmetric in its arguments.
3. **Monotonicity** F should be monotonic in each of its arguments.

Positivity and symmetry are simple consequences of the definition of h. Monotonicity is a reasonable demand to make of F as one would expect that as the interaction between two vertices increases or decreases the strength of the hyperedge would be indicative of that change. Within these constraints there are still very many choices for F. In this paper we consider the family of functions F_p parameterized by the positive scalar p.

$$F_p(x_1, x_2, \ldots, x_u) = \left(\lambda(u) \sum_{i=1}^{u} x_i^p\right)^{1/p}, \quad u = \binom{k}{2} \tag{2.63}$$

where λ is a scalar function of the arity of F_p. We can now write Equation (2.62) as

$$w(e) = \left(\lambda\left(\binom{k}{2}\right) \sum_{\substack{v_i, v_j \in e \\ i<j}} \left(w^h(v_i, v_j)\right)^p\right)^{1/p} \tag{2.64}$$

For brevity we will write $\lambda(k) = \lambda(\binom{k}{2})$. Using this and taking the p^{th} power on both sides gives us

$$(w(e))^p = \lambda(k) \sum_{\substack{v_i, v_j \in e \\ i<j}} (w^a(v_i, v_j))^p \tag{2.65}$$

We note that the above equation states that the l_p norm of the clique weights is proportional to the hyperedge weight. It is also worth noting that as the value of p increases the l_p norm is biased towards the largest clique weight. For a given h and a fixed p this is a linear system in $(w^a(v_i, v_j))^p$. Thus without any loss of generality we can restrict our analysis to the case $p = 1$. With this in mind let us interpret the above equation. Modulo a constant the above equation states that the weight of a hyperedge is the arithmetic mean of the edge weights in the clique it induces. Thus a natural choice for $\lambda(k)$ is $\binom{k}{2}^{-1}$. Other choices for $\lambda(k)$ are possible and will amount to different weighting schemes when working with hyperedges of varying sizes. When working with hyperedges of the same size, which is the case in the current study, the choice of $\lambda(k)$ amounts to a uniform scaling of the resulting graph edge weights. As spectral clustering algorithms are insensitive to such scalings, the exact choice of $\lambda(k)$ is immaterial. For the sake of concreteness we will use the arithmetic mean interpretation of the above equation and thus the name Clique Averaging.

Also without loss of generality we will assume that the set of hyperedges has been ordered in a lexicographic order based on the vertices incident on each hyperedge. A similar ordering is done on the set of graph edges too. We can now define the incidence matrix $\Phi = [\phi_{ij}]$. Φ is a zero-one matrix, that represents the incidence relationship between a hyperedge in H and an edge in G.

$$\phi_{ij} = \begin{cases} 1 & \text{if edge } j \text{ is incident on hyperedge } i \\ 0 & \text{otherwise} \end{cases} \tag{2.66}$$

We say an edge is incident on a hyperedge if the hyperedge contains both of its vertices. The rectangular matrix Φ has $\binom{n}{k}$ rows and $\binom{n}{2}$ columns. Note that Φ is an extremely sparse matrix with each row containing only $\binom{k}{2}$ non-zero entries. Now recall that w^a denotes the vector of graph edge weights of length $\binom{n}{2}$ and, w denotes the vector of hyperedge weights. Then Equation (2.65) for the case of $p = 1$ can be written in matrix form as

$$\Phi w^a = \lambda(k) w \tag{2.67}$$

This equation assumes that $w^a \geq 0$, i.e., each element of the vector w^a is non-negative. Hence when solving for w^a given w, we will explicitly enforce this constraint. When working with hypergraphs with edge weights that are bounded above as in the case of affinities; we will enforce an upper bound $w^a \leq 1$ also. Since the linear system is over-determined, the solution to Equation (2.67) has to be determined by minimizing the least squares error. Thus for the case of a hypergraph with hyperedge weights bounded in the interval $[0, 1]$, its graph approximation is given by the edge weight vector w^a that minimizes the following constrained minimization problem:

$$\min_{w^a} \|\lambda(k)\Phi w^a - w\|_F^2 \qquad 0 \leq w^a \leq 1 \tag{2.68}$$

The above optimization problem is an instance of the Bounds Constrained Least Squares problem. However as we noted earlier Φ is a sparse matrix and thus we can exploit efficient iterative methods for solving it [21]. We use `lsqlin` in MATLAB's Optimization Toolbox.

2.8.1 Duality

In this section we analyze the link between Clique Averaging and Clique Expansion. In the previous section we saw that the graph edge weights as a result of Clique Expansion are given by

$$w^c(u,v) = \frac{1}{\mu(n,k)} \left(\sum_{u,v \in e} w(e) \right). \tag{2.69}$$

using the notation used to describe clique averaging can be re-written as

$$w^e = \frac{1}{\mu(n,k)} \Phi^\top w. \tag{2.70}$$

We use the superscript e to indicate Clique Expansion. From Equation (2.67), we have

$$\lambda(k)\Phi w^a = w \tag{2.71}$$

Equations (2.70) and (2.71) are duals of each other. Multiplying both sides of Equation (2.70) by Φ we get

$$\Phi w^c = \frac{1}{\mu(n,k)} \Phi\Phi^\top w. \tag{2.72}$$

Note that modulo a constant, Equations (2.71) and (2.72) differ only in the right hand side by a pre-multiplication by the matrix $C = \Phi\Phi^\top$. To understand the action of this pre-multiplication let us consider the structure of the matrix C.

C is a symmetric matrix, with rows and columns corresponding to the hyperedges H. The entry in the e row and e' column corresponds to the inner product of the e^{th} and the e'^{th} rows of Φ. Φ as we noted earlier is a zero-one matrix, hence the dot product counts the number of edges in the graph G that the two hyperedges share. These entries are easily calculated, for if $l = |e \cap e'|$ denotes the number of vertices the two hyperedges have in common then

$$C_{ee'} = \binom{|e \cap e'|}{2} = \binom{l}{2}.$$

Let the distance between two hyperedges of size k be $k - l$, then multiplication with the e^{th} row of C is equivalent to multiplying each element of w by a decreasing function of the distance from the hyperedge e and summing over them. This is in fact a convolution of the hyperedge weights by a quadratically decreasing kernel. Thus Cw is a low passed version of w. This implies that Clique Expansion solves the same approximation problem as Clique Averaging, but instead of operating on the original hypergraph it operates on a low passed version of it. We know from basic signal processing theory that low pass filtering is an operation that loses information and in the limit transforms the weight vector w into a constant vector. Hence the approximation produced by Clique Averaging is of a higher quality and better preserves the cluster structure present in the hypergraph H.

2.8.2 Clustering using Clique Averaging

We are now ready to put everything together and describe the full procedure for clustering a data set.

Given a data set, the first step is the construction of the affinity hypergraph H by calculating the affinity for every distinct k-tuple in the dataset. However, calculating $\binom{n}{k}$ hyperedge weights can be computationally prohibitive.

In many cases, the user has a choice of the size of hyperedge when constructing the hypergraph. Using a simple counting argument one can show that since the number of within-cluster hyperedges to the number of between cluster hyperedges goes down geometrically with increasing hyperedge size, the smallest possible value of k should be chosen. Further, we subsample H to obtain a sparse hypergraph H'. Since the column rank of Φ is $\binom{n}{2}$, we need at least that many rows, which in turn puts a lower bound on the number of hyperedges in H'. In our experiments we fix $n_{samples} = 5pn^2$ where p is the number of partitions that the data is to be divided into. We then use Clique Averaging to construct a graph G. To partition the graph into p parts, we use the Normalized Cuts algorithm which uses the first p eigenvectors of the normalized Laplacian of the graph and performs k-means clustering on the resulting k-dimensional embedding [12, 101, 121].

2.9 Experiments

In this section we study the performance of six different algorithms out of which five are hypergraph partitioning algorithms. The sixth algorithm is a multi-round variant of RANSAC. We report the performance of the algorithms on two datasets. The algorithm are

1. Clique Averaging+Ncut (CAVERAGE): The hypergraph is approximated using Clique Averaging and the resulting graph is partitioned using the Normalized Cuts

algorithm.

2. Clique Expansion+Ncut (CEXPAND): The hypergraph is approximated using Clique Expansion and the resulting graph is partitioned using the Normalized Cuts algorithm [148].

3. Gibson's Algorithm-Sum Model (GIBSONS): Gibson et al.'s algorithm operating under the sum model [50].

4. Gibson's Algorithm-Product Model (GIBSONP): Gibson et al.'s algorithm operating under the product model [50].

5. kHMeTiS (KHMETIS): The leading tool for hypergraph partitioning in the VLSI community based on multi-level iterative refinement. We use the publically available implementation [82].

6. Cascading RANSAC (CRANSAC): A simple multi-round extension to the RANSAC algorithm. In the i^{th} round a number of trials are performed to identify that k-tuple that has the highest number of inliers. This k-tuple and its associated inliers are identified as the i^{th} group in the dataset and removed from it.

Reporting unbiased performance comparison of clustering algorithms is a hard problem, since each algorithm that one compares against has one or more free parameters that must be set according to the particular problem at hand. Thus while comparing performance across problems, an approach giving each algorithm the best shot would need to perform a sweep over all possible parameter values. While this might report the best behavior of the algorithm it is clearly not informative about the robustness of the algorithm to parameter choice, a property that is of vital importance to a user who is using the algorithm on a novel dataset. Thus it is important to use an experimental protocol that is as close as possible to real world usage.

One of the ways in which algorithms are tuned is by running them on a small pilot dataset similar to the real problem. This is the basis of our experimental protocol. When running an algorithm over a suite of experiments, we choose a problem that lies at the center of the set of experiments in terms of complexity and choose the best performing parameters using a parameter sweep. This parameter setting is used for all the experiments in the test suite. To be fair to CRANSAC in terms of computation resources, we set the total number of trials it could perform to be equal to the number of hyperedges. GIBSONS and CAVERAGE were run with $p = 1$. The only free parameter

Table 2.2 Clustering results for the k-lines problem. This table reports the results of applying the various clustering algorithms to the k-lineas clustering problem. A total of 350 points were used with 70 points in each cluster. The results are an average of 30 runs.

CAVERAGE	12.6	CEXPAND	12.9
GIBSONS	17.3	GIBSONP	55.1
KHMETIS	18.0	CRANSAC	23.4

across all the hypergraph partitioning algorithms was the parameter σ that was used to convert a dissimilarity d into the affinity $e^{-d/\sigma}$. In case of CRANSAC the error threshold for inlier detection was the free parameter.

2.9.1 k-lines Clustering

In the first experiment we consider the k-lines problem in spaces of dimension greater than two, i.e., given a set of points in $\mathbb{R}^d$, the task then is to partition them into a number of d-dimensional lines. In the case of lines in two dimensions the Hough transform solves this problem quite effectively, but with three or more dimensions there is no convenient parameterization that can be used. Pairwise measures of similarity are not applicable here since any two points are collinear, thus it takes at least three points to determine a measure of collinearity. This is an example of a triadic relationship. The dissimilarity measure on triples of points is their distance to the best fitting line. Our dataset consists of points sampled from gently curving lines with additive noise. All the lines pass through the origin. Thus all clusters overlap with each other to some degree. The lines are generated as arcs of circles with a controllable radius of curvature. We consider the performance of the six algorithms. The results are reported over a dataset containing 5 lines in the cube $[-1, 1]^5$. We sample 70 points from each line for a total of 350 points. The hypergraph was generated by sampling $k^2\binom{n}{2} = 549675$ 3-tuples. For this dataset we considered the performance the five hypergraph partitioning algorithms over varying values of σ. Results in terms of mean error are reported over 30 trials.

A more elaborate picture emerges when one looks at the performance of the algorithms over a range of values of σ. Figure 2.1 and 2.2 plots this behavior. The graph has a number of notable features. We begin by noting that CEXPAND and CAVERAGE are the two best performing algorithms and for small and moderate values of σ there is virtually no difference between their performance. It is however interesting to note that as σ increases further the performance of CEXPAND sharply degrades and reaches 80% error, which is the same as chance. CAVERAGE on the other hand continues to perform well at about 30% error while the other four algorithms are operating at 70% − 80% error. The error curve for KHMETIS is disjointed because for certain values of σ the program crashed.

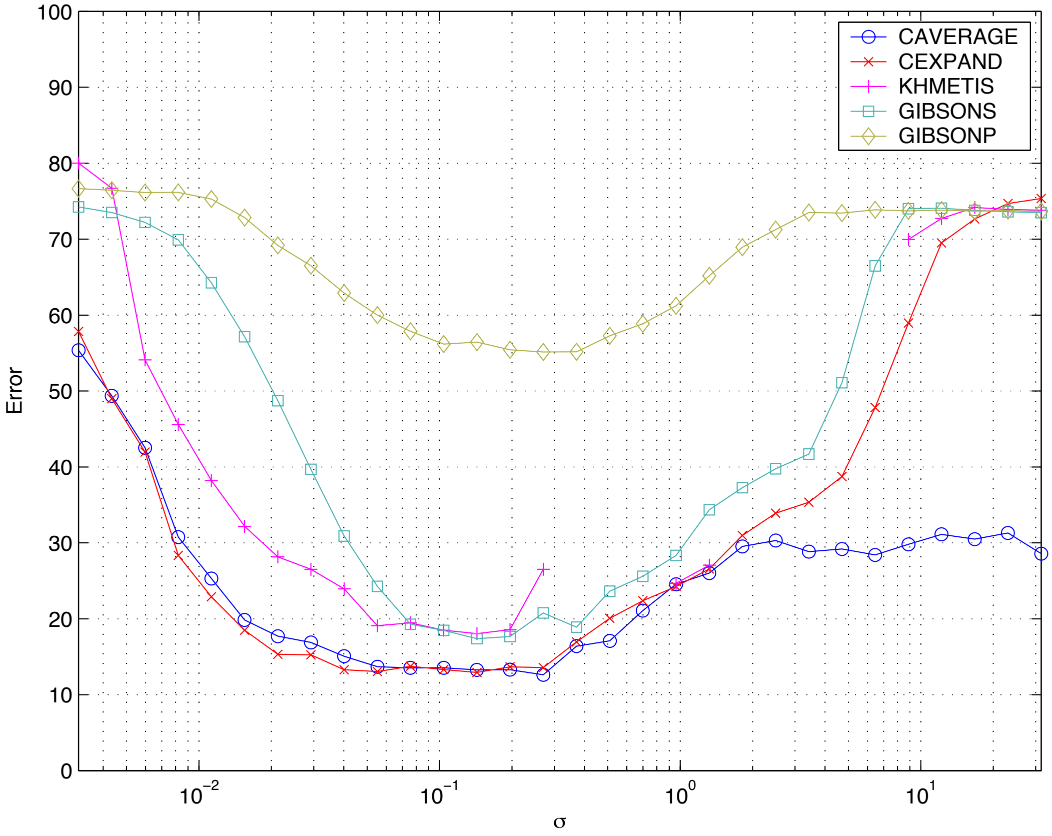


Figure 2.1 Performance of the five hypergraph partitioning algorithms on the k-lines dataset respectively as the scale parameter σ is varied. Note that despite similar best case performance, CAVERAGE is more robust to scale changes than CEXPAND.

2.9.2 Illumination Invariant Clustering

It has been shown that all the images of a Lambertian object illuminated by a point light source lie in a three dimensional subspace [17]. This leads to a natural measure of dissimilarity over four (tetradic) or more images and allows us to perform clustering using it. Indeed this is a generalization of the k-lines problem to the k-subspaces problem. If we assume that the four images under consideration form the columns of a matrix, then

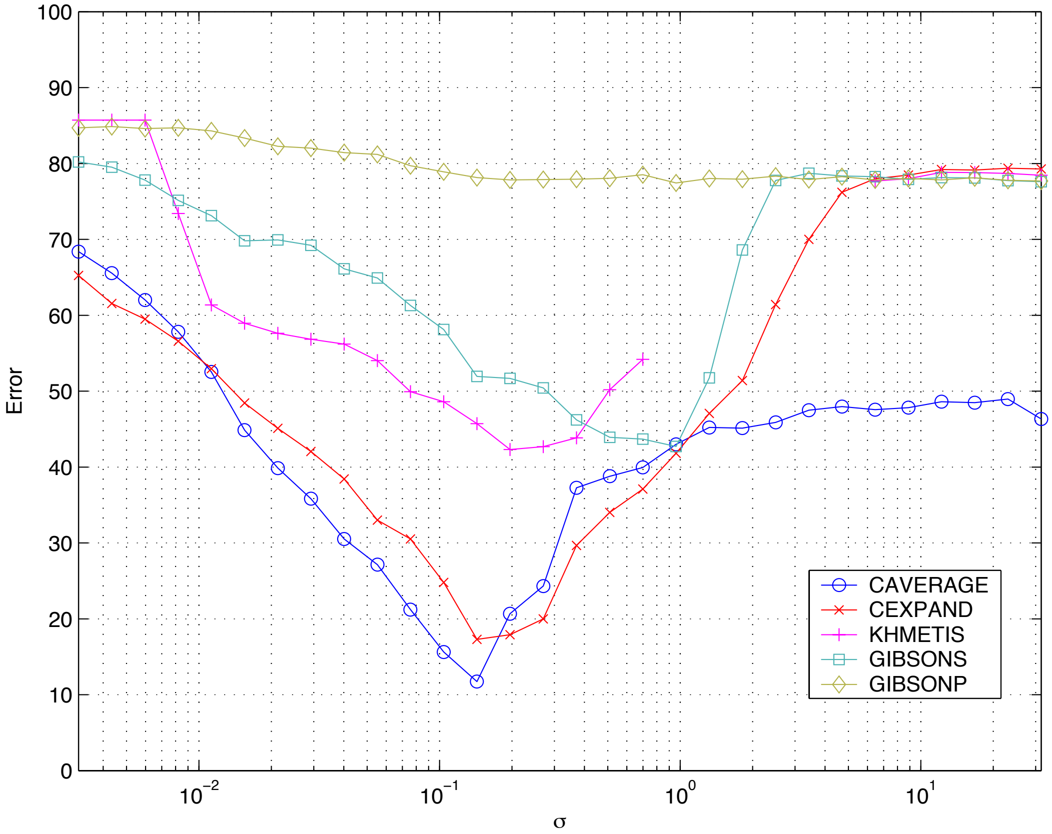


Figure 2.2 This graph plots the performance of the five hypergraph partitioning algorithms on the Yale face data set as the scale parameter σ is varied. Note that like in the case of the k-lines dataset, despite similar best case performance, CAVERAGE is more robust to scale changes than CEXPAND.

$$d = \frac{s_4^2}{s_1^2 + \cdots + s_4^2}$$

serves as a measure of dissimilarity where s_i is the i^{th} singular value of this matrix.

In out experiments we use the Yale database, which contains 45 images each of 10 individuals [49]. The aim of the clustering procedure is to partition the images into groups by identity.

Figure 2.2 shows the result of performing a parameter sweep over the parameter σ for the case of 7 identities. The gross behavior of the algorithms in Figure 2.1 and Figure 2.2 is very similar. Again CAVERAGE and CEXPAND are consistently the best performing algorithms and CAVERAGE is much more robust to changes in the value of the scaling parameter σ. This experiment was used as the basis for tuning the parameters for individual algorithms for the following experiment.

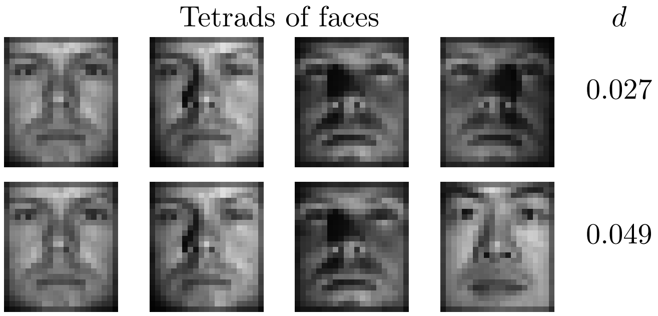


Figure 2.3 Illumination invariant clustering. Row 1: Value of d for four faces of the same individual as the illumination varies. Row 2: d increases when one of the faces belongs to another individual.

Table 2.3 presents the results of running the six algorithms on four subsets of the Yale face dataset with increasing number of points and clusters. Each algorithm was run 30 times with parameters picked by running a parameter sweep over σ case of 7 identities. The results are in the form of mean error/standard deviation.

Table 2.3 Clustering results for the Yale face dataset. This table reports the results of applying the various clustering algorithms to the illumination invariant clustering of the Yale face dataset. The results are an average of 30 runs and the standard deviations are reported in brackets.

	4	6	8	10
CAVERAGE	4.2 (6.3)	12.7 (8.4)	17.4 (4.0)	16.0 (3.0)
CEXPAND	11.8 (3.4)	17.6 (5.4)	21.8 (5.4)	24.9 (4.3)
GIBSONS	25.9 (7.3)	42.2 (3.8)	47.7 (3.0)	51.5 (2.1)
GIBSONP	67.4 (2.3)	75.2 (1.2)	79.7 (0.8)	82.8 (0.7)
KHMETIS	21.5 (4.3)	41.9 (6.8)	38.4 (4.7)	58.3 (3.3)
CRANSAC	16.2 (9.5)	23.6 (9.2)	35.1 (7.9)	37.1 (6.6)

As can be seen in the above table, CAVERAGE beats all other algorithms across the board.

While two problem sets do not make for conclusive evidence, they are indicative of a few general trends. CAVERAGE is much less sensitive to changes in the dynamic range of hyperedge weights, providing empirical verification of the relationship established between CEXPAND and CAVERAGE in Section 2.8.1. It is also consistently the best performing algorithm amongst the six we have tested. It can be shown that the only difference between GIBSONS and CEXPAND is that the former uses the unnormalized Laplacian, while the latter uses the normalized Laplacian. This set of experiments is further evidence that it is preferable to use the normalized Laplacian over its unnormalized variant.

2.10 Discussion

In this chapter we considered the problem of clustering with higher order relations as a hypergraph partitioning problem. We surveyed the various Laplace like operators that have been constructed to analyze the structure of hypergraphs. We showed that all of these methods despite their very different formulations, can be reduced to two graph constructions — the Star Expansion and the Clique Expansion—and the study of their associated Laplacians. We have also shown that for the commonly occurring case of k-uniform graphs these two constructions are identical. In the case of non-uniform graphs, we showed that the essential difference between the two constructions is how they weigh the evidence from hyperedges of differing sizes.

Leveraging a simple additive generative model, we have introduced a new class of hypergraph approximation algorithms which have provably better behavior than existing approximations, for which we have presented empirical proof. We also compared the performance of our proposed algorithm to four existing hypergraph partitioning algorithms and a multi-round variant of RANSAC. In all our experiments, Clique Averaging outperformed its competitors both in terms of clustering error as well as insensitivity to parameter changes in the data.

There remain a number of open questions and directions for future work. The most important question is that of computational complexity. Since we solve for all the graph edge weights, the sampling complexity for the algorithm is lower bounded by $O(n^2)$. However there is evidence that for data that is clusterable into a small number of clusters, spectral clustering can be performed using far fewer than $O(n^2)$ graph edges [48], thus it seems a significant reduction in the sampling complexity of Clique Averaging is possible.

The relation considered between hypergraphs and graphs was a very simple additive model. An interesting area of research is the investigation of more sophisticated generative models.

Hypergraph representations of similarities have close relations to factor graphs used in the study of graphical models, and there seem to be some intriguing connections that are worth exploring. In particular, clustering problem involving hypergraphs in some cases are also related to the area of submodular optimization. Submodular functions are discrete analogs of convex functions and have been the subject of active research in the past decade. It will be interesting to see if its possible to extend recent developments in polynomial time submodular optimization methods to the problem of hypergraph partitioning.

Portions of this dissertation are based on “Higher Order Learning with Graphs” by S. Agarwal, K. Branson, S. Belongie [2] and “Beyond Pairwise Clustering,” by S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman and S. Belongie [4].

I developed the primary equivalence between star expansion and clique expansion, did the literature survey for the papers, developed and analyszed the Clique Averaging algorithm. I also contributed to the design and execution and analysis of the experiments and the writing of the papers.

3

Multidimensional Scaling

> *"I have come to believe that the whole world is an enigma, a harmless enigma that is made terrible by our own mad attempt to interpret it as though it had an underlying truth."*
>
> *Umberto Eco*

Multidimensional scaling (MDS) refers to the general task of assigning Euclidean coordinates to a set of objects such that a given set of dissimilarity, similarity or ordinal relations between the points are obeyed. This assignment of coordinates is also known as an embedding.

Multidimensional scaling algorithms can be classified into two distinct classes based on the kind of structure they pay attention to and preserve in a data set. An algorithm that takes as input pairwise distances or dissimilarities and outputs an embedding that attempts to best preserve these distances/dissimilarities is a *Metric* multidimensional scaling algorithm. On the other hand, an algorithm that only pays attention to the ordinal structure of the dissimilarities, i.e., the relative ordering of the dissimilarities between pairs of points is known as a *Non-Metric* multidimensional scaling algorithm. Non-metric multidimensional scaling finds extensive usage in psychometrics and psychophysics where the actual magnitude of the dissimilarity between pairs is not available, too difficult to measure, or not reliable.

In this chapter we will present a new non-metric multidimensional algorithm which takes as input a set of $\mathcal{S}$ of paired comparisons, i.e.

$$\mathcal{S} = \{(i, j, k, l) | \; \|\mathbf{x}_i - \mathbf{x}_j\|_2 < \|\mathbf{x}_k - \mathbf{x}_l\|_2\} \tag{3.1}$$

This is a special tetradic relation that finds extensive usage in psychophysics experiments where subjects are asked to indicate which out of two pairs of stimuli are more similar. We consider one such case in the Section 3.3 where we will use the proposed non-metric multidimensional scaling algorithm to construct a Euclidean space for the human perception of how surfaces reflect light.

The chapter is organized as follows. In Section 3.1, we start by giving an overview of metric multidimensional scaling, where along the way we will also introduce a semidefinite program for minimizing Kruskal's STRESS-1 error functional. Section 3.2 begins by considering the Kruskal-Shephard non-metric scaling problem and proposes a new algorithm for solving it. We then consider a more general case of non-metric scaling called paired comparisons and propose a semidefinite program for solving it. Section 3.3 describes an experiment for measuring the human perception of reflection and the results of applying our novel non-metric scaling algorithm to the data collected in the experiment.

3.1 Metric Multidimensional Scaling

Let us begin by defining some notation. l_p^d denotes the space $\mathbb{R}^d$ equipped with the l_p norm. In many of the problems we consider, we will be interested in embeddings without any explicit constraints on d. In such cases it is convenient to consider the space l_p which is the space of all infinite sequences $\mathbf{x} = (x_1, x_2, \ldots)$ of real numbers with $\|\mathbf{x}\|_p < \infty$. Note that the space l_p contains each space l_p^d isometrically.

$\mathbb{S}^n$ is the set of symmetric $n \times n$ matrices. $\mathbb{S}_h^n$ is the set of symmetric hollow matrices, i.e., symmetric matrices with a zero diagonal. $\mathbb{S}_+^n$ is the set of symmetric positive semidefinite matrices and $\mathbb{S}_{++}^n$ is the set of $n \times n$ symmetric positive definite matrices. Membership of a matrix K in $\mathbb{S}_+^n$ and $\mathbb{S}_{++}^n$ will be indicated by $K \succeq 0$ and $K \succ 0$ respectively. A matrix $D = [d_{ij}]$ is a Euclidean distance matrix if there exists a matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, such that $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$.

$\mathbb{EDM}^n$ denotes the set of $n \times n$ Euclidean distance matrices. The set $\mathbb{EDM}^n$ is a very rich object with some very interesting properties. We refer the reader to [36] for a comprehensive account. For our purposes, we note that $\mathbb{EDM}^n$ is a convex cone. That $\mathbb{EDM}^n$ is a cone is seen easily by noting that if $D \in \mathbb{EDM}^n$ and X is a Euclidean embedding corresponding to it, then for all $\lambda \geq 0$, $\sqrt{\lambda}X$ is an embedding for λD. Thus, $\lambda D \in \mathbb{EDM}^n$.

Schoenberg [114] showed that

$$D \in \mathbb{EDM}^n \iff \begin{cases} d_{ij} \geq 0 \\ D \in \mathbb{S}_h^n \\ -VDV \succeq 0 \end{cases} . \tag{3.2}$$

where $V = I - \mathbf{1}\mathbf{1}^\top \in \mathbb{S}_h^n$. Here $\mathbf{1}$ is the $n \times 1$ vector of ones. The convexity of $\mathbb{EDM}^n$ then follows from the simple observation that for all $\lambda_1, \lambda_2 \geq 0$ and $D_1, D_2 \in \mathbb{EDM}^n$, $\lambda_1 D_1 + \lambda_2 D_2 \in \mathbb{S}_h^n$. Further, $-V\lambda_1 D_1 V - V\lambda_2 D_2 V \succeq 0$, since a positive combination of positive semidefinite matrices is positive semidefinite. Thus $\lambda_1 D_1 + \lambda_2 D_2 \in \mathbb{EDM}^n$.

It is also straightforward to show that if X is an embedding of $D \in \mathbb{EDM}^n$ such that it is centered at the origin, i.e. the mean of the columns is the zero vector, then if we let $K = [k_{ij}] = X^\top X$ be the Gram matrix, then $K \succeq 0$ and

$$K = -\frac{1}{2}VDV \tag{3.3}$$

or

$$d_{ij} = k_{ii} - 2k_{ij} + k_{jj}. \tag{3.4}$$

$d_{ij}(X) = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$ denotes the (i, j)-th entry of the Euclidean distance matrix $D(X)$ corresponding to the columns of X. The matrix $\Delta = [\delta_{ij}]$ will be used to denote the input matrix of dissimilarities, which in the noise free case we will assume to be a Euclidean distance matrix, i.e., the dissimilarities are squared Euclidean distances. Δ is also referred to as the *pre-distance* matrix.

3.1.1 Measures of fit

A number of different objective functions have been proposed for measuring the agreement between a pre-distance matrix and its corresponding embedding. Following popular usage within the multidimensional scaling literature we will refer to them collectively as Stress functionals. We list the most widely used ones below.

Definition 3.1 (Raw STRESS).

$$\sigma_r = \sum_{ij} \left(\sqrt{\delta_{ij}} - \sqrt{d_{ij}(X)} \right)^2 \tag{3.5}$$

$$= \sum_{ij} \left(\sqrt{\delta_{ij}} - \|\mathbf{x}_i - \mathbf{x}_j\|_2 \right)^2 \tag{3.6}$$

Definition 3.2 (SSTRESS).

$$\sigma_r^2 = \sum_{ij} \left(\delta_{ij} - d_{ij}(X) \right)^2 \tag{3.7}$$

$$= \sum_{ij} \left(\delta_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \right)^2 \tag{3.8}$$

Definition 3.3 (Sammon).

$$\sigma_s = \sum_{ij} \frac{\left(\sqrt{\delta_{ij}} - \sqrt{d_{ij}(X)} \right)^2}{\sqrt{\delta_{ij}}} \tag{3.9}$$

$$= \sum_{ij} \frac{\left(\sqrt{\delta_{ij}} - \|\mathbf{x}_i - \mathbf{x}_j\|_2 \right)^2}{\sqrt{\delta_{ij}}} \tag{3.10}$$

Definition 3.4 (STRESS-1).

$$\sigma_1 = \left(\frac{\sum_{ij} \left(\sqrt{\delta_{ij}} - \sqrt{d_{ij}(X)} \right)^2}{\sum_{ij} d_{ij}(X)} \right)^{1/2} \tag{3.11}$$

$$= \left(\frac{\sum_{ij} \left(\sqrt{\delta_{ij}} - \|\mathbf{x}_i - \mathbf{x}_j\|_2 \right)^2}{\sum_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2} \right)^{1/2} \tag{3.12}$$

Definition 3.5 (STRESS-2).

$$\sigma_2 = \left(\frac{\sum_{ij} \left(\sqrt{\delta_{ij}} - \sqrt{d_{ij}(X)} \right)^2}{\sum_{ij} \left(\sqrt{d_{ij}(X)} - (1/n^2) \left(\sum_{ij} \sqrt{d_{ij}(X)} \right) \right)^2} \right)^{1/2} \tag{3.13}$$

$$= \left(\frac{\sum_{ij} \left(\sqrt{\delta_{ij}} - \|\mathbf{x}_i - \mathbf{x}_j\|_2 \right)^2}{\sum_{ij} \left(\|\mathbf{x}_i - \mathbf{x}_j\|_2 - (1/n^2) \left(\sum_{kl} \|\mathbf{x}_k - \mathbf{x}_l\|_2 \right) \right)^2} \right)^{1/2} \tag{3.14}$$

In each instance, we are solving a matrix nearness problem with the constraint that it belong to the cone of Euclidean distance matrices [65]. Closely related to these matrix nearness problems are Euclidean distance matrix completion problems which have been studied extensively over the past decade [13, 36, 69, 89, 90].

A related problem is the problem of embedding finite metric spaces into normed spaces [95]. The object of study is the pair (X, δ) where X is a set and $\delta : X \times X \to [0, \infty]$ is a metric, i.e a function defined on every pair of elements of X with the following properties

1. $\delta(x, y) = 0 \iff x = y$
2. $\delta(x, y) = \delta(y, x)$
3. $\delta(x, y) + \delta(y, z) \geq \delta(x, z)$

Typically (X, δ) is specified with an $n \times n$ matrix Δ as is the case with multidimensional scaling. Here, we have the additional constraint that δ is a metric. Typical multidimensional scaling problems do not obey this constraint.

Let us now define the following notion of $\mu -$ *embedding.*

Definition 3.6 (μ–embedding of metric spaces)**.** Let (X, δ) and (Y, σ) be two metric spaces. Then a mapping $f : X \to Y$, is called a μ-embedding where $\mu \geq 1$ is a real number, if there exists a real number $r > 0$ such that for all $x, y \in X$

$$r\delta(x, y) \leq \sigma(f(x), f(y)) \leq \mu r\delta(x, y). \tag{3.15}$$

The infimum of the numbers μ such that f is a μ–embedding is called the distortion of f.

The aim is to find embeddings with the smallest possible distortion. The following two propositions, both of which were proved by Bourgain give upper and lower bounds on the distortion of metric embeddings into l_2.

Theorem 3.7 (Bourgain [25])**.** *For all n, there exist n-point metric spaces that cannot be embedded into l_2 with distortion smaller than $c \log n / \log \log n$, where $c > 0$ is a suitable positive constant.*

Theorem 3.8 (Bourgain [25])**.** *Every n-point metric space can be embedded into a Euclidean space with distortion at most $O(\log n)$.*

We now return our attention to the case when Δ is an arbitrary pre-distance matrix which may or may not satisfy the metric constraints. Each of the error functions defined above is a non-linear functional of the matrix X. In addition to the pre-distance matrix, the user may constrain the dimension d of the embedding. In other cases, the search for the best d is part of the multidimensional scaling process.

When the choice of the embedding dimension is free, then given two embeddings with the same residual error, one would in general prefer the one with the lower dimensionality. There are a number of reasons for this. An obvious one is computational complexity. A lower dimensional embedding is computationally easier to work with and to visualize. Another reason is that we want our embedding not only to explain the observed data but also to generalize well to unseen data. Statistical learning theory [134] informs us that for the same training error a simpler model is expected to perform better than a more complex one and should be preferred. This is a formalization of the well known *Occam's Razor*.

3.1.2 Minimizing Stress

We now consider the problem of finding an X that minimizes the desired variant of Stress defined in the last section. Each of the objective functions defined above are non-linear in the embedding matrix X and the distance matrix D. Further, they are non-convex in the embedding matrix X. Thus solving for the globally optimal X is a hard problem. Thus we must satisfy ourselves with approximate solutions.

We will consider two Stress minimization approaches here. The first and the more well known of the two is called *Iterative Majorization*. We illustrate the method by constructing an iterative majorization algorithm for minimizing σ_r. Our presentation closely follows [24].

3.1.3 Iterative Majorization

Iterative majorization as the name suggests is an iterative procedure that directly solves for $X \in \mathbb{R}^{d \times n}$. In each iteration it minimizes a convex quadratic approximation of the objective function. It is guaranteed to reduce the Stress of the embedding at each iteration. Since Stress is bounded from below, in the limit this iterative process is guaranteed to converge to a local minimum. The algorithm is closely related to the popular Expectation Maximization algorithm in the particular form of the approximation function that it uses [37]. We begin with a definition.

Definition 3.9 (Majorizing Function)**.** Let, z be a fixed value. A function $g(x, z)$ is said to majorize the function $f(x)$ if

1. $\forall x,\ f(x) \leq g(x,z)$.

2. $f(z) = g(z,z)$.

Given the majorizing function $g(x,z)$, let $g(x,z)$ be minimized at x^*, then observe that

$$f(x^*) \leq g(x^*,z) \leq g(z,z) = f(z) \tag{3.16}$$

Thus, starting with some point $x^{(0)}$, the sequence

$$x^{(i+1)} = \arg\min_{x} g\left(x, x^{(i)}\right) \tag{3.17}$$

will result in a sequence $\{x^{(i)}\}$ for which the sequence $\{f(x^i)\}$ is a monotonically decreasing sequence. This sequence is bounded below by zero and thus in the limit will converge to a local minimum of $f(x)$.

Minimizing σ_r

Now,

$$\begin{aligned} d_{ij}(X) &= \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \\ &= (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j) \\ &= \text{Trace}\left(X^\top A_{ij} X\right) \end{aligned} \tag{3.18}$$

Where, $A_{ij} = [a_{ij}] \in \mathbb{S}^n$ is a matrix with $a_{ii} = a_{jj} = 1$ and $a_{ij} = a_{ji} = -1$. Further, let $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{d \times n}$, then by Cauchy-Schwarz inequality we have

$$\begin{aligned} (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{z}_i - \mathbf{z}_j) &\leq \|\mathbf{x}_i - \mathbf{x}_j\|_2 \|\mathbf{z}_i - \mathbf{z}_j\|_2 \\ -\|\mathbf{x}_i - \mathbf{x}_j\|_2 &\leq -\frac{(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{z}_i - \mathbf{z}_j)}{\|\mathbf{z}_i - \mathbf{z}_j\|_2} \\ -\sqrt{d_{ij}(X)} &\leq -\frac{\text{Trace}\left(X^\top A_{ij} Z\right)}{\sqrt{d_{ij}(Z)}} \end{aligned} \tag{3.19}$$

Thus, $-\frac{\text{Trace}\left(X^\top A_{ij} Z\right)}{\sqrt{d_{ij}(Z)}}$ majorizes $-\sqrt{d_{ij}(X)}$. Using these two observations, we can now construct a function $\tau(X,Z)$ that majorizes the weighted version of $\sigma_r(X)$ as follows,

$$
\begin{aligned}
\sigma_r(X) &= \sum_{ij} w_{ij} \left(\sqrt{\delta_{ij}} - \sqrt{d_{ij}(X)}\right)^2 \\
&= \sum_{ij} w_{ij}\delta_{ij} - 2w_{ij}\sqrt{\delta_{ij}}\sqrt{d_{ij}(X)} + w_{ij}d_{ij}(X) \\
&\leq \sum_{ij} w_{ij}\delta_{ij} - 2w_{ij}\frac{\text{Trace}\left(X^\top A_{ij} Z\right)}{\sqrt{d_{ij}(Z)}} + w_{ij}\,\text{Trace}\left(X^\top A_{ij} X\right) \\
&= \delta^* - 2\,\text{Trace}\left(X^\top \left(\sum_{ij} \frac{w_{ij}A_{ij}}{\sqrt{d_{ij}(Z)}}\right) Z\right) + \left(X^\top \left(\sum_{ij} w_{ij}A_{ij}\right) X\right) \\
\tau(X, Z) &\triangleq \delta^* - 2\,\text{Trace}\left(X^\top B(Z) Z\right) + \text{Trace}\left(X^\top C X\right)
\end{aligned}
\tag{3.20}
$$

Thus, $\tau(X, Z)$ majorizes $\sigma_r(X)$. Where,

$$\delta^* = \sum_{ij} w_{ij}\delta_{ij} \tag{3.21}$$

$$B(Z) = \sum_{ij} \frac{w_{ij}A_{ij}}{\sqrt{d_{ij}(Z)}} \tag{3.22}$$

$$C = \sum_{ij} w_{ij}A_{ij}. \tag{3.23}$$

Given a fixed Z, minimizing the quadratic form $\tau(X, Z)$ w.r.t. X is simple. We take derivaties w.r.t. X and set it to zero to obtain

$$\nabla\tau(X, Z) = 2CX - 2B(Z)Z = 0 \tag{3.24}$$

which gives us the solution

$$X = C^\dagger B(Z) Z \tag{3.25}$$

Where, $C^\dagger = (C^\top C)^{-1} C^\top$ is the Moore-Penrose inverse of C. In the unweighted case, i.e., $w_{ij} = 1, \forall\ i, j = 1, \ldots, n$, $C^\top$ has a special form:

$$C^\dagger = \frac{1}{n}V = \frac{1}{n}I - \frac{1}{n^2}11^\top \tag{3.26}$$

Thus, the iterative majorization algorithm for minimizing σ_r is given by the following iterative update rule.

$$X^{(k)} = C^\dagger B\left(X^{(k-1)}\right) X^{(k-1)} \tag{3.27}$$

3.1.4 Semidefinite Programming for Multidimensional Scaling

The second approach to Stress minimization is based on the observation that, if we ignore the rank constraint, then the optimization over $X \in \mathbb{R}^{n \times n}$ can be replaced with an optimization over Euclidean distance matrices $D \in \mathbb{EDM}^n$ or Gram matrices $K \in \mathbb{S}^n_+$. As we observed earlier, both $\mathbb{EDM}^n$ and $\mathbb{S}^n_+$ are convex sets. Moreoever, since every matrix in $\mathbb{EDM}^n$ corresponds to a matrix in $\mathbb{S}^n_+$, optimization problems over $\mathbb{EDM}^n$ can frequently can be written as semidefinite programming problems.

Semidefinite programming refers to optimization problems of the form

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{subject to} \quad & F_0 + x_1 F_1 + x_2 F_2 + \ldots + x_n F_n \succeq 0, \quad F_i \in \mathit{symset} \\ & A\mathbf{x} = \mathbf{b} \end{aligned} \tag{3.28}$$

Semidefinite programs(SDPs) are convex optimization problems that contain Linear programs as a special case. They can be solved efficiently using interior point methods [7, 100, 133]. Several software packages exist for solving SDPs [19, 23, 123, 125].

Semidefinite programming based approaches to solving multidimensional problems are then two step procedures. In the first step the rank constraint on the Stress functional is relaxed to obtain a convex semidefinite program, and the optimal solution to this SDP is determined, let it be K^*. In the second step, a matrix $X \in \mathbb{R}^{d \times n}$ is extracted from K^* by observing that if $K^* = U \Sigma U^\top$ is the eigenvalue decomposition of K^* and U_d and Σ_d are submatrices of U and Σ corresponding to the d-largest eigenvalues of K^*. Then $X^{*\top} X^*$, where $X^* = \Sigma_d^{1/2} U_d^\top$ is the best rank d approximation to K [42].

We refer to X^* as the best rank d projection of of K^* and denote it by $X = \Pi_d(K^*)$.

Classical multidimensional scaling

The most well known instance of the rank constrained semidefinite programming based approach to multidimensional scaling is Classical multidimensional scaling (CMDS). The objective function in CMDS is σ_r^2 (SSTRESS). SSTRESS is the same as the squared Frobenius norm of the difference between the pre-distance matrix Δ and the Euclidean distance matrix $D(X)$,

$$\sigma_r^2 = \|\Delta - D(X)\|_F^2. \tag{3.29}$$

The classical approach is to transform this from a Euclidean distance matrix nearness problem to a Gram matrix nearness problem. Δ is converted into a *pre-gram* matrix using the the same transform that converts $D(X)$ into a Gram matrix , i.e. the following problem is considered

$$\min_X \|V\Delta V - VD(X)V\|_F^2 \tag{3.30}$$

$$\equiv \min_X \|V\Delta V - K(X)\|_F^2 \tag{3.31}$$

As long as the rank is is not constrained, the problem can be stated purely in terms of the Gram matrix K

$$\begin{aligned} \min_K \quad & \|V\Delta V - K\|_F^2 \\ \text{subject to} \quad & K \succeq 0 \end{aligned} \tag{3.32}$$

This is a convex semidefinite program. However, it is not necessary to use a SDP solver to solve it, and a closed form solution is possible as a consequence of the following theorem

Theorem 3.10 (Higham [64]). *Let $A \in \mathbb{R}^{n\times n}$ and let $B = (A + A^\top)/2$ and $B = U\Sigma U^\top$ be its eigenvalue decomposition. Then if Σ_+ is the diagonal matrix obtained from Σ by setting all its negative entries to zero, then $U\Sigma_+U^\top$ is the best symmetric positive semidefinite matrix approximation to A in the Frobenius norm.*

Thus, the solution returned by Classical multidimensional scaling is $\Pi_d(U\Sigma_+U^\top)$.

The first step of the above algorithm where the matrix nearness problem is solved for K instead of D is suspect since solving for K involves solving a weighted version of the original problem. The weighting matrix V, which can depending upon the matrix Δ can cause warping of the matrix K. Further, it has been shown that the SSTRESS functional minimization creates artifacts in the resulting embedding [73].

In [28] the authors argue that a better two-step procedure is to solve

$$\begin{aligned} \min_{D} \quad & \|\Delta - D\|_F^2 \\ \text{subject to} \quad & D \in \mathbb{EDM} \end{aligned} \tag{3.33}$$

or

$$\begin{aligned} \min_{D} \quad & \|\Delta - D\|_F^2 \\ \text{subject to} \quad & -VDV \succeq 0 \\ & d_{ij} \geq 0 \\ & D \in \mathbb{S}_h^n \end{aligned} \tag{3.34}$$

This is also a semidefinite program and can be solved to global optimality. However unlike the CMDS strategy of working with the Gram matrix, no closed form solution to the above problem is known and thus a full blown SDP solver must be used where an eigensolver sufficed earlier. Semidefinite programming is a fairly recent development and SDP solvers in general consume significantly more time and memory space than an eigensolver. Thus the simplicity and efficiency of the classical MDS solution procedure explains its popularity despite the various issues associated with it.

Minimizing σ_r

The minimization of STRESS (σ_r) using semidefinite programming is little less intuitive. We show the rank unconstrained solution presented in [36].

$$\sigma_r = \sum_{ij} \left(\sqrt{\delta_{ij}} - \sqrt{d_{ij}} \right)^2 \tag{3.35}$$

$$= \sum_{ij} \delta_{ij} - 2\sqrt{\delta_{ij}}\sqrt{d_{ij}} + d_{ij} \tag{3.36}$$

Now, the optimization problem to be solved can be stated as

$$\begin{aligned} \min_{D} \quad & \sum_{ij} \delta_{ij} - 2\sqrt{\delta_{ij}}\sqrt{d_{ij}} + d_{ij} \\ \text{subject to} \quad & D \in \mathbb{EDM}^n \end{aligned} \tag{3.37}$$

Let us now introduce a matrix $T = [t_{ij}],\ t_{ij} \geq 0$ of non-negative auxilliary variables that are lower bounds for $\sqrt{d_{ij}}$. Then the above optimization problem can be re-written as

$$\begin{aligned} \min_{D,T} \quad & \sum_{ij} \delta_{ij} - 2\sqrt{\delta_{ij}} t_{ij} + d_{ij} \\ \text{subject to} \quad & \sqrt{d_{ij}} \geq t_{ij} \\ & t_{ij} \geq 0 \\ & D \in \mathbb{EDM}^n \end{aligned} \tag{3.38}$$

or

$$\begin{aligned} \min_{D,T} \quad & \sum_{ij} d_{ij} - 2\sqrt{\delta_{ij}} t_{ij} \\ \text{subject to} \quad & \begin{bmatrix} 1 & t_{ij} \\ t_{ij} & d_{ij} \end{bmatrix} \succeq 0 \\ & t_{ij} \geq 0 \\ & D \in \mathbb{EDM}^n \end{aligned} \tag{3.39}$$

The first set of constraints is a collection of n^2 positive semidefinite constraints. The particular form is a consequence of the Schur Complement Lemma [26].

Sammon's Mapping

Sammon's error function is just a weighted version of the σ_r functional and can be optimized by solving the following simple variant of the above optimization problem

$$\begin{aligned} \min_{T,D} \quad & \sum_{ij} \frac{d_{ij}}{\sqrt{\delta_{ij}}} - 2t_{ij} \\ \text{subject to} \quad & \begin{bmatrix} 1 & t_{ij} \\ t_{ij} & d_{ij} \end{bmatrix} \succeq 0 \\ & t_{ij} \geq 0 \\ & D \in \mathbb{EDM}^n \end{aligned} \tag{3.40}$$

Minimizing σ_1

In this section we consider the problem of minimizing σ_1 or Kruskal's STRESS-1 using a semidefinite program To the best of our knowledge, the SDP derived below is not known in the literature.

$$\sigma_1^2 = \frac{\sum_{ij} \left(\sqrt{\delta_{ij}} - \sqrt{d_{ij}}\right)^2}{\sum_{ij} d_{ij}} \tag{3.41}$$

$$= \frac{\sum_{ij} \delta_{ij} - 2\sqrt{\delta_{ij}}\sqrt{d_{ij}} + d_{ij}}{\sum_{ij} d_{ij}} \tag{3.42}$$

The STRESS-1 minimization problem can now be re-written as

$$\begin{aligned} \min_{D} \quad & \frac{\sum_{ij} \delta_{ij} - 2\sqrt{\delta_{ij}}\sqrt{d_{ij}} + d_{ij}}{\sum_{ij} d_{ij}} \\ \text{subject to} \quad & D \in \mathbb{EDM}^n \end{aligned} \tag{3.43}$$

Similar to the previous section we can transform this optimization problem into the following form:

$$\begin{aligned} \min_{D,T} \quad & \frac{\sum_{ij} \delta_{ij} - 2\sqrt{\delta_{ij}}t_{ij}}{\sum_{ij} d_{ij}} \\ \text{subject to} \quad & \begin{bmatrix} 1 & t_{ij} \\ t_{ij} & d_{ij} \end{bmatrix} \succeq 0 \\ & t_{ij} \geq 0 \\ & D \in \mathbb{EDM}^n \end{aligned} \tag{3.44}$$

The above optimization problem has a linear fractional objective. Let us introduce a scalar γ, which is an upper bound on the objective function. Then the above objective function can be re-written as

$$\begin{aligned} \min_{D,T,\gamma} \quad & \gamma \\ \text{subject to} \quad & \gamma \geq \frac{\sum_{ij} \delta_{ij} - 2\sqrt{\delta_{ij}}t_{ij}}{\sum_{ij} d_{ij}} \\ & \begin{bmatrix} 1 & t_{ij} \\ t_{ij} & d_{ij} \end{bmatrix} \succeq 0 \\ & t_{ij} \geq 0 \\ & D \in \mathbb{EDM}^n \end{aligned} \tag{3.45}$$

which can further be re-written as

$$\begin{aligned}
\min_{D,T,\gamma} \quad & \gamma \\
\text{subject to} \quad & \gamma \sum_{ij} d_{ij} \geq \sum_{ij} \delta_{ij} - 2\sqrt{\delta_{ij}} t_{ij} \\
& \begin{bmatrix} 1 & t_{ij} \\ t_{ij} & d_{ij} \end{bmatrix} \succeq 0 \\
& t_{ij} \geq 0 \\
& D \in \mathbb{EDM}^n
\end{aligned} \tag{3.46}$$

Observe that for a fixed value of γ the above optimization problem is reduced to a feasibility problem that is a semidefinite program. This allows us to solve for the minimum value of γ by using a bisection search on γ where at each step we solve the following feasibility problem

$$\begin{aligned}
\sum_{ij} \gamma d_{ij} + 2\sqrt{\delta_{ij}} t_{ij} \geq \sum_{ij} \delta_{ij} \\
\begin{bmatrix} 1 & t_{ij} \\ t_{ij} & d_{ij} \end{bmatrix} \succeq 0 \\
t_{ij} \geq 0 \\
D \in \mathbb{EDM}^n
\end{aligned} \tag{3.47}$$

The resulting solution can then be projected onto a rank d subspace as usual.

3.2 Non-metric Multidimensional Scaling

The discussion in the last few sections was focused on finding a Euclidean distance matrix that was close to some given pre-distance matrix. We now turn to Non-metric multidimensional scaling. Consider the following two problems:

Problem 1 (Shepard-Kruskal) Given $\Delta = [\delta_{ij}]$ a predistance matrix, find $X \in \mathbb{R}^{d\times n}$ such that

$$\forall\ i,j,k,l \quad d_{ij}(X) \leq d_{kl}(X) \iff \delta_{ij} \leq \delta_{kl} \tag{3.48}$$

Problem 2 (Paired Comparisons) Given a set $\mathcal{S}$ of quadruples (i, j, k, l) find $X \in \mathbb{R}^{d\times n}$ such that

$$(i, j, k, l) \in \mathcal{S} \iff d_{ij}(X) \leq d_{kl}(X) \tag{3.49}$$

In the first case, we are interested in finding an embedding that respects the ordinal information contained in a given pre-distance matrix Δ. In the second case we are explicitly given a set of ordinal relation that relate two pairs of distances and the objective is to find an embedding that respects these relations. In both cases the object of interest is the special tetradic relation $d_{ij} < d_{kl}$. It is easy to see that the first problem is a special case of the second, as any matrix Δ can be converted into a set of ordinal relations $\mathcal{S}$.

In the following we will begin by considering Shephard-Kruskal non-metric scaling, one of the earliest and best known variants of non-metric multidimensional scaling. This algorithm solves the first problem (3.48) mentioned above. We briefly consider the classical solution to this problem and then propose a new algorithm for solving it based on semidefinite programming. We then consider the more general case of problem two (3.49) and propose a algorithm for solving it. Finally we apply this algorithm to analyze the human perception of how surfaces reflect light and construct a perceptual space for reflectance.

3.2.1 Shephard-Kruskal Non-metric Scaling

Shepard and Kruskal considered the following variant of Kruskal's STRESS-1 functional

$$\sigma_{1,n} = \frac{\sum_{ij}\left(\sqrt{d_{ij}(X)} - \theta\left(\delta_{ij}\right)\right)^2}{\sum_{ij} d_{ij}(X)} \tag{3.50}$$

Here, $\theta(\cdot)$ is an arbitrary monotonic function of its arguments. Comparing to 3.4 we note that unlike the metric case, we do not take the square root of $\theta(\cdot)$ in the numerator. Since, the square root is a monotonic function, it is subsumed in the definition of $\theta(\cdot)$.

The optimization is done over X and $\theta(\cdot)$ simultaneously. Since θ is an arbitrary monotonic transform of the data, the above objective function only pays attention to the relative ordering of amongst the elements of Δ. Kruskal proposed an alternating minimization algorithm that alternates between finding the best fitting matrix $D(X)$ to $\theta(\delta_{ij})$ which is an instance of the standard metric multidimensional scaling problem for the input dissimilarities $\theta(\delta_{ij})$, and finding the monotonic transformation $\theta(\cdot)$ that will best *align* $D(X)$ and Δ. This alignment step is also known as Isotonic Regression [16].

Isotonic regression in this case can be formulated as the following simple quadratic program.

$$\begin{aligned} \min_{\theta_{ij}} \quad & \sum_{ij} \left(\sqrt{d_{ij}} - \theta_{ij}\right)^2 \\ \text{subject to} \quad & \theta_{ij} < \theta_{kl} \ \forall \ (i,j,k,l) \ s.t. \ \delta_{ij} \leq \delta_{kl} \end{aligned} \tag{3.51}$$

The above is a convex quadratic programming problem and can be solved optimally in polynomial time [26].

A New Algorithm

Algorithm 1 A new alternating minimization algorithm for solving Shephard-Kruskal Non-metric multidimensional scaling

```
Require: Predistance matrix $\Delta$, output dimension $d$, tolerance $\epsilon$.
$r = \infty, n = 0$
$\theta_{ij}^n = \delta_{ij}$
while $r > \epsilon$ do
    $n = n + 1$
    Solve for the $D^n$ that best approximates $\Theta^{n-1} = [\theta_{ij}^{n-1}]$ using (3.47). {Minimize Kruskal's STRESS-1.}
    Solve for $\Theta^n = [\theta_{ij}^n]$ that best fits $D^n$ using (3.51). {Isotonic regression.}
    $r = \|\Theta^n - \Theta^{n-1}\|_2$
end while
$X = \Pi_d(-VD^nV)$ {Convert to a Gram matrix and find best rank $d$ embedding}.
```

In light of the method described in Section 3.1.4 we can now propose a new alternating minimization algorithm for non-metric scaling problem, see Algorithm 1. Each step of this procedure solves a convex optimization problem which can be solved to the optimum in polynomial time. At each step the STRESS-1 for the embedding is reduced. Since STRESS-1 is lower bounded by 0, the algorithm will eventually converge to a local minimum.

3.2.2 Paired Comparisons

We now turn our attention to the more general case of Problem 2 mentioned earlier. As we mentioned earlier, Problem 2 subsumes Problem 1. We will present a novel algorithm for learning a low rank embedding from such a collection of order relations. The method is related in spirit to the recent work on learning kernel matrices [88] and learning distance metrics from relative comparisons [116]. We will pose the problem as a regularized rank unconstrained optimization problem.

$\mathcal{S}$ is a set of 4-tuples (i, j, k, l) defined as

$$\mathcal{S} = \{(i, j, k, l) | d_{ij} < d_{kl}\} \tag{3.52}$$

The set $\mathcal{S}$ is allowed to have repetitions and inconsistencies. As in classical MDS we convert the problem into one that can be stated in terms of the Gram matrix K.

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = k_{ii} - 2k_{ij} + k_{jj}, \tag{3.53}$$

This equality allows us to define the set $\mathcal{S}$ as

$$\mathcal{S} = \{(i, j, k, l) | k_{ii} - 2k_{ij} + k_{jj} < k_{kk} - 2k_{kl} + k_{ll}\} \tag{3.54}$$

Our aim is to find a Gram matrix, K, which satisfies inequality constraints of the above form for every quadruplet (i, j, k, l) that is a member of $\mathcal{S}$.

The set of inequality constraints above are not sufficient to determine a positive semidefinite matrix K uniquely. This is because the relative comparison constraint has a scale, translation and rotation ambiguity. Translating a point set in space does not change the interpoint distances and scaling the entire point set preserves the relative ordering of every pair of distances. These ambiguities can lead to numerical instabilities in the convex solver. Finally, even though the embedding we construct is ambiguous up to a rotation as is the case with classical MDS, the Gram matrix K is rotation invariant.

The translation ambiguity is eliminated by demanding that the embedding be centered at the origin, i.e.,

$$\sum_b \mathbf{x}_b = 0, \tag{3.55}$$

which can be restated as

$$\begin{aligned} \left\| \sum_b \mathbf{x}_b \right\|_2^2 &= 0 \\ \left(\sum_a \mathbf{x}_a \right)^\top \left(\sum_b \mathbf{x}_b \right) &= 0 \\ \sum_{ab} \mathbf{x}_a^\top \mathbf{x}_b &= 0 \\ \sum_{ab} k_{ab} &= 0. \end{aligned} \tag{3.56}$$

This is a linear equation in the entries of the matrix K.

Handling the scale ambiguity is a bit more complicated. To prevent the embedding from collapsing into the origin, we constrain the scale of the embedding from below. We will demand that for a relative comparison to be valid the two distances should be different by at least 1 unit distance.

$$k_{ii} - 2k_{ij} + k_{jj} + 1 \leq k_{kk} - 2k_{kl} + k_{ll}. \tag{3.57}$$

Two things should be noted here. One, what was a strict inequality earlier has now been converted into a non-strict one. Two, the choice of 1 as the minimum difference between pairs of distances is arbitrary and does not affect the quality of the embedding. The minimum separation constraint, bounds the scale of the embedding from below. We have not constrained the scale of the embedding from above yet. We will deal with this shortly.

Collecting the constraints that we have imposed on our Gram matrix, we get the following feasibility problem

$$\begin{aligned} k_{kk} - 2k_{kl} + k_{ll} &\geq k_{ii} - 2k_{ij} + k_{jj} + 1 \quad \forall (i,j,k,l) \in \mathcal{S} \\ \sum_{ab} k_{ab} &= 0 \\ K &\succeq 0 \end{aligned} \tag{3.58}$$

The above formulation assumes that there does indeed exist a positive semidefinite matrix which satisfies every relative comparison in the collected data. This is clearly not true in general. To get around this problem we introduce slack variables ξ_{ijkl} in every inequality constraint which allow for violations of the inequality, and consider that Gram matrix which minimizes the total slack violation.

$$\begin{aligned}
\min_{K,\xi_{ijkl}} \quad & \sum_{(i,j,k,l)\in\mathcal{S}} \xi_{ijkl} \\
\text{subject to} \quad & k_{kk} - 2k_{kl} + k_{ll} - k_{ii} + 2k_{ij} - k_{jj} \geq 1 - \xi_{ijkl} \quad \forall (i,j,k,l) \in \mathcal{S} \\
& \sum_{ab} k_{ab} = 0 \\
& K \succeq 0
\end{aligned} \tag{3.59}$$

This introduction of slack variables is very similar to the formulation of soft-margin support vector machines [115].

At this stage, we can proceed in a manner similar to the methods for metric multidimensional scaling described earlier and solve the above optimization problem to learn the Gram matrix K and then use its eigenvector decomposition to obtain an embedding. However, in many instances as we discussed earlier, there are additional properties that the user may demand of the embedding obtained in this manner. Of particular interest are embeddings with low rank, i.e. embeddings in which the data is embedded in a low dimensional space. In such cases we introduce a regularizer in the above optimization problem that trades off embedding complexity with the fitting error or total slack. For instance, we consider the following optimization problem

$$\begin{aligned}
\min_{K,\xi_{ijkl}} \quad & \sum_{(i,j,k,l)\in\mathcal{S}} \xi_{ijkl} + \lambda \,\mathrm{rank}(K) \\
\text{subject to} \quad & k_{kk} - 2k_{kl} + k_{ll} - k_{ii} + 2k_{ij} - k_{jj} \geq 1 - \xi_{ijkl} \quad \forall (i,j,k,l) \in \mathcal{S} \\
& \sum_{ab} k_{ab} = 0 \\
& K \succeq 0
\end{aligned} \tag{3.60}$$

Here, λ is a positive scalar that controls the tradeoff between the violations and the rank of the matrix, i.e., the complexity of our model. This is also known as the Bias–Variance tradeoff. Unfortunately, this is an optimization problem that we cannot solve efficiently, as the rank of a matrix is a non-convex function. Indeed, minimizing the rank of a symmetric positive semidefinite matrix subject to linear inequality constraints is an NP-hard problem [43].

To deal with the problem of non-convexity of the objective function we use a standard heuristic from the convex programming literature. Instead of solving the original problem, we relax the rank(K) to its convex envelope Trace(K) [43]. Reformulating the objective function in terms of the slack variables and the trace of the matrix has an additional benefits. It constrains the scale of K from above since its a minimization problem.

The relaxation gives us the following semidefinite program

$$\begin{aligned}
\min_{K,\xi_{ijkl}} \quad & \sum_{(i,j,k,l)\in\mathcal{S}} \xi_{ijkl} + \lambda \,\mathrm{Trace}(K) \\
\text{subject to} \quad & k_{kk} - 2k_{kl} + k_{ll} - k_{ii} + 2k_{ij} - k_{jj} \geq 1 - \xi_{ijkl} \quad \forall (i,j,k,l) \in \mathcal{S} \\
& \sum_{ab} k_{ab} = 0 \\
& K \succeq 0
\end{aligned} \tag{3.61}$$

There is an intuitive explanation for using the trace of the matrix K as the convex regularizer. The rank of a symmetric matrix can be restated as the number of non-zero eigenvalues, or the l_0(counting) norm of the vector of its eigenvalues. A commonly used convex relaxation for problems involving finding the sparsest vector is to replace the objective with the l_1 norm of this vector. For a symmetric positive semidefinite matrix the trace is exactly that, the l_1 norm of the vector of eigenvalues. Another way of interpreting the regularizer is to note that the sum of the eigenvalues of K is the variance of the embedding. Thus it implies that for all embeddings with the same slack violation we will choose the one that has the lowest variance.

A Special Case

A particularly interesting special case for which we shall see applications in the next section is when $j = l$, which gives rise to the following regularized optimization problem.

$$\begin{aligned}
\min_{K,\xi_{ijkl}} \quad & \sum_{(i,j,k)\in\mathcal{S}} \xi_{ijk} + \lambda \operatorname{Trace}(K) \\
\text{subject to} \quad & k_{kk} - 2k_{kj} - k_{ii} + 2k_{ij} \geq 1 - \xi_{ijk} \quad \forall (i,j,k) \in \mathcal{S} \\
& \sum_{ab} k_{ab} = 0 \\
& K \succeq 0
\end{aligned} \tag{3.62}$$

Learning Distance Functions

In the previous section, we developed a method for performing multi-dimensional scaling for paired comparisons. This method relies entirely on the paired comparisons to construct the embedding. Now suppose, we are given a vector representation $X \in \mathbb{R}^{d\times n}$. Say for example vectors indicating actual photometric measurements of the reflectance of a surface. There is nothing to indicate that Euclidean distance between the points $\{\mathbf{x}_i\}$, will capture how humans perceive the relative similarity between pairs of surfaces reflecting light. In such a case one would want to to learn a transformation of the matrix X such that the distances in the transformed space agree with human preference.

Let us consider the case of measuring Euclidean distances induced by a linear transformation A applied to X. Then, givem two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ the distance between the transformed vectors is

$$\begin{aligned}
\|A\mathbf{x}_i - A\mathbf{x}_j\|_2^2 &= (A\mathbf{x}_i - A\mathbf{x}_j)^\top (A\mathbf{x}_i - A\mathbf{x}_j) \\
&= (\mathbf{x}_i - \mathbf{x}_j)^\top A^\top A(\mathbf{x}_i - \mathbf{x}_j)
\end{aligned} \tag{3.63}$$

Now, let $B = A^\top A$, then $B \succeq 0$ is a symmetric positive semidefinite matrix. If X is the data matrix, then the gram matrix in the transformed space is given by $K = X^\top BX$. The search for the optimal linear transformation can now be replaced with the search for the optimal positive semidefinite matrix B. Following a derivation similar to the one in the previous section the corresponding optimization problem is as follows:

$$\begin{aligned}
\min_{B,\xi_{ijkl}} \quad & \sum_{(i,j,k,l)\in\mathcal{S}} \xi_{ijkl} \\
\text{subject to} \quad & k_{kk} - 2k_{kl} + k_{ll} - k_{ii} + 2k_{ij} - k_{jj} \geq 1 - \xi_{ijkl} \quad \forall (i,j,k,l) \in \mathcal{S} \\
& K = X^\top B X \\
& B \succeq 0
\end{aligned} \tag{3.64}$$

Notice, there is no zero-mean constraint in the above optimization problem. This is because the vectors are already situated in a vector space and there is no translation ambiguity. The above program does not have a regularization term. If it is desired, a suitable regularization term $\Omega(K)$ in the form of $\Omega(K) = \text{Trace}(K)$ or $\Omega(K) = \text{Trace}(K^2)$ can be added to the objective function as before.

Finally we note that while the above formulation is linear, it is simple to extend it to non-linear distance functions [88].

3.3 Experiments

In this section we use the non-metric MDS algorithm described in Section 3.2.2 to construct a low dimensional space that captures how humans perceive the reflection of light from a surface.

The Bi-directional Reflectance Distribution Function (BRDF) is a local model of how light is reflected from a surface [102]. It is the ratio of reflected radiance exiting from a surface in a particular direction to the irradiance incident on it. It is a four dimensional function, two to parameterize the direction of incident illumination and two to parameterize the direction of reflected illumination.

While there has been a great deal of progress in creating physically-based analytic BRDF models [10, 33, 60, 129, 136] and more recently measurement driven models [97]. For each of these reflectance models, there exists some underlying space of possible reflectances whose dimension is given by the parameters of the model. Yet, these models and their resulting spaces do not account for the ways people actually perceive materials, e.g. which attributes of a material are significant and which ones are ignored.

In this section we analyze the perception of reflectance and introduce a methodology for deriving a space for reflectance from results of psychophysical experiments using measured BRDF data. We use the MIT-MERL BRDF database as basis of our study [96, 97].

Reflectance can often be broken into two distinct components: chromatic and achromatic. In this study we will restrict our attention to the achromatic aspects of the BRDF, also known as gloss. Gloss was originally studied in the paper industry [74] and has been formalized by the American Society for Testing and Materials (ASTM). The ASTM defines gloss as "the angular selectivity of reflectance, involving surface-reflected light, responsible for the degree to which reflected highlights or images of objects may be seen as superimposed on a surface" [11]. In a BRDF, gloss is responsible for changes in the magnitude and spread of the specular highlight as well as the change in reflectance that occurs as light moves away from the normal toward grazing angles.

We chose to consider only gloss because the largest publicly available database of reflectance measurements (the MIT-MERL database [97]) consists of only 55 usable isotropic BRDFs. This is a very small subset of the vast variety of reflectance functions. Color is such a strong perceptual cue that given the sparseness of our BRDF database, differences in color between two BRDFs will completely overwhelm differences due to the gloss. Thus, in the following, the term BRDF will refer to the achromatic aspects of reflectance; when we refer to the chromatic aspects, we will make specific note of it.

3.3.1 Experimental Design

The aim of our experiment is to capture the perceptual similarity for varying reflectance functions. Each participant was shown a series of triplets of rendered images with constant geometry and illumination, but with varying BRDFs and was asked to indicate whether the center image was more similar to the image on the left or to the image on the right (Figure 3.1 shows a screenshot from one such test).

The images used in the experiment all contain the Stanford bunny [131] rendered under constant illumination and viewing direction with 55 BRDFs from the MIT/MERL BRDF database [97]. The database contains a large representative set of materials including metals, paints, fabrics, minerals, synthetics, and organic materials. Examples of some of these BRDFs appear in figure 3.2.

We used natural illumination since it has been shown that subjects have more discriminative power under this type of illumination than under simple and/or synthetic lighting [46]. We used the same illumination conditions as Fleming et al. [46]. We chose the bunny model because it is simple yet provides a more varied distribution of surface

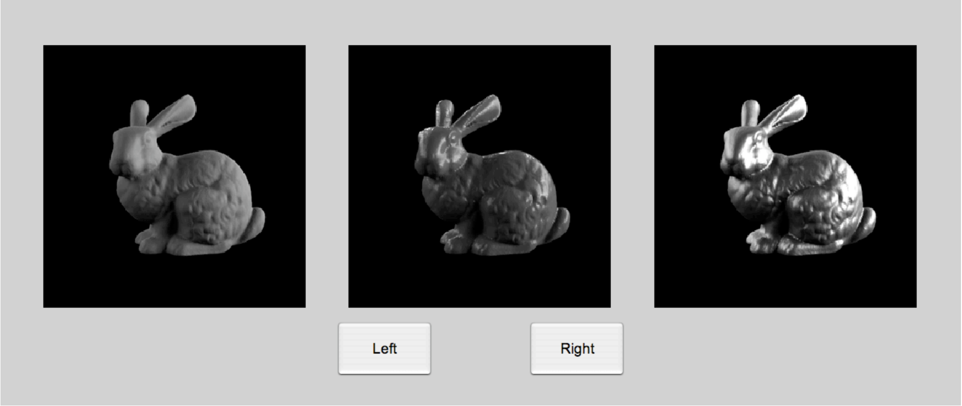


Figure 3.1 Screen capture from the distance comparison test. The subject is asked to click on the appropriate button to indicate which pair appears more similar, **Left**: Left + Middle, or **Right**: Middle + Right.

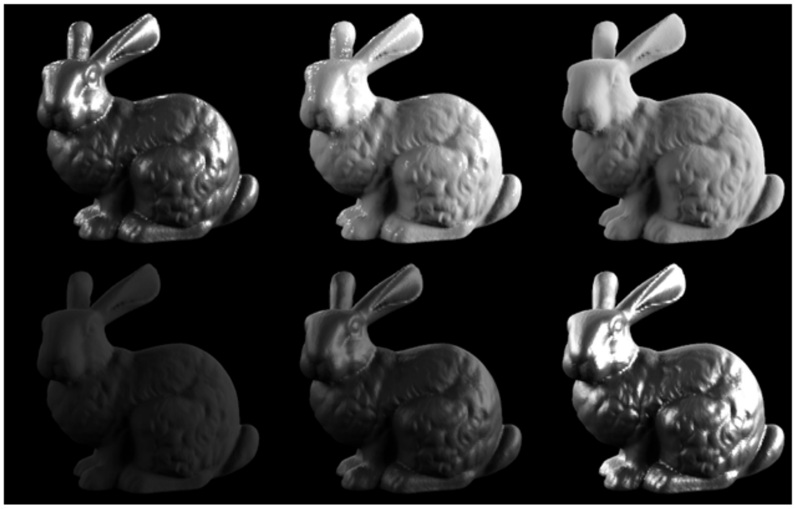


Figure 3.2 Example BRDFs. Six of the 55 images used in our psychophysics study. While monochromatic, they have widely varying gloss properties. The BRDFs used include metals, paints, fabrics, minerals, synthetics, and organic materials.

normal/incident direction combinations than a sphere. Each image was rendered under the same high dynamic range illumination using structured importance sampling [6]. As in previous work [46, 103], we used Tumblin's rational sigmoid [130] to map the rendered high dynamic range images to our low dynamic range displays. The images were rendered in color and then converted to grayscale for our experiment. Our displays have a maximum brightness of 180 cd/m^2.

As there are over $78,000$ possible triplets, only a randomly sampled subset of comparisons could be performed. In our study 75 subjects performed 200 comparisons each for a total of 15,000 comparisons (there were a small number of repeated comparisons). All subjects were unaware of the aim of the experiment and all had normal or corrected to normal vision. The triplets were chosen at random for each subject.

3.3.2 Analysis

Though the MIT-MERL database with 55 BRDFs is a significant step forward in terms of the availability of measured reflectance data. 55 BRDFS capture a small fraction of the space of BRDFs. It is therefore important that care is taken before making any inferences from it. The inferences we make should not just explain the observed data points, but their expected performance over the unobserved portions of the space should be good as well. Only then can we be sure that our conclusions are not just an artifact of our particular data set but say something more general about perception. Given a set of subject responses to paired comparisons on the same 55 BRDFs, we measure the error of an embedding as the average number of paired comparisons that are violated if we use the pairwise distance between BRDFs in the embedded space as our estimate of the distance between them.

The expected error of an estimator over an independent test set is called the *test error* or *generalization error* [58]. The training error is typically smaller than testing error, the two quantities can differ by an arbitrary amount. Thus, when reporting the performance of our statistical estimates from the data, it is important to report an estimate of the generalization error and not just the training error.

Another problem that one faces in problems like the one we are solving is that of model selection. In the last section we argued that simpler models or lower complexity estimates are to be preferred to higher complexity ones. However, it is also the case that higher complexity models typically fit the training data better than lower complexity models. In our case the regularization parameter λ controls the complexity of our embedding. But how does one choose the optimal value of λ? If one could estimate the generalization error for the various choices of λ then one could choose that λ for which the error was the lowest.

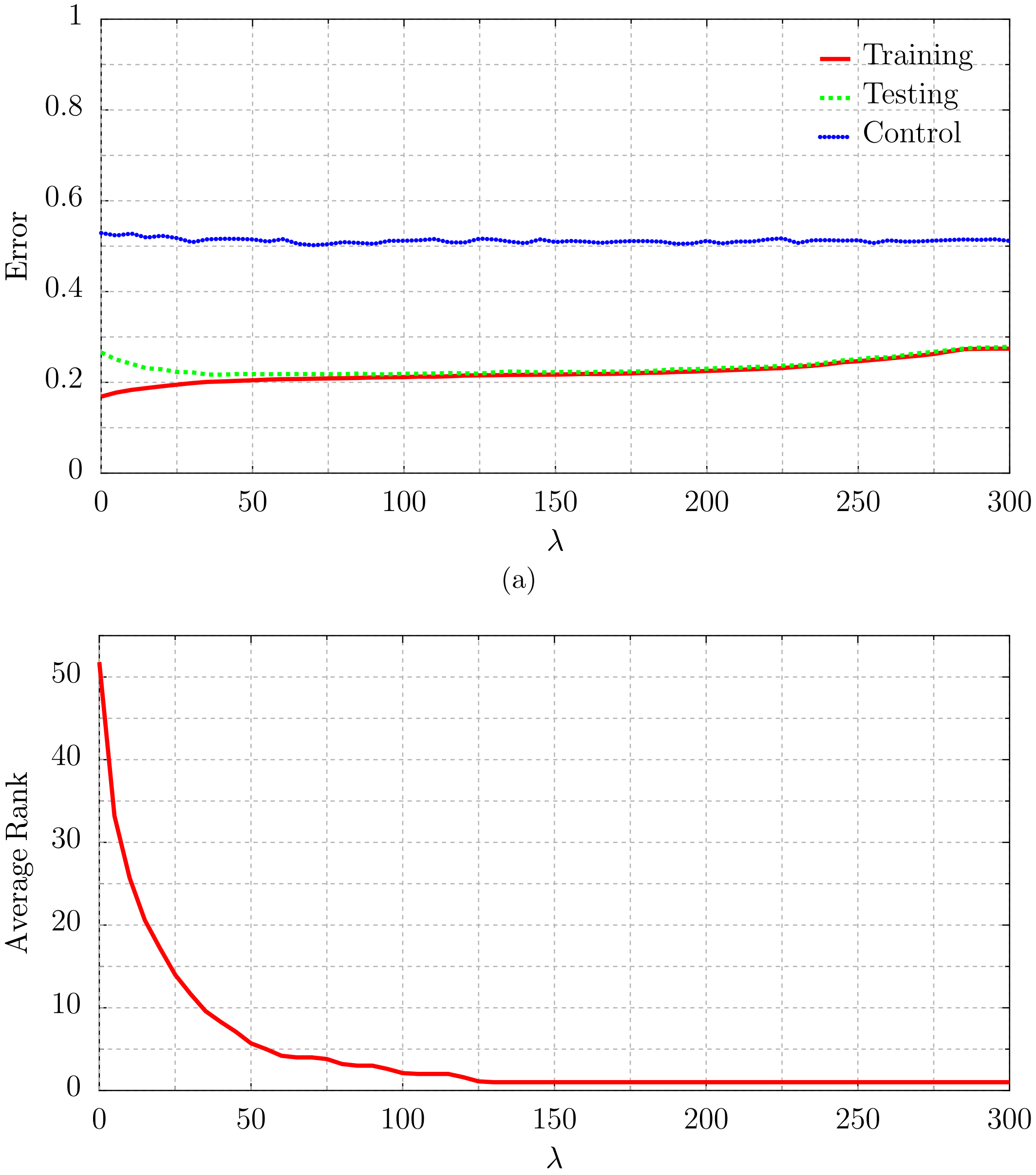
The most widely used method for estimating generalization error and model selection is cross-validation [58]. In k-fold cross-validation the data set (the set of human responses) is split into k roughly equal parts. At the i^{th} iteration, the model is fitted (embedding is learned) using $k-1$ parts of the data excluding the i^{th} part which is then used for measuring the prediction error. The final prediction error estimate is the mean of the k error estimates obtained in this manner. Typical choices of k are 5 or 10. We use 10-fold cross-validation in this study.

We ran our MDS algorithm on the data for varying values of λ between 0 and 300, and performed 10-fold cross-validation for each value of λ. Figure 3.3(a) plots the training (Red) and testing error (Green) as a function of the parameter λ. Figure 3.3(b) plots the average rank of the embedding as a function of λ. As expected, the rank of the embedding goes down as λ is increased.

To ensure, that our data does indeed contain structure and we are learning from it, we performed the following control experiment. We generated a new dataset by taking each triplet (i, j, k) in our dataset and randomly swapping i and k. This is equivalent to a random observer's response if he were shown exactly the same set of comparisons. We then learned an embedding for varying values of λ and measured the test error using cross-validation.

In Figure 3.3(a) the blue curve plots this error. As can be seen the test error never goes below 50%. The consistent and significant gap between the blue and the green curves indicates that our data set is far from purely random.

The choice of a non-zero λ indicates a tradeoff between the rank of the matrix K and total amount of violation in the paired comparisons. Setting $\lambda = 0$ would focus the attention of the MDS algorithm entirely on reducing the violations. Doing so results in matrix K that has a training error of 17%. The resulting embedding has 53 dimensions (which is only 2 less than the maximum). The cross-validation error for $\lambda = 0$ is 27%. This is a significant gap and indicates the poor generalization ability of this embedding. Without any regularization the algorithm comes up with a complex model that overfits to noise in the training data, resulting in poor performance on test data. As the regularization increases, the training error increases, but the testing error decreases at first and then starts to go back up again. This is because as we increase the penalty for higher rank embeddings, the algorithm trades model complexity for training error.



(a)

(b)

Figure 3.3 Cross Validation and Rank. (a) Training (red) and Testing (green) error curves for varying choices of the regularization parameter λ for our MDS algorithm. Testing error (blue) for the randomized control set. (b) Average rank as a function of the regularization parameter.

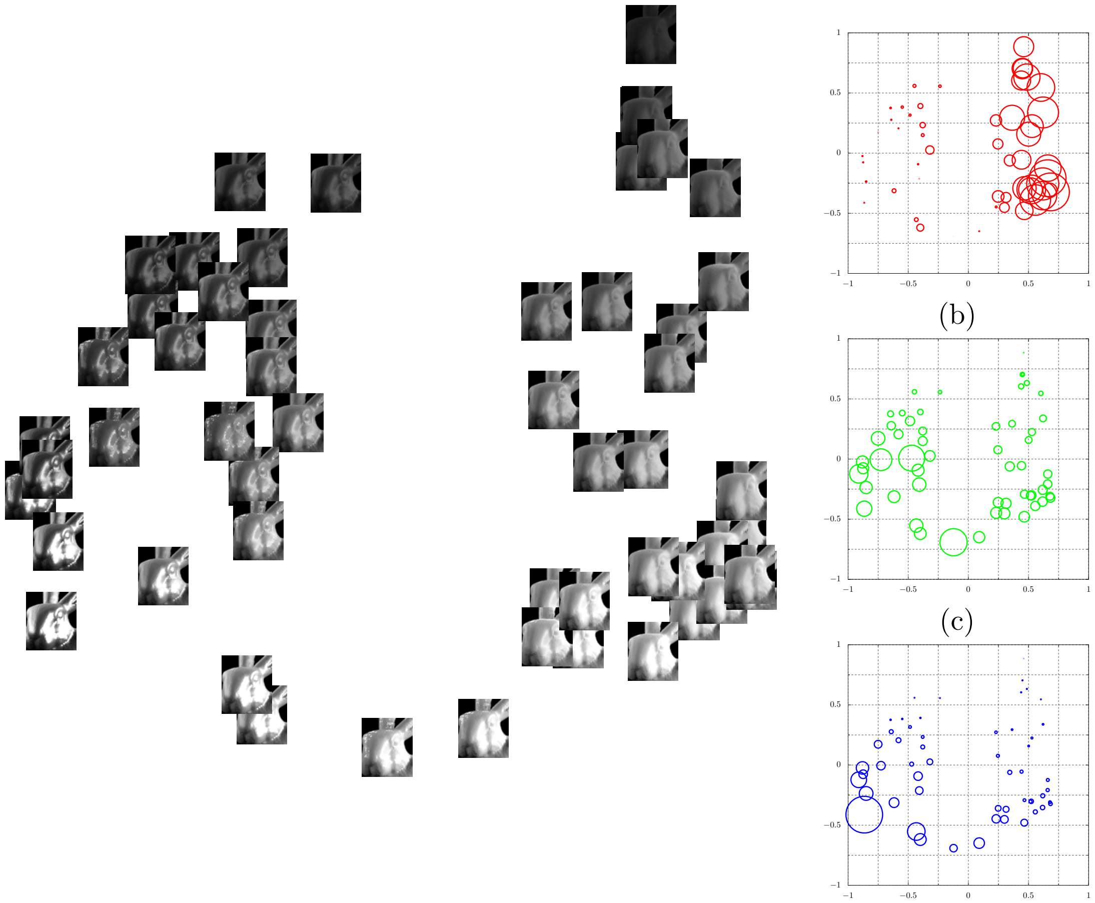
The simpler lower dimensional model does not overfit to the noise leading to an increase in generalization performance. However, as the regularization parameter continues increasing, the algorithm is biased too strongly towards a low rank embedding, ultimately being restricted to one dimension. One dimension is not enough to explain the set of relative comparisons leading to a high test error.

The embedding with the lowest measured cross-validation error has a training error of 21.9% and a test error of 21.3%. The embedding has over 95% of the variance contained in the first two dimensions. Truncating the embedding at two dimensions increased the test error by 0.5%. We do not consider this significant. This embedding is a significant improvement in terms of test error as well as the complexity of the embedding obtained with $\lambda = 0$. To put these numbers in perspective, a trivial upper bound on the test error of an embedding is 50% since a purely random predictor or even one that gives the same answer every time will on average get half of the paired comparisons right. Using the L_2 distance between sampled BRDF vectors as the distance function results in 37.5% error and the inter-subject error was 17%.

Further, we analyzed the stability of this embedding. We constructed 55 different embeddings corresponding to leaving the response data corresponding to one of the BRDFs out at a time. Each of the embeddings produced in this manner was then aligned via a similarity transformation to its corresponding 54 points in the final embedding reported above, and the average squared distortion was measured [132]. Paired comparisons are invariant to similarity transformations. To establish a scale for these errors, the average distance between pairs of points in the global embedding was calculated.

The root mean squared distortion was 2.7e-2 and the average distance between points in the global embedding was 8.7e-1. This is an error of 3% or an order of magnitude difference. This is indicative of the stability of the embedding produced by analyzing our data using the algorithm proposed in this paper.

Figure 3.4(a) shows the optimal 2-D embedding with cropped windows of the BRDF images displayed in the locations of the BRDF in the new space. Notice the clustering of the BRDFs into two distinct clumps and the similarity amongst the images corresponding to them in each clump. There are also two pronounced trends in the embedding, a vertical trend with the darker BRDFs at the top gradually getting brighter with the brightest BRDFs at the bottom. The other roughly breaks the BRDFs into two clusters: the primarily diffuse BRDFs and those that have a strong glossy or specular component. It is also interesting that the metallic BRDFs are all in the lower left corner



(b)

(c)

(a)

(d)

Figure 3.4 Perceptual Embedding. (a) The optimal 2-D embedding with cropped windows of the BRDF images displayed in the locations of the BRDF in the new space. (b-d): Contrast Gloss (b), Specular Gloss (c) and Haze (d) values shown for BRDFs in the embedding. The diameter of the circles corresponds to the value for each property.

and the fabrics are in the upper right corner. This embedding is based entirely on the user preference data; no BRDF or image data was used, which points to the significant descriptive power contained in the paired comparison data.

The American Society for Testing and Materials defines six dimensions for the perception of gloss. Figure 3.4(b)-(d) show plots of three of the ASTM gloss dimensions in our embedding space. The position of each circle corresponds to one of the BRDFs in the embedding space and the diameter corresponds to the measurement of the BRDF in the ASTM gloss dimension. We chose to plot contrast gloss, specular gloss and haze since they were the ASTM dimensions mentioned as significant in previous work [103, 138].

Figure 3.4(b) shows the measurements of each BRDF for contrast gloss. Notice that there is a strong horizontal trend with contrast gloss increasing from left to right. Since contrast gloss is the ratio of light reflected far from the specular direction to the light reflected in the specular direction, it will be higher for matte materials and lower for materials with a strong specular component.

Figure 3.4(c) shows the measurements of each BRDF for specular gloss at 20°. The measurements exhibit a trend increasing from the lower left corner to the upper right corner. This correlates to the trend we noticed before with the glossy materials on the left and the metallic materials in the lower left corner.

Figure 3.4(d) shows the measurements of each BRDF for haze. There is a strong trend increasing from the lower left corner to the upper right corner. This is a measure of the light that is reflected 5° off specular and may be less sensitive to noise, which may explain the fewer number of outliers when compared to the measurements taken on specular.

3.4 Discussion

In this section we considered the problem of multidimensional scaling from the perspective of modern convex optimization. In particular we focused our attention on non-metric multidimensional scaling. We proposed a new alternating minimization procedure for minimizing Kruskal's STRESS-1 functional. We then considered a more general formulation for non-metric multidimensional scaling which subsumes the Kruskal-Shepard formulation. A special case of this formulation was used for analyzing human subject data from a psychophysics experiment designed to capture the human perception of material reflection.

A number of very interesting questions and avenues of future research have arisen as a consequence of the work described in this chapter. We describe some of them below.

Counting comparisons Let $n > 1$ be a natural number. What is the minimum number of paired comparisons needed to completely order some set of n objects where a total ordering of these objects is known to exist? If you could choose which comparisons to make, then you would ask for $O(n \log n)$ comparisons, since the number of comparisons needed to sort a list of n numbers. But now if you were to be given these comparisons *a priori*, then can you do better than $O(n^2)$ comparisons? What is the expected number of comparisons needed? This question is important in that the answer to it tells the investigator how much data to collect so that the resulting embedding is well constrained.

Counting EDMs We say two Euclidean distance matrices $A \in \mathbb{EDM}^n$ and $B \in \mathbb{EDM}^n$ are *order equivalent* if $\forall i, j, k, l\ A_{ij} \leq A_{kl} \iff B_{ij} \leq B_{kl}$. Two Euclidean distance matrices that are not order equivalent are called order-distinct. It is easy to see that order equivalence is an equivalence relation. The question now is, what is the number of order equivalent classes in $\mathbb{EDM}^n$. The number of order equivalent classes is the number of unique non-metric embeddings in the noise free case. The same question can be asked for elements of the $\mathbb{S}_+^n$.

Limit behaviour Recently there has been some research that has focused on analyzing the large sample behavior of the combinatorial and the normalized Laplacian operators, where it has been shown that under suitable conditions they converge to their continuous analogs. Similarly it would be useful to show that in the limit of infinite data, the non-metric multdimensional scaling algorithm presented in this chapter will recover the true distances between every pair of points.

Custom Solvers While interior point methods for solving semidefinite programs have polynomial space and time complexity, they are still fairly expensive when it comes to solving large scale problems, space complexity being a particularly troublesome issue. In many instances it is possible to exploit problem structure to construct specialized solvers for problems which have significantly lower space complexity. Examples include projected gradient/subgradient and other first order methods. Development of such methods for the various semidefinite programs described in this chapter will facilitate their use in the analysis of large real life data sets.

Out of Sample Extension Suppose we have constructed an embedding for n objects based on a set $\mathcal{S}$ of relative comparison involving them. An interesting and important problem is that of *out of sample extension*, where we now have relative comparisons involving a new object. Now is it possible to find the coordinates of this new point in the embedding space in an efficient manner without recalculating the entire embedding with the relative comparisons corresponding to the new object added to $\mathcal{S}$. Initial analysis indicates that this is a hard non-convex optimization problem.

Clustering Given a set of relative comparisons over some set of objects, is it possible to construct an algorithm for clustering these objects on the basis of these relative comparisons? One answer is to use the kernel matrix K learnt as part of the embedding process as the similarity matrix in a pairwise clustering algorithm. But this requires solving an SDP first, which while efficient is still expensive. Ideally one would like a clustering algorithm that works directly with the relative comparisons without constructing an embedding as an intermediate step. This has applications in constructing collaborative filtering and recommendations systems.

Portions of this chapter are based “Toward a Perceptual Space for Reflectance” by J. Wills, S. Agarwal, D. Kriegman and S. Belongie [141]. I was responsible for the development of the non-metric multidimensional scaling algorithm, performed the data analysis, helped with the psychophysics experiments and contributed to the writing of the paper.

4

Other Work

"This is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning"

Winston Churchill

In my time in graduate school, I have worked on a variety of problems. This dissertation presents the results of my work on two of these problems. In the following I will briefly mention the other problems I have worked on and my contributions to their study.

4.1 Illumination Sampling

Environment maps are high dynamic range spherical photographs that are used to capture the directional illumination in a scene. They are used to render synthetic objects in a manner that is radiometrically correct for subsequent compositing into the scene.

In [6] we introduced a new technique for efficiently rendering scenes illuminated using an environment map. The technique is based on two developments, one, a new importance metric that takes into account both the illumination intensity and the expected variance due to occlusion in the scene and two, a novel hierarchical stratification or clustering algorithm that uses this metric to partition the environment map into a set of directional light sources. The resulting approximation trades a small amount of bias for one to two orders of magnitude increase in rendering speed over the best previous Monte Carlo techniques.

This method has been implemented for production use at Weta Digital Inc.

and Rhythm & Hues Inc., and was used for rendering scenes in the feature film King Kong.

4.2 Refraction

Refraction is an optical phenomenon that has not received much attention in the machine vision literature, despite its widespread occurrence all around us. In [5] we derived a novel generalization of the optical flow equation for the case of refraction and presented a method for recovering the refractive structure of an object from a video sequence acquired as the background behind the refracting object moves. By structure here we mean a representation of how the object warps and attenuates (or amplifies) the light passing through it. We distinguished between the cases when the background motion is known and unknown and showed that when the motion is unknown, the refractive structure can only be estimated up to a six-parameter family of solutions without additional sources of information. The optical flow constraint derived in this work is in fact not limited to the study of refraction and can be applied to general image distortion functions such as reflection from curved mirrors.

4.3 Perception of Reflectance

The Bidirectional Reflectance Distribution Function (BRDF) describes the way a surface reflects light. BRDFs are complex mathematical objects that, while allowing for a complete radiometric description of light reflecting from a surface, can be difficult to use in practice. As it stands today, a digital artist has to develop a feel for the parameters of the various analytical or data driven reflectance models before he can use them to produce the desired effect. This is due to the complex relationship between model parameters and the resulting perceptual sensation. Learning this relationship is a complicated and error prone process based on repeated trial and error. Imagine trying to select a desired color by specifying parameters that define a spectral density function. A perceptual space for reflectance allows computer graphics artists to readily navigate the space of BRDFs and to work with BRDFs in a manner similar to how they work with various color spaces. Additionally, shading is a strong cue in human vision and an understanding of reflectance perception will give us insight into the priors and constraints used by humans to solve various shading related problems, e.g., shape from shading and recognition over variable and unknown lighting.

In [141], we design and carry out a comprehensive psychophysical study of the perception of measured reflectance. This is the largest study of its kind to date, and the first to use real material measurements. The data collected is analyzed using a novel multidimensional scaling algorithm. As part of this analysis we estimate the dimensionality of the space of reflectance perception and construct a perceptually meaningful embedding of these BRDFs. We also introduce a novel perceptual interpolation scheme that uses the embedding obtained from human subject responses and the geometry of the space of BRDFs to provide the user with an intuitive interface for navigating the space of reflectances and constructing new ones.

4.4 Tissue Microarray Analysis

Analysis of protein expression directly within cells in histological sections is an important tool in the fight against cancer. Correlating protein expression patterns across many patients with one type of cancer requires reading the protein levels in the tissues, a task that has largely been carried out manually thus far. An important tool in this kind of analyses are tissue microarrays (TMA). A single tissue microarray slide can represent as many as a few thousand patients' tumors. These tissue samples are stained with multiple clinical stains with overlapping color spectra that can vary from one staining to another. The pathologists task is then to evaluate the protein expression by measuring the amount of staining due to a particular stain. This is a tedious and time consuming task, limiting the rate at which these studies can be done.

In [107, 108], we modeled the light transport in stained tissue sections and used non-negative matrix factorization to decompose multi-spectral images of the tissue microarray into constituent stain images. The procedure is fully automated, eliminating the need for a trained pathologist and the resulting decompositions match ground truth obtained from single staining and the scoring algorithm matches the scores given by a trained pathologist to within 6% error. This is the best reported performance on this task till date.

4.5 Global Optimization for Multiview Geometry

Multiview geometry is one of the success stories of computer vision. Methods for recovering the three dimensional structure of a scene from multiple images and the projective transformations that relate the scene and its images are now the workhorse subroutines in applications ranging from specialized tasks like match move in film making to consumer products like image mosaicing for digital camera users.

The key step in each of these methods is the solution of an appropriately formulated optimization problem. These optimization problems are typically highly non-linear and finding their global optimum in general has been shown to be NP-hard. Methods for solving these problems are based on a combination of heuristic initialization and local optimization to converge to a *locally* optimal solution. Very little if anything can be said about the quality of such solutions and their relation to the true global optimum.

In on-going work [3,29] we have developed methods for finding provably optimal solutions to a number of problems in projective geometry, including camera resectioning, homography estimation, multi-view triangulation and the metric upgrade step in camera auto-calibration. Unlike traditional methods which may get trapped in local minima due to the non-convex nature of these problems, this approach provides a theoretical guarantee of global optimality. This work relies on and extends recent developments in the theory of convex under-estimators and modern convex solver technology.

4.6 Cryo-Electron Microscopy

Cryo-Electron Microscopy (cryo-EM) is an emerging technique in structural biology for 3D structure (density) estimation of a specimen preserved in ice. Unlike tomography where a large number of images of a single specimen can be acquired in known orientations, the number of cryo-electron micrographs of a single particle that can be gathered is limited because radiation damages the particle during the imaging process. In Cryo-EM, the specimen consists of identical copies of the same protein macromolecule, but they are embedded at random and unknown 3-D orientations within the ice. Thus the problem is an instance of computed tomography with unknown structure and motion.

In [93] we developed a novel elementary method for solving the so-called angular reconstitution problem with a complete characterization of its degeneracy conditions. Angular reconstitution plays the same role in Cryo-EM as the eight point algorithm in multi-view geometry. This then serves as the basis of a novel sampling based method for robustly constructing an estimate of the structure of the molecule. We show significant improvement over the estimates constructed using current state of the art tools.

4.7 Image Segmentation

Image segmentation is an area of computer vision that has received much attention in recent years. The motivation being to move up from the pixel level description of an image and describe it in terms of perceptually meaningful pieces. These pieces may or may not have object level meaning.

In [1], we have shown how the problem of efficiently storing the segmentation map of an image is related to the problem of chromatic entropy minimization over graphs and how combinatorial lower bounds on chromatic entropy can be misleading. Concretely, we showed that there exist k-colorable vertex weighted graphs, for which there is a $k+1$ vertex coloring with lower chromatic entropy than the best k-coloring.

The motion of objects in a scene is a very powerful cue for understanding the structure of images. With a few exceptions most of this work on utilizing motion for segmenting images is devoted to the case of small differential motion between image frames. In [139, 140] we proposed a new motion segmentation algorithm utilizing fast graph cuts. This allowed us to construct a segmentation algorithm whose results were within a factor of two of the optimal solution independent of problem size. This work predates the current excitement around graph cut based algorithms in computer vision and graphics.

Bibliography

[1] S. Agarwal and S. Belongie. On the non-optimality of four color coding of image partitions. In *Proc. Int'l. Conf. Image Processing*, 2002. in review.

[2] S. Agarwal, K. Branson, and S. Belongie. Higher order learning with graphs. In *Proceedings of the International Conference on Machine Learning*, 2006.

[3] S. Agarwal, M. Chandraker, F. Kahl, D. Kriegman, and S. Belongie. Practical global optimization for multiview geometry. In *Proceedings of the European Conference on Computer Vision*, 2006.

[4] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. J. Kriegman, and S. Belongie. Beyond pairwise clustering. In *CVPR (2)*, pages 838–845, 2005.

[5] S. Agarwal, S. Mallick, D. Kriegman, and S. Belongie. On refractive optical flow. In *Proc. European Conf. Comput. Vision*, Prague, Czech Republic, May 2004.

[6] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. *SIGGRAPH '03*, 22(3):605–612, 2003.

[7] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5(1):13–51, 1995.

[8] C. J. Alpert and A. B. Kahng. Recent directions in netlist partitioning: A survey. *Integration: The VLSI Journal*, 19(1–2):1–81, 1995.

[9] C. J. Alpert and S.-Z. Yao. Spectral partitioning: The more eigenvectors, the better. In *Proceedings of the* 32^{nd} *ACM/IEEE Design Automation Conference*, pages 195–200, 1995.

[10] M. Ashikhmin, S. Premoze, and P. Shirley. A microfacet-based brdf generator. In *SIGGRAPH*, pages 65–74, 2000.

[11] ASTM. *E284-05a: "Standard Termninology of Appearance"*. ASTM International, 2005.

[12] F. R. Bach and M. I. Jordan. Learning spectral clustering. Technical Report UCB/CSD-03-1249, EECS Department, University of California, Berkeley, 2003.

[13] M. Bakonyi and C. Johnson. The euclidean distance matrix completion problem. *SIAM Journal on Matrix Analysis and Applications*, 16(2):646–654, 1995.

[14] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[15] H.-J. Bandelt and A. W. M. Dress. An order theoretic framework for overlapping clusters. *Discrete Mathematics*, 136:21–37, 1994.

[16] R. E. Barlow, D. Bartholomew, J. M. Bremner, and H. D. Brunk. *Statistical inference under order restrictions; The theory and application of isotonic regression.* Wiley and Sons, New York, 1972.

[17] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions? *IJCV*, 28(3):245–260, 1998.

[18] M. Belkin and P. Niyogi. Semi-supervised learning on Riemannian manifolds. *Mach. Learn.*, 56(1-3):209–239, 2003.

[19] S. Benson and Y. Ye. DSDP5 User Guide The Dual-Scaling Algorithm for Semidefinite Programming. Technical report, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, Tech. Rep. ANL/MCSTM-255, 2004.

[20] C. Berge. *Hypergraphs.* North Holland, 1973.

[21] A. Björck. *Numerical methods for Least Squares Problems.* SIAM, 1996.

[22] M. Bolla. Spectra, Euclidean representations and clusterings of hypergraphs. *Discrete Mathematics*, 117(1-3), 1993.

[23] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

[24] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications.* Springer Series in Statistics. Springer Verlag, 1997.

[25] J. Bourgain. On lipschitz embedding of finite metric spaces in hilbert space. *Israel J. Math*, 52(1):46–52, 1985.

[26] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[27] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckert-Young decomposition. *Psychometrika*, 35(3):283–319, September 1970.

[28] L. Cayton and S. Dasgupta. Robust Euclidean embedding. In *Proceedings of the International Conference on Machine Learning*, 2006.

[29] M. Chandraker, S. Agarwal, F. Kahl, D. Kriegman, and S. Belongie. Optimal metric upgrade. (in preparation).

[30] J. Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. *Problems in Analysis*, pages 195–199, 1970.

[31] F. R. Chung. The Laplacian of a hypergraph. In J. Friedman, editor, *Expanding Graphs (DIMACS series)*, pages 21–36. AMS, 1993.

[32] F. R. K. Chung. *Spectral Graph Theory.* Number 92 in CBMS Regional Conference Series in Mathematics. AMS, 1997.

[33] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *Computer Graphics (SIGGRAPH 1981)*, 15(4):187–196, 1981.

[34] T. Cox and M. Cox. *Multidimensional Scaling.* Chapman & Hall/CRC, 2000.

[35] T. Cox, M. Cox, and J. Branco. Multidimensional scaling for n-tuples. *Brit. J. Math. Stat. Psy.*, 44:195–206, 1991.

[36] J. Dattorro. *Convex Optimization and Euclidean Distance Geometry.* Meboo Publishing, USA, 2005.

[37] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Statist. Soc.*, 39 B:1–38, 1977.

[38] M.-M. Deza and I. Rosenberg. n-Semimetrics. *European Journal of Combinatorics*, 21:797–806, 2000.

[39] W. Donath and A. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–452, 1973.

[40] R. Dubes and A. Jain. *Algorithms for Clustering Data*, volume 355. Prentice Hall, 1988.

[41] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[42] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[43] M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In *Proceedings of Americal Control Conference*, June 2004.

[44] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *IEEE Conf. on Design Automation*, pages 175–181, 1982.

[45] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(100):619–633, 1975.

[46] R. W. Fleming, R. O. Dror, and E. H. Adelson. Real-world illumination and the perception of surface reflectance properties. *J. Vis.*, 3(5):347–368, 7 2003.

[47] R. Forman. Bochner's method for cell complexes and combinatorial Ricci curvature. *Discrete & Computational Geometry*, 29(3):323–374, 2003.

[48] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nystrom method. *PAMI*, 26(2):214–225, 2004.

[49] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *PAMI*, 23(6):643–660, 2001.

[50] D. Gibson, J. M. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamical systems. In *VLDB*, pages 311–322, 1998.

[51] A. V. Goldberg and S. Rao. Beyond the flow decomposition barrier. In *FOCS*, pages 2–11, 1997.

[52] V. M. Govindu. A tensor decomposition for geometric grouping and segmentation. In *CVPR*, 2005.

[53] J. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325, 1966.

[54] S. W. Hadley. Approximation techniques for hypergraph partitioning problems. *Disc. Appl. Math.*, 59(2):115–127, 1995.

[55] L. Hagen and A. Kahng. New spectral methods for ratio cut partitioning and clustering. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 11(9):1074–1085, 1992.

[56] K. M. Hall. An r-dimensional quadratic placement algorithm. *Management Sciences*, 17(3):219–229, November 1970.

[57] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming: Series A and B*, 79(1-3):191–215, 1997.

[58] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer Verlag, 2001.

[59] C. Hayashi. Two dimensional quantification based on the measure of dissimilarity among three elements. *Ann. I. Stat. Math.*, 24:251–257, 1972.

[60] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. GReenberg. A comprehensive physical model for light reflection. *Computer graphics (SIGGRAPH 1991)*, 25(4):175–186, 1991.

[61] W. J. Heiser and M. Bennani. Triadic distance models: Axiomatization and least squares representation. *J. Math. Psy.*, 41:189–206, 1997.

[62] G. Herden. Some aspects of clustering functions. *SIAM Journal on Algebraic and Discrete Methods*, 5(1):101–116, 1984.

[63] D. J. Higham and M. Kibble. A unified view of spectral clustering. Technical Report 2, University of Strathclyde, Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, UK, January 2004.

[64] N. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra Appl*, 103:103–118, 1988.

[65] N. J. Higham. Matrix nearness problems and applications. In M. J. C. Gover and S. Barnett, editors, *Applications of Matrix Theory*, pages 1–27. Oxford University Press, 1989.

[66] T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *PAMI*, 19(1):1–14, 1997.

[67] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge Univ Press, 1985.

[68] T. Hu and K. Moerder. Multiterimnal flows in hypergraphs. In T. Hu and E. S. Kuh, editors, *VLSI Circuit Layout: Theory and Design*, pages 87–93. IEEE Press, 1985.

[69] H. Huang, Z. Liang, and P. Pardalos. Some Properties for the Euclidean Distance Matrix and Positive Semidefinite Matrix Completion Problems. *Journal of Global Optimization*, 25(1):3–21, 2003.

[70] L. Hubert. Some extensions of Johnson's hierarchial clustering algorithms. *Psychometrika*, 37(3):261–274, September 1972.

[71] L. Hubert. Some applications of graph theory to clustering. *Psychometrika*, 39(3):283–309, September 1974.

[72] L. Hubert. A set-theoretical approach to the problem of hierarchial clustering. *J. Math. Psy.*, 15:70–88, 1977.

[73] N. Hughes and D. Lowe. Artefactual Structure from Least Squares Multidimensional Scaling. In *Advances in Neural Information Processing systems*, pages 937–944. MIT; 1998, 2003.

[74] R. S. Hunter and R. W. Harold. *The measurement of appearance.* Wiley, New York, 1987.

[75] E. Ihler, D. Wagner, and F. Wagner. Modeling hypergraphs by graphs with the same mincut properties. *Inform. Process. Lett.*, 45:171–175, 1993.

[76] D. Jacobs, P. Belhumeur, and R. Basri. Comparing images under variable illumination. In *CVPR*, pages 610–677, 1998.

[77] M. F. Janowitz. An order theoretic model for cluster analysis. *SIAM Journal on Applied Mathematics*, 34(1):55–72, January 1978.

[78] M. F. Janowitz. Monotone equivariant cluster methods. *SIAM Journal on Applied Mathematics*, 37(1):148–165, August 1979.

[79] N. Jardine and R. Sibson. A model for taxonomy. *Math. Biosci.*, 2:465–482, 1968.

[80] S. C. Johnson. Hierarchial clustering schemes. *Psychometrika*, 32(3):241–254, September 1967.

[81] S. Joly and G. L. Calvé. Three-way distances. *J. Classification*, 12:191–205, 1995.

[82] G. Karypis and V. Kumar. Multilevel k-way hypergraph partitioning. In *IEEE Conf. on Design Automation*, pages 343–348, 1999.

[83] M. Kendall and K. D. Gibbons. *Rank Correlation Methods.* Oxford University Press, 1990.

[84] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Sys. Tech. J.*, 49:291–307, 1970.

[85] T. G. Kolda. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255, 2001.

[86] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[87] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, 1964.

[88] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[89] M. Laurent. Polynomial instances of the positive semidefinite and euclidean distance matrix completion problems. *SIAM Journal on Matrix Analysis and Applications*, 22(3):874–894, 2000.

[90] M. Laurent, M. A connection between positive semidefinite and Euclidean distance matrix completion problems. *Linear Algebra and its Applications*, 273(1):3, 1998.

[91] W. Li and P. Solé. Spectra of regular graphs and hypergraphs and orthogonal polynomials. *European Journal of Combinatorics*, 17(5):461–477, 1996.

[92] S. P. Lloyd. Least squares quantization in PCM's. *Bell Telephone Laboratories Paper, Murray Hill, NJ*, 1957.

[93] S. P. Mallick, S. Agarwal, D. Kriegman, S. Belongie, B. Carraghar, and C. Potter. Structure and view estimation for tomographic reconstruction: A Bayesian approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2006.

[94] K. V. Mardia. *Multivariate Analysis*. Probability and Mathematical Statistics. Academic Press, 1979.

[95] J. Matousek. *Lectures on Discrete Geometry*. Springer-Verlag New York, 2002.

[96] W. Matusik. *A Data-Driven Reflectance Model*. PhD thesis, MIT, 2003.

[97] W. Matusik, H. Pfister, M. Brand, and L. McMillian. A data-driven reflectance model. In *SIGGRAPH '03*, pages 759–769, New York, NY, USA, 2003. ACM Press.

[98] M. Meilă and J. Shi. Learning segmentation with random walk. In *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, pages 873–879, 2001.

[99] J. R. Munkres. *Elements of Algebraic Topology*. The Benjamin/Cummings Publishing Company, 1984.

[100] A. Nemirovskii and Y. Nesterov. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*. SIAM, Philadelphia, 1994.

[101] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.

[102] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometric considerations and nomenclature for reflectance. Monograph 161, National Bureau of Standards (US), October 1977.

[103] F. Pellacini, J. A. Ferwerda, and D. P. Greenberg. Toward a psychophysically-based light reflection model for image synthesis. In *SIGGRAPH '00*, pages 55–64., New York, NY, USA, 2000. ACM Press.

[104] P. Perona and W. T. Freeman. A factorization approach to grouping. In *Proc. 5th Europ. Conf. Comput. Vision*, 1998.

[105] J. Pistorius and M. Minoux. An improved direct labeling method for the max-flow min-cut computation in large hypergraphs and applicatsions. *Int. Trans. Oper. Res.*, 10(1):1–11, 2003.

[106] A. Pothen, H. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, July 1990.

[107] A. Rabinovich, S. Agarwal, C. Laris, J. Price, and S. Belongie. Unsupervised spectral decomposition of histologically stained tissue samples. In *Advances in Neural Information Processing Systems 15: Proceedings of the 2003 Conference*, 2003. In review.

[108] A. Rabinovich, S. Agarwal, J. H. Price, and S. Belongie. Accuracy of unsupervised spectral decomposition for densitometr y of histological sections. (in review).

[109] B. D. Ripley. *Pattern recognition and neural networks.* Cambridge University Press, Cambridge, 1996.

[110] J. A. Rodríguez. On the Laplacian eigenvalues and metric parameters of hypergraphs. *Linear and Multilinear Algebra*, 50(1):1–14, 2002.

[111] J. A. Rodríguez. On the Laplacian spectrum and walk-regular hypergraphs. *Linear and Multilinear Algebra*, 51(3):285–297, September 2003.

[112] S. Rosenberg. *The Laplacian on a Riemannian Manifold.* London Mathematical Society, 1997.

[113] S. Sarkar and K. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. In *CVPR*, 1996.

[114] I. Schoenberg. Remarks to Maurice Frechet's article "Sur La definition Axiomatique D'une Classe D'espace Distances Vectoriellement Applicable Sur L'espace De Hilbert". *The Annals of Mathematics*, 36(3):724–732, 1935.

[115] B. Scholkopf and A. Smola. *Learning with Kernels.* Cambridge, MA: MIT Press, 2002.

[116] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2003.

[117] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[118] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and vision. In *ICML*, 2005.

[119] A. Shashua, R. Zass, and T. Hazan. Multi-way clustering using super-symmetric non-negative tensor factorization. Technical report, Hebrew University, School of Eng. and Computer Science, 2005.

[120] R. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika*, 27(2):125–140, 1962.

[121] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, August 2000.

[122] A. Smola and I. Kondor. Kernels and regularization on graphs. In B. Schölkopf and M. Warmuth, editors, *Proceedings of the Annual Conference on Computational Learning Theory*, Lecture Notes in Computer Science. Springer, 2003.

[123] J. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.

[124] J. Tenenbaum, V. de Silva, and J. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[125] K. Toh, M. Todd, and R. Tutuncu. SDPT3–a Matlab software package for semidefinite programming, version 2.1. *Optimization Methods and Software*, 11:545–581, 1999.

[126] W. Torgerson. *Theory and methods of scaling*. Wiley New York, 1958.

[127] W. Torgerson. Multidimensional scaling of similarity. *Psychometrika*, 30(4):379–393, 1965.

[128] P. H. S. Torr. Geometric motion segmentation and model selection. *Phil. Trans. of the R. Soc. A*, pages 1321–1340, 1998.

[129] K. E. Torrance and E. M. Sparrow. Theory off off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 57:1105–1114, Sept. 1967.

[130] J. Tumblin, J. K. Hodgins, and B. K. Guenter. Two methods for display of high contrast images. *ACM Trans. Graph.*, 18(1):56–94, January 1999.

[131] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *SIGGRAPH '94*, pages 311–318, 1994.

[132] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *PAMI*, 13(4):376–380, April 1991.

[133] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[134] V. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

[135] R. Vidal, Y. Ma, and J. Piazzi. A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In *CVPR*, 2004.

[136] G. J. Ward. Measuring and modelling anisotropic reflection. *Computer Graphics (SIGGRAPH 1992)*, 26(2):265–272, 1992.

[137] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Proc. 7th Int'l. Conf. Computer Vision*, pages 975–982, 1999.

[138] H. B. Westlund and G. W. Meyer. Applying appearance standards to light reflection models. In *SIGGRAPH '01*, pages 501–51., New York, NY, USA, 2001. ACM Press.

[139] J. Wills, S. Agarwal, and S. Belongie. What went where. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, volume 1, pages 37–44, 2003.

[140] J. Wills, S. Agarwal, and S. Belongie. A feature-based approach for determining dense long range correspondences. *International Journal of Computer Vision*, 68(2):125–143, June 2006.

[141] J. Wills, S. Agarwal, D. Kriegman, and S. Belongie. Toward a perceptual space for reflectance. submitted to Transactions on Graphics, 2006.

[142] S. X. Yu and J. Shi. Multiclass spectral clustering. In *Proceedings of the International Conference on Computer Vision*, 2003.

[143] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. *Advances in Neural Information Processing Systems*, 17:1601–1608, 2004.

[144] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. *Advances in Neural Information Processing Systems*, 14:1057–1064, 2002.

[145] D. Zhou, J. Huang, and B. Schölkopf. Beyond pairwise classification and clustering using hypergraphs. Technical Report 143, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 2005.

[146] D. Zhou and B. Schölkopf. Regularization on discrete spaces. *Pattern Recognition, Proceedings of the 27th DAGM Symposium*, pages 361–368, 08 2005.

[147] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin, 2005.

[148] J. Y. Zien, M. D. F. Schlag, and P. K. Chan. Multi-level spectral hypergraph partitioning with arbitrary vertex sizes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(9):1389–1399, 1999.