Spectral Curvature Clustering for Hybrid Linear Modeling

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Guangliang Chen

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

Doctor Of Philosophy

July, 2009

Spectral Curvature Clustering for Hybrid Linear Modeling

by Guangliang Chen

ABSTRACT

The problem of Hybrid Linear Modeling (HLM) is to model and segment data using a mixture of affine subspaces. Many algorithms have been proposed to solve this problem, however, probabilistic analysis of their performance is missing. In this thesis we develop the Spectral Curvature Clustering (SCC) algorithm as a combination of Govindu's multi-way spectral clustering framework (CVPR 2005) and Ng et al.'s spectral clustering algorithm (NIPS 2001) while introducing a new affinity measure. Our analysis shows that if the given data is sampled from a mixture of distributions concentrated around affine subspaces, then with high sampling probability the SCC algorithm segments well the different underlying clusters. The goodness of clustering depends on the within-cluster errors, the between-clusters interaction, and a tuning parameter applied by SCC. Supported by the theory, we then present several novel techniques for improving the performance of the algorithm. Specifically, we suggest an iterative sampling procedure to improve the existing uniform sampling strategy, an automatic scheme of inferring the tuning parameter from data, a precise initialization procedure for K-means, as well as a simple strategy for isolating outliers. The resulting algorithm requires only linear storage and takes linear running time in the size of the data. We compare it with other state-of-the-art methods on a few artificial instances of affine subspaces. Application of the algorithm to several real-world problems is also discussed.

Acknowledgements

My first and foremost thanks go to Gilad Lerman for being an extremely helpful advisor. Despite his busy schedule, Professor Lerman is always available to discuss research. He is very patient with all sorts of questions. He is also exceedingly considerate for his students. He would do everything possible to help his students grow academically. For example, he even spent much of his time going over my job search documents and gave me many valuable comments. He is undoubtedly the best advisor in all aspects and a most beneficial friend that a student can expect to find.

The members of my dissertation committee and preliminary oral exam committee, Snigdhansu Chatterjee, Dennis Cook, Peter Olver, and Fadil Santosa, have also generously given their time and provided insightful comments. I thank them for their service in the committees.

I am also grateful to many friends, colleagues, teachers, and staff in the university community who have advised, assisted, and supported my research and thesis writing. Especially, I need to express my gratitude and deep appreciation to Antoine Choffrut and Tyler Whitehouse, whose friendship, hospitality, knowledge, and wisdom have encouraged, enlightened, and entertained me during my PhD studies.

My thanks must also go to brothers and sisters in the Twin Cities Christian Assembly (TCCA) who, like a big family to me, have accompanied me through the six years at the University of Minnesota. I thank them for their constant love, strong support, and

persistent prayers. Without them my life in Minnesota would have been a much more difficult one.

I would like to finally acknowledge the firm support I received from my wife Paifang Tsai, father Datong Chen, mother Yongxia Liu, and brother Guangfa Chen while I was pursing a doctorate degree. Words are probably not enough to express my thanks to my parents who have worked very hard all their lifetime. I need to specially thank Paifang who has been my affectionate company during the dissertation writing period. Her unreserved love is always a source of strength to me.

Dedication

This dissertation is dedicated to my father Datong Chen and mother Yongxia Liu for their hard work and for loving and supporting me over the years.

Table of Contents

List of Tables

List of Figures

Introduction

This work addresses the problem of *hybrid linear modeling (HLM)*. Roughly speaking, we assume a data set that can be well approximated by a mixture of affine subspaces, or equivalently, flats, and wish to estimate the parameters of the flats as well as the membership of the given data points associated with them (see also formulations in [1] and [2]). This problem has diverse applications in many areas, such as motion segmentation in computer vision, hybrid linear representation of images, classification of face images, and temporal segmentation of video sequences (see [2] and references therein). Also, it is closely related to sparse representation and manifold learning [3, 4].

Many algorithms and strategies can be applied to this problem. For example, RANSAC [5, 6, 7], K-Flats [8, 9, 10, 11], Subspace Separation [12, 13, 14], Mixtures of Probabilistic PCA [15], Independent Component Analysis [16], Tensor Voting [17], Multi-way Clustering [18, 19, 20, 21], Generalized Principal Component Analysis [1, 2], Manifold Clustering [22], Local Subspace Affinity [23], Grassmann Clustering [24], Algebraic Multigrid [25], Agglomerative Lossy Compression [26] and Poisson Mixture Model [27]. However, we are not aware of any probabilistic analysis of the performance of such algorithms given data sampled from a corresponding hybrid linear model (with additive noise). One of the goals of this thesis is to rigorously justify a particular solution to the HLM problem.

For simplicity, we restrict the discussion to *d-flats clustering*, i.e., all the underlying

flats have the same dimension $d \geq 0$, although our theory extends to mixed dimensions by considering only the maximal dimension. We also assume here that the intrinsic dimension, d, and the number of clusters, K, are both known, and leave their estimation to future work.

Our solution to HLM, the Spectral Curvature Clustering (SCC) algorithm, follows the multi-way spectral clustering framework of Govindu [19]. This framework (when applied to HLM) starts by computing an affinity measure quantifying d-dimensional flatness for any $d+2$ points of the data. It then forms pairwise weights by decomposing the corresponding $(d+2)$-way affinity tensor. At last, it suggests to apply spectral clustering (e.g., [28]) with the pairwise weights.

However, the above steps are based only on heuristic arguments [19], with no formal justification for them. Also, there are critical numerical issues associated with Govindu's framework that need to be thoroughly addressed. First of all, as the size of data and the intrinsic dimension d increase, it is computationally prohibitive to calculate or store, not to mention process, the affinity tensor. Approximating this tensor by a small subset of uniformly sampled "fibers" [19] is insufficient for large d and data of moderate size. Better numerical techniques have to be developed while maintaining both reasonable performance and fast speed. Secondly, the multi-way affinities contain a tuning parameter, which sensitively affects clustering. It is not clear how to select its optimal value while avoiding an exhaustive search. Last of all, there are also smaller issues, e.g., how to deal with outliers.

Our algorithm, Spectral Curvature Clustering (SCC), combines Govindu's framework [19] and Ng et al.'s spectral clustering algorithm [29], while introducing *the polar tensor* (defined in Section 1.3). We justify the algorithm following the strategy of [29] in two steps. First, we consider in Chapter 3 a general affinity tensor instead of the polar tensor, and control the "goodness of clustering" of SCC by the deviation of the affinity

tensor from an ideal tensor. Next, in Chapter 4 we show that for the specific choice of the polar tensor and data sampled from a hybrid linear model, the SCC algorithm clusters the data well with high sampling probability. In addition, we express the goodness of clustering in terms of the within-cluster errors (which depend directly on the flatness of the underlying measures), the between-clusters interaction (which depends on the separation of the measures), and a tuning parameter applied by TSCC.

The SCC algorithm also provides solutions to the above-mentioned numerical issues. More specifically, it contributes to the advancement of multi-way spectral clustering in the following aspects.

- It introduces an iterative sampling procedure to significantly improve accuracy over the standard random sampling scheme used in [19] (see Section 5.1.1).
- It suggests an automatic way of selecting the tuning parameter that is commonly used in multi-way spectral clustering methods (see Section 5.1.2).
- It employs an efficient way of applying K-means in its setting (see Section 5.1.3).
- It proposes a simple strategy to isolate outliers while clustering flats (see Section 5.4).

The rest of the thesis is organized as follows. In Chapter 1 we review some theoretical background. In particular, we formulate more precisely the problem of hybrid linear modeling, introduce the polar curvature and at last form the affinity tensor. In Chapter 2 we present the theoretical version of the SCC algorithm as a combination of Govindu's framework [19] and Ng et al.'s algorithm [29] while using the specific polar curvatures. Chapters 3 and 4 analyze the performance of the TSCC algorithm. Chapter 3 presents main technical estimates for a large class of affinity tensors while quantifying fundamental notions, in particular, the goodness of clustering. Chapter 4

assumes a hybrid linear probabilistic model and the use of the polar tensor, and relates the estimates of Chapter 3 to the sampling distribution of the model. Chapter 5 introduces various techniques that are used to make the theoretical version practical, and the SCC algorithm is formulated incorporating those techniques. We compare our algorithm with other competing methods using various kinds of artificial data sets as well as several real-world applications in Chapter 6. Chapter 7 concludes with a brief discussion and possible avenues for future work. Mathematical proofs are provided in Appendix A.

Chapter 1

Background

In this chapter we present some background material that is necessary for the subsequent development of the thesis. We first define the problem of hybrid linear modeling in a theoretical setting (Section 1.1), then introduce a class of curvatures, in particular, the polar curvature, for measuring the flatness of a simplex (Section 1.2), and finally form affinity tensors and their matrix representations (Section 1.3).

Notation and Basic Definitions

Throughout this paper we assume an ambient space $\mathbb{R}^D$ and a collection of d-flats, i.e., d-dimensional flats, that are embedded in $\mathbb{R}^D$, with $0 \leq d < D$.

We denote scalars with possibly large values by upper-case plain letters (e.g., N, C), and scalars with relatively small values by lower-case Greek letters (e.g., α, ε); vectors by boldface lower-case letters (e.g., $\mathbf{u}, \mathbf{v}$); matrices by boldface upper-case letters (e.g., $\mathbf{A}$); tensors by calligraphic capital letters (e.g., $\mathcal{A}$); and sets by upper-case Roman letters (e.g., X).

For any integer $n > 0$, we denote the n-dimensional vector of ones by $\mathbf{1}_n$, and the

$n \times n$ matrix of ones by $\mathbf{1}_{n\times n}$. The $n \times n$ identity matrix is written as $\mathbf{I}_n$.

The (i, j)-element of a matrix $\mathbf{A}$ is denoted by A_{ij}, and the $(i_1, \ldots, i_n)$-element of an n-way tensor $\mathcal{A}$ by $\mathcal{A}(i_1, \ldots, i_n)$. We denote the transpose of a matrix $\mathbf{A}$ by $\mathbf{A}'$ and that of a vector $\mathbf{v}$ by $\mathbf{v}'$. The Frobenius norm of a matrix/tensor, denoted by $\|\cdot\|_{\mathrm{F}}$, is the ℓ_2 norm of the quantity when viewed as a vector.

For a positive semidefinite square matrix $\mathbf{A}$, we use $E_n(\mathbf{A})$ to denote the subspace spanned by the top n eigenvectors of $\mathbf{A}$, and $P^n(\mathbf{A})$ to represent the orthogonal projector onto $E_n(\mathbf{A})$.

Let $\mathbf{x} \in \mathbb{R}^D$ and F be a d-flat in $\mathbb{R}^D$. We denote the orthogonal distance from $\mathbf{x}$ to F by $\mathrm{dist}(\mathbf{x}, F)$. For any $r > 0$, the ball centered at $\mathbf{x}$ with radius r is written as $\mathrm{B}(\mathbf{x}, r)$. If $c > 0$, then $c \cdot \mathrm{B}(\mathbf{x}, r) := \mathrm{B}(\mathbf{x}, c \cdot r)$. If S is a subset of $\mathbb{R}^D$, we denote its diameter by $\mathrm{diam}(\mathrm{S})$ and its complement by S^c. If S is furthermore discrete, we use $|\mathrm{S}|$ to denote its number of elements.

Let μ be a measure on $\mathbb{R}^D$. We denote the support of μ by $\mathrm{supp}(\mu)$, its restriction to a given set S by $\mu|_{\mathrm{S}}$, and the product measure of n copies of μ by μ^n. The d-dimensional Lebesgue measure is denoted by $\mathcal{L}_d$. Also, we use $(\mathbb{R}^D)^n$ to denote the Cartesian product of n copies of $\mathbb{R}^D$.

We use $\mathrm{P}(n, r)$ to denote the number of permutations of size r from a sequence of n available elements. That is, $\mathrm{P}(n, r) := n(n-1)\cdots(n-r+1)$.

1.1 The Problem of Hybrid Linear Modeling

We formulate here a version of the HLM problem. We will introduce further restrictions on its setting throughout the paper. Before presenting the problem we need to define the notions of *d-dimensional least squares errors and flats.*

If μ is a Borel probability measure, then the *least squares error* of approximating μ

by a d-flat is denoted by $e_2(\mu)$ and defined as follows:

$$e_2(\mu) := \sqrt{\inf_{d\text{-flats } F} \int \operatorname{dist}^2(\mathbf{x}, F)\, \mathrm{d}\mu(\mathbf{x})}. \tag{1.1}$$

Any minimizer of the above quantity is referred to as a *least squares d-flat*.

We now incorporate the above definitions and present the problem of hybrid linear modeling below.

Problem 1. *Let $\mu_1, \ldots, \mu_K$ be Borel probability measures and assume that their d-dimensional least square errors $\{e_2(\mu_k)\}_{k=1}^K$ are sufficiently small and that their least squares d-flats do not coincide. Suppose a data set $\mathrm{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^D$ generated as follows: For each k, N_k points are sampled independently and identically from μ_k, so that $N = N_1 + \cdots + N_K$. The goal of hybrid linear modeling is to partition X into K subsets representing the underlying d-flats and simultaneously estimate the parameters of the underlying flats.*

We remark that the above notion of sufficiently small least square errors combined with non-coinciding least squares d-flats is quantified for our particular solution later in Section 4.2 (by restricting the size of the constant α of equation (4.5)). We also remark that we will restrict in Section 1.2 the above setting by requiring the measures $\mu_1, \ldots, \mu_K$ to be "regular and possibly d-separated" (see Remark 1.2.5) and later in Section 3.2 by imposing the comparability of sizes of $N_1, \ldots, N_K$ (see equation (3.4)).

1.2 The Polar Curvature

For any $d+2$ distinct points $\mathbf{z}_1, \ldots, \mathbf{z}_{d+2} \in \mathbb{R}^D$, we denote by $V_{d+1}(\mathbf{z}_1, \ldots, \mathbf{z}_{d+2})$ the $(d+1)$-volume of the $(d+1)$-simplex formed by these points. The polar sine at each vertex $\mathbf{z}_i, 1 \leq i \leq d+2$, is

$$\operatorname{psin}_{\mathbf{z}_i}(\mathbf{z}_1, \ldots, \mathbf{z}_{d+2}) := \frac{(d+1)! \cdot V_{d+1}(\mathbf{z}_1, \ldots, \mathbf{z}_{d+2})}{\prod_{1 \leq j \leq d+2,\, j \neq i} \|\mathbf{z}_j - \mathbf{z}_i\|_2}. \tag{1.2}$$

Definition 1.2.1. The polar curvature of $\mathbf{z}_1, \ldots, \mathbf{z}_{d+2}$ is

$$c_{\mathrm{p}}(\mathbf{z}_1, \ldots, \mathbf{z}_{d+2}) := \operatorname{diam}(\{\mathbf{z}_1, \ldots, \mathbf{z}_{d+2}\}) \cdot \sqrt{\sum_{i=1}^{d+2} \operatorname{psin}^2_{\mathbf{z}_i}(\mathbf{z}_1, \ldots, \mathbf{z}_{d+2})}. \tag{1.3}$$

Remark 1.2.2. The notion of *curvature* here designates a function of $d+2$ variables generalizing the distance function. Indeed, when $d = 0$, the polar curvature coincides with the Euclidean distance. We use this name (and probably abuse it) due to the comparability when $d = 1$ of the polar curvature with the Menger curvature multiplied by the square of the corresponding diameter (see [30]).

Let μ be a Borel probability measure on $\mathbb{R}^D$. We define the polar curvature of the measure μ to be

$$c_{\mathrm{p}}(\mu) := \sqrt{\int c^2_{\mathrm{p}}(\mathbf{z}_1, \ldots, \mathbf{z}_{d+2}) \, \mathrm{d}\mu(\mathbf{z}_1) \ldots \mathrm{d}\mu(\mathbf{z}_{d+2})}. \tag{1.4}$$

The polar curvatures of randomly sampled $(d+1)$-simplices can be used to estimate the least squares errors of approximating certain probability measures by d-flats. We start with two preliminary definitions and then state the main result, which is proved in [31] (following the methods of [32, 30, 33]).

Definition 1.2.3. We say that a Borel probability measure μ on $\mathbb{R}^D$ is *d-separated* (with parameters $0 < \delta, \omega < 1$) if there exist $d+2$ balls $\{B_i\}_{i=1}^{d+2}$ in $\mathbb{R}^D$ with μ-measures at least δ such that

$$V_d(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{d+1}}) > \omega \cdot \operatorname{diam}(\operatorname{supp}(\mu))^d, \tag{1.5}$$

for any $\mathbf{x}_{i_k} \in 2B_{i_k}, 1 \leq k \leq d+1$ and $1 \leq i_1 < \cdots < i_{d+1} \leq d+2$.

Definition 1.2.4. We say that a Borel probability measure μ on $\mathbb{R}^D$ is *regular* (with parameters C_μ and γ) if there exist constants $\gamma > 2$ and $C_\mu \geq 1$ such that for any $\mathbf{x} \in \operatorname{supp}(\mu)$ and $0 < r \leq \operatorname{diam}(\operatorname{supp}(\mu))$:

$$\mu(B(\mathbf{x}, r)) \leq C_\mu \cdot r^\gamma. \tag{1.6}$$

If $D = 2$ (or $\text{supp}(\mu)$ is contained in a 2-flat), then one can allow $1 < \gamma \leq 2$ while strengthening the above equation as follows:

$$C_\mu^{-1} \cdot r^\gamma \leq \mu(B(\mathbf{x}, r)) \leq C_\mu \cdot r^\gamma. \tag{1.7}$$

Theorem 1.2.1. *For any regular and d-separated Borel probability measure μ there exists a constant C (depending only on the d-separation parameters, i.e., ω, δ, and the regularity parameters, i.e., γ, C_μ) such that*

$$C^{-1} \cdot e_2(\mu) \leq c_{\text{p}}(\mu) \leq C \cdot e_2(\mu). \tag{1.8}$$

The following two curvatures also satisfy Theorem 1.2.1 [31]:

$$c_{\text{dls}}(\mathbf{z}_1, \ldots, \mathbf{z}_{d+2}) := \sqrt{\inf_{d-\text{flats } F} \sum_{i=1}^{d+2} \text{dist}^2(\mathbf{z}_i, F)}, \tag{1.9}$$

$$c_{\text{h}}(\mathbf{z}_1, \ldots, \mathbf{z}_{d+2}) := \min_{1 \leq i \leq d+2} \text{dist}(\mathbf{z}_i, F_{(i)}), \tag{1.10}$$

where $F_{(i)}$ is the $(d-1)$-flat spanned by all the $d+2$ points except $\mathbf{z}_i$. In this paper we use c_{p} as a representative of the class of curvatures that satisfy Theorem 1.2.1, since it seems computationally faster than the above two (using the numerical framework described later in Section 5.3). However, all the theory developed in this paper applies to the rest of the class.

Remark 1.2.5. Since we will use Theorem 1.2.1 in Section 4.3 to justify our proposed solution to HLM, we need to assume that the measures $\mu_1, \ldots, \mu_K$ of Problem 1 are regular and d-separated. However, those restrictions could be relaxed or avoided as follows. If either c_{dls} or c_{h} is used instead of c_{p}, then Theorem 1.2.1 holds for merely d-separated probability measures (no need for regularity). Moreover, in Section 4.3 we may only use the right hand side of equation (1.8), i.e., the upper bound of $c_{\text{p}}(\mu)$ in terms of $e_2(\mu)$ (though it is preferable to have a tight estimate as suggested by the

full equation). For such a bound it is enough to assume that μ is merely a regular probability measure. If we use instead of c_{p} any of the curvatures c_{dls}, c_{h}, then this upper bound holds for any Borel probability measure. We also comment that the regularity conditions described in Definition 1.2.4 could be further relaxed when replacing $\mathrm{diam}(\{\mathbf{z}_1, \ldots, \mathbf{z}_{d+2}\})$ in equation (1.3) with e.g., a geometric mean of corresponding edge lengths. More details appear in [31].

1.3 Affinity Tensors and their Matrix Representations

Throughout the rest of this paper, we consider $(d+2)$-way tensors of the form

$$\{\mathcal{A}(i_1, \ldots, i_{d+2})\}_{1 \le i_1, \ldots, i_{d+2} \le N}.$$

We assume that their elements are between zero and one, and invariant under arbitrary permutations of the indices $(i_1, \ldots, i_{d+2})$, i.e., these tensors are super-symmetric.

Most commonly, we form the following affinities using the polar curvature (see equation (1.3)):

$$\mathcal{A}_{\mathrm{p}}(i_1, \ldots, i_{d+2}) := \begin{cases} e^{-c_{\mathrm{p}}(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{d+2}})/\sigma}, & \text{if } \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{d+2}} \text{ are distinct;} \\ 0, & \text{otherwise.} \end{cases} \tag{1.11}$$

The corresponding tensor $\mathcal{A}_{\mathrm{p}}$ is referred to as *the polar tensor*.

In the special case of underlying linear subspaces (instead of general affine ones), we may work with the following $(d+1)$-way tensor:

$$\mathcal{A}_{\mathrm{p,L}}(i_1, \ldots, i_{d+1}) := \begin{cases} e^{-c_{\mathrm{p}}(\mathbf{0}, \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{d+1}})/\sigma}, & \text{if } \mathbf{0}, \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{d+1}} \text{ are distinct;} \\ 0, & \text{otherwise.} \end{cases} \tag{1.12}$$

In most of the paper we use the $(d+2)$-tensor $\mathcal{A}_{\mathrm{p}}$, while in a few places we refer to the $(d+1)$-tensor $\mathcal{A}_{\mathrm{p,L}}$.

Given a $(d+2)$-way affinity tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times \cdots \times N}$ we unfold it into an $N \times N^{d+1}$ matrix $\mathbf{A}$ in a similar way as in [34, 35]. The i-th row of $\mathbf{A}$ contains all the elements in the i-th "slice" of $\mathcal{A}$: $\{\mathcal{A}(i, i_2, \ldots, i_{d+2}) \mid 1 \leq i_2, \ldots, i_{d+2} \leq N\}$, according to an arbitrary but fixed ordering of the last $d+1$ indices $(i_2, \ldots, i_{d+2})$, e.g., the lexicographic ordering. This ordering (when fixed for all rows) is not important to us, since we are only interested in the uniquely determined matrix $\mathbf{W} := \mathbf{A} \cdot \mathbf{A}'$ (see Algorithm 1 below).

Chapter 2

Theoretical Spectral Curvature Clustering (TSCC)

We combine Govindu's framework of multi-way spectral clustering [19] and Ng et al.'s spectral clustering algorithm [29] while incorporating the polar affinities (equation (1.11)), to formulate below (Algorithm 1) the Theoretical Spectral Curvature Clustering (TSCC) algorithm for solving Problem 1.

Algorithm 1: Theoretical Spectral Curvature Clustering (TSCC)

Input : $\mathrm{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\} \subset \mathbb{R}^D$: data set, d: common dimension of flats, K: number of d-flats, σ: the tuning parameter for computing $\mathcal{A}$

Output: K disjoint clusters $\mathrm{C}_1, \ldots, \mathrm{C}_K$

begin

1. Construct the polar tensor $\mathcal{A}_{\mathrm{p}}$ using equation (1.11) and the given σ
2. Unfold $\mathcal{A}_{\mathrm{p}}$ to obtain the affinity matrix $\mathbf{A}$, and form the weight matrix $\mathbf{W} := \mathbf{A} \cdot \mathbf{A}'$
3. Compute the degree matrix $\mathbf{D} := \mathrm{diag}\{\mathbf{W} \cdot \mathbf{1}_N\}$, and use it to normalize $\mathbf{W}$ to get $\mathbf{Z} := \mathbf{D}^{-1/2} \cdot \mathbf{W} \cdot \mathbf{D}^{-1/2}$
4. Find the top K eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K$ of $\mathbf{Z}$ and define $\mathbf{U} := [\mathbf{u}_1 \mathbf{u}_2 \ldots \mathbf{u}_K] \in \mathbb{R}^{N \times K}$
5. (optional) Normalize the rows of $\mathbf{U}$ to have unit length or using other methods (see Section 3.3.1)
6. Apply K-means [36] to the rows of $\mathbf{U}$ to find K clusters, and partition the original data into K subsets $\mathrm{C}_1, \ldots, \mathrm{C}_K$ accordingly

end

The performance of the TSCC algorithm is evaluated by computing two types of errors: $e_{\mathrm{OLS}}, e_{\%}$. For any K detected clusters $\mathrm{C}_1, \ldots, \mathrm{C}_K$, the total (squared) Orthogonal Least Squares (OLS) error is defined as follows:

$$e_{\mathrm{OLS}} = \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} \mathrm{dist}^2(\mathbf{x}, F_k), \tag{2.1}$$

where F_k is the OLS d-flat approximating C_k (can be obtained by Principal Component Analysis (PCA)). In situations where we know the true membership of the data points,

we also compute the percentage of misclassified points. That is,

$$e_{\%} = \frac{\#\text{ of misclassified points}}{N} \cdot 100\%. \tag{2.2}$$

We refer to the above algorithm as theoretical because its complexity and storage requirement can be rather large (even though polynomial). In Chapter 5 we develop various numerical techniques to make the algorithm practical. In particular, we suggest a sampling strategy to approximate the matrix $\mathbf{W}$ in an iterative way, an automatic scheme of tuning the parameter σ, and a straightforward procedure to initialize K-means for clustering the rows of $\mathbf{U}$.

The TSCC algorithm can be seen as two steps of embedding data followed by K-means. First, each data point $\mathbf{x}_i$ is mapped to $\mathbf{A}(i,:)$, the i-th row of the matrix $\mathbf{A}$, which contains the interactions between the point $\mathbf{x}_i$ and all d-flats spanned by any $d+1$ points in the data (indeed, each column corresponds to $d+1$ data points). Second, $\mathbf{x}_i$ is further mapped to the i-th row of the matrix $\mathbf{U}$. The rows of $\mathbf{U}$ are treated as points in $\mathbb{R}^K$, to which K-means is applied.

The question of whether or not to normalize the rows of the matrix $\mathbf{U}$ is an interesting one. For ease of the subsequent theoretical development, we do not normalize the rows of $\mathbf{U}$. Such a choice is also adopted in Chapter 5 where the practical implementation of the TSCC algorithm yields good numerical results. In Section 3.3.1 we discuss more carefully the normalization of the matrix $\mathbf{U}$ and show the advantage of such practice.

We remark that one can replace the polar tensor (applied in Step 1 of Algorithm 1) with other affinity tensors, based on the polar curvature or other ones that satisfy Theorem 1.2.1, to form different versions of TSCC. For example, when the underlying subspaces are known to be linear, one may use the $(d+1)$-tensor $\mathcal{A}_{\mathrm{p,L}}$ of equation (1.12), forming the Theoretical Linear Spectral Curvature Clustering (TLSCC) algorithm. Another example is the following class of affinity tensors that are based on the powers of

the polar curvature:

$$\mathcal{A}_{\mathrm{p},q}(i_1,\ldots,i_{d+2}) := \begin{cases} e^{-\frac{c_{\mathrm{p}}^q\left(\mathbf{x}_{i_1},\ldots,\mathbf{x}_{i_{d+2}}\right)}{\sigma^q}}, & \text{if } \mathbf{x}_{i_1},\ldots,\mathbf{x}_{i_{d+2}} \text{ are distinct;} \\ 0, & \text{otherwise,} \end{cases} \tag{2.3}$$

where $q \geq 1$ (see Remark 4.3.1 for interpretation). While Algorithm 1 uses $q = 1$, its practical version, Algorithm 2, uses $q = 2$ for faster convergence.

We justify the TSCC algorithm in two steps. In Chapter 3 we analyze the TSCC algorithm with a very general tensor (replacing the polar tensor), and develop conditions under which TSCC is expected to work well. In particular, the corresponding analysis applies to the polar tensor. Chapter 4 relates this analysis to the sampling of Problem 1, and correspondingly formulates a probabilistic statement for TSCC with its own polar tensor.

Chapter 3

Perturbation Analysis of TSCC

Following a strategy of Ng et al. [29], we analyze the performance of the TSCC algorithm with a general affinity tensor (replacing the polar tensor in Step 1 of Algorithm 1) in two steps. First, we define a "perfect" tensor representing the ideal affinities, and show that in such a hypothetical situation, the K underlying clusters are correctly separated by the TSCC algorithm. Next, we assume that TSCC is applied with a general affinity tensor, and control the goodness of clustering of TSCC by the deviation of the given tensor from the perfect tensor. Finally, we discuss the effect of the two normalizations in the TSCC algorithm (Steps 3 and 5 of Algorithm 1).

Notational Convenience

We maintain the common setting of Problem 1 and all the notation used in the TSCC algorithm.

We denote the K underlying clusters by $\widetilde{\mathrm{C}}_1, \ldots, \widetilde{\mathrm{C}}_K$. Each $\widetilde{\mathrm{C}}_k$ has N_k points, so that $N = \sum_{1 \le k \le K} N_k$. For ease of presentation we suppose that $N_1 \le N_2 \le \cdots \le N_K$, and that the points in X are ordered according to their membership. That is, the first

N_1 points of X are in $\widetilde{\mathrm{C}}_1$, the next N_2 points in $\widetilde{\mathrm{C}}_2$, etc..

We define K index sets $\mathrm{I}_1, \ldots, \mathrm{I}_K$ having the indices of the points in $\widetilde{\mathrm{C}}_1, \ldots, \widetilde{\mathrm{C}}_K$ respectively, that is,

$$\mathrm{I}_k := \{n \in \mathbb{N} \mid \sum_{1 \le j \le k-1} N_j < n \le \sum_{1 \le j \le k} N_j\}, \quad \text{for each } 1 \le k \le K. \tag{3.1}$$

We let $\mathbf{u}^{(i)}$, $1 \le i \le N$, denote the i-th row of $\mathbf{U}$ and $\mathbf{c}^{(k)}$, $1 \le k \le K$, denote the center of the k-th cluster, i.e.,

$$\mathbf{c}^{(k)} := \frac{1}{N_k} \sum_{j \in \mathrm{I}_k} \mathbf{u}^{(j)}. \tag{3.2}$$

3.1 Analysis of TSCC with the Perfect Tensor

We define here the notion of a perfect tensor and show that TSCC obtains a perfect segmentation with such a tensor.

Definition 3.1.1. The *perfect tensor* associated with Problem 1 is defined as follows. For any $1 \le i_1, \ldots, i_{d+2} \le N$,

$$\widetilde{\mathcal{A}}(i_1, \ldots, i_{d+2}) := \begin{cases} 1, & \text{if } \mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{d+2}} \text{ are distinct and in the same } \widetilde{\mathrm{C}}_k; \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

We designate quantities derived from the perfect tensor $\widetilde{\mathcal{A}}$ (by following the TSCC algorithm) with the tilde notation, e.g., $\widetilde{\mathbf{A}}, \widetilde{\mathbf{W}}, \widetilde{\mathbf{D}}, \widetilde{\mathbf{Z}}, \widetilde{\mathbf{U}}$.

Remark 3.1.2. When $d = 0$, the perfect tensor $\widetilde{\mathcal{A}}$ reduces to a block diagonal matrix, with the blocks corresponding to the underlying clusters. Ng et al. [29] also considered an ideal affinity matrix with a block diagonal structure. However, they maintained the diagonal blocks that are computed from the data, while we assume a more extreme case in which the elements of these blocks are identically one (except at the diagonal entries).

With our assumption it is possible to follow the steps of TSCC and exactly compute each quantity.

Our result for TSCC with the perfect tensor $\widetilde{\mathcal{A}}$ is formulated as follows (see proof in Appendix A.1).

Proposition 3.1.1. *If $N_k > d+2$ for all $k = 1, \ldots, K$, then*

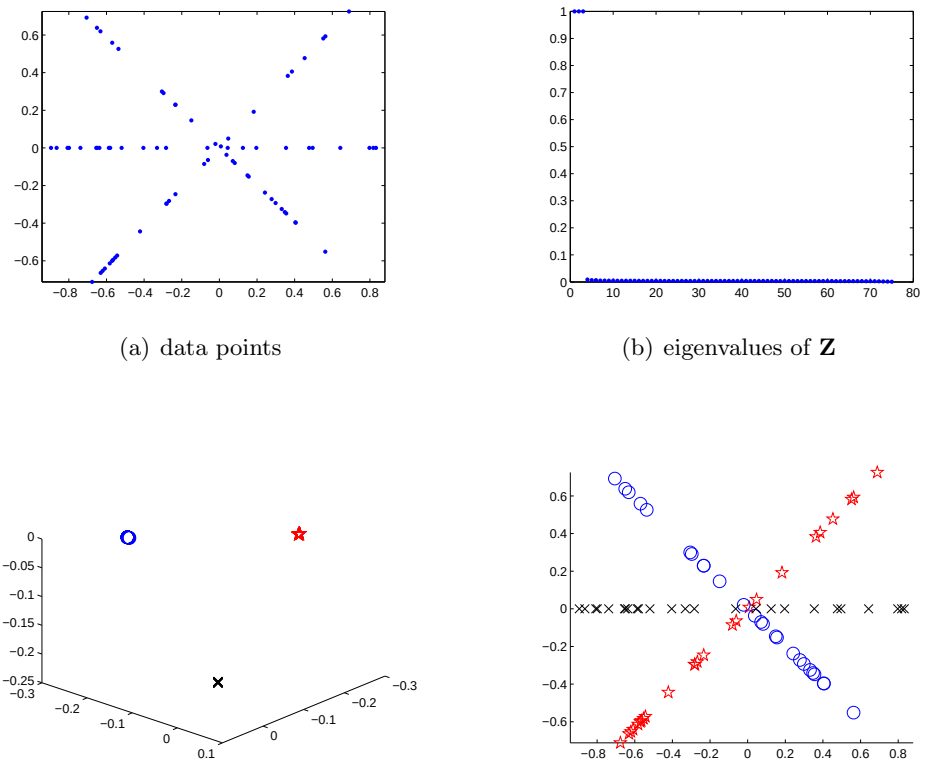
1. *$\widetilde{\mathbf{Z}}$ has exactly K eigenvalues of one; the rest are $\frac{d+1}{(N_k-1)(N_k-d-1)}, 1 \leq k \leq K$, each replicated $N_k - 1$ times.*

2. *The rows of $\widetilde{\mathbf{U}}$ are K mutually orthogonal vectors in $\mathbb{R}^K$. Moreover, each vector corresponds to a distinct underlying cluster.*

Remark 3.1.3. For the TLSCC algorithm, the corresponding perfect tensor $\widetilde{\mathcal{A}}_{\mathrm{L}}$ is a $(d+1)$-dimensional equivalent of the $(d+2)$-way tensor $\widetilde{\mathcal{A}}$ of equation (3.3). Proposition 3.1.1 still holds for $\widetilde{\mathcal{A}}_{\mathrm{L}}$ but with d replaced by $d-1$.

Example 3.1.4. Illustration of the perfect tensor analysis: We randomly generate three clean linear lines in $\mathbb{R}^2$ and then sample 25 points from each line (see Figure 3.1(a)). We then apply TSCC with the polar tensor of equation (1.11) and $\sigma = .00001$. The corresponding tensor is a close approximation to the perfect tensor, because taking the limit of equation (1.11) as $\sigma \to 0+$ essentially yields the perfect tensor. Intermediate and final clustering results are reported in Figures 3.1(b)-3.1(d).

In this case, the top three eigenvalues are hardly distinguished from 1, and the rest are close to zero (see Figure 3.1(b)). The rows of $\mathbf{U}$ accumulate at three orthogonal vectors (see Figure 3.1(c)), and thus form three tight clusters, each representing an underlying line (see Figure 3.1(d)).



(a) data points

(b) eigenvalues of $\mathbf{Z}$

(c) rows of $\mathbf{U}$

(d) detected clusters

Figure 3.1: Illustration of the perfect tensor analysis

3.2 Perturbation Analysis of TSCC with a General Affinity Tensor

We assume that the underlying clusters have comparable and adequate sizes, more precisely, there exists a constant $0 < \varepsilon_1 \leq 1$ such that

$$N_k \geq \max\left(\varepsilon_1 \cdot N/K, 2d+3\right), \quad k = 1, \ldots, K. \tag{3.4}$$

We also assume that all the affinity tensors $\mathcal{A}$ considered in this section are supersymmetric, and with elements between 0 and 1. Moreover, they satisfy the following condition.

Assumption 1. There exists a constant $\varepsilon_2 > 0$ such that

$$\mathbf{D} \geq \varepsilon_2 \cdot \widetilde{\mathbf{D}}. \tag{3.5}$$

Remark 3.2.1. We feel the need to have some lower bound on $\mathbf{D}$, possibly even weaker than that of Assumption 1, to ensure that the TSCC algorithm would work well. Indeed, for each $i \in \mathrm{I}_k, 1 \leq k \leq K$, the sum $\sum_{j \in \mathrm{I}_k} W_{ij}$ measures the "connectedness" between the point $\mathbf{x}_i$ and the other points in $\widetilde{\mathrm{C}}_k$, and thus should be sufficiently large. Accordingly, since $D_{ii} \geq \sum_{j \in \mathrm{I}_k} W_{ij}$, these diagonal entries of the matrix $\mathbf{D}$ should be correspondingly large as well. In Section 4.4 we discuss the existence of this condition for the polar tensor while taking into account the restrictions on the tuning parameter σ implied by Theorem 4.2.1.

3.2.1 Measuring Goodness of Clustering of the TSCC Algorithm

We use two equivalent ways to quantify the goodness of clustering of the TSCC algorithm when applied with a general affinity tensor $\mathcal{A}$. In Section 3.3.1 we relate them to the more absolute notion of "clustering identification error".

We first investigate each of the K underlying clusters in the $\mathbf{U}$ space, i.e., $\{\mathbf{u}^{(i)}\}_{i\in \mathrm{I}_k}, 1 \leq k \leq K$, and estimate the sum of their variances. We refer to this sum as the *total variation* of the matrix $\mathbf{U}$.

Definition 3.2.2. The total variation of $\mathbf{U}$ (with respect to the K underlying clusters) is

$$\mathrm{TV}(\mathbf{U}) := \sum_{1\leq k\leq K} \sum_{i\in \mathrm{I}_k} \left\| \mathbf{u}^{(i)} - \mathbf{c}^{(k)} \right\|_2^2, \tag{3.6}$$

where $\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(K)}$ are the centers of the underlying clusters in the $\mathbf{U}$ space (see equation (3.2)).

The smaller the total variation $\mathrm{TV}(\mathbf{U})$ is, the more concentrated the underlying clusters in the $\mathbf{U}$ space are. In fact, the following lemma (proved in Appendix A.3) implies that the smaller $\mathrm{TV}(\mathbf{U})$ is, the more separated the centers are from the origin and from each other.

Lemma 3.2.1.

$$\sum_{1\leq k\leq K} N_k \cdot \left\| \mathbf{c}^{(k)} \right\|_2^2 = K - \mathrm{TV}(\mathbf{U}), \tag{3.7}$$

$$\sum_{1\leq k<\ell\leq K} N_k N_\ell \cdot \langle \mathbf{c}^{(k)}, \mathbf{c}^{(\ell)} \rangle^2 \leq \mathrm{TV}(\mathbf{U})\,. \tag{3.8}$$

The other measurement of the goodness of clustering of TSCC is motivated by the fact that, in the ideal case, the subspace spanned by the top K eigenvectors of $\widetilde{\mathbf{Z}}$, $E_K(\widetilde{\mathbf{Z}})$, leads to a perfect segmentation (see Proposition 3.1.1). When given a general affinity tensor $\mathcal{A}$, the eigenspace $E_K(\mathbf{Z})$ determines the clustering result of TSCC. We thus suggest to measure the discrepancy between these two eigenspaces, $E_K(\mathbf{Z})$ and $E_K(\widetilde{\mathbf{Z}})$, by comparing the orthogonal projectors onto them, $P^K(\mathbf{Z})$ and $P^K(\widetilde{\mathbf{Z}})$, in the following way.

Definition 3.2.3. The distance between the two subspaces $E_K(\widetilde{\mathbf{Z}})$ and $E_K(\mathbf{Z})$ is

$$\operatorname{dist}(E_K(\mathbf{Z}), E_K(\widetilde{\mathbf{Z}})) := \left\| P^K(\mathbf{Z}) - P^K(\widetilde{\mathbf{Z}}) \right\|_{\mathrm{F}}. \tag{3.9}$$

A geometric interpretation of the above distance is provided using the notion of *principal angles* [37]. The principal angles $0 \le \theta_1 \le \cdots \le \theta_K \le \pi/2$ between two K-dimensional subspaces S and T are defined recursively as follows (see e.g., [37]):

$$\cos\theta_1 = \max_{\mathbf{x}\in S, \|\mathbf{x}\|_2=1} \max_{\mathbf{y}\in T, \|\mathbf{y}\|_2=1} \mathbf{x}'\mathbf{y} = \mathbf{x}_1'\mathbf{y}_1, \tag{3.10}$$

$$\cos\theta_2 = \max_{\substack{\mathbf{x}\in S, \|\mathbf{x}\|_2=1 \\ \mathbf{x}\perp\mathbf{x}_1}} \max_{\substack{\mathbf{y}\in T, \|\mathbf{y}\|_2=1 \\ \mathbf{y}\perp\mathbf{y}_1}} \mathbf{x}'\mathbf{y} = \mathbf{x}_2'\mathbf{y}_2, \tag{3.11}$$

$$\cdots\cdots$$

$$\cos\theta_K = \max_{\substack{\mathbf{x}\in S, \|\mathbf{x}\|_2=1 \\ \mathbf{x}\perp\{\mathbf{x}_1,\ldots,\mathbf{x}_{K-1}\}}} \max_{\substack{\mathbf{y}\in T, \|\mathbf{y}\|_2=1 \\ \mathbf{y}\perp\{\mathbf{y}_1,\ldots,\mathbf{y}_{K-1}\}}} \mathbf{x}'\mathbf{y} = \mathbf{x}_K'\mathbf{y}_K. \tag{3.12}$$

Another formula for the cosines of the principal angles is obtained in the following way. Let $\mathbf{S}$ and $\mathbf{T}$ be two matrices whose columns define orthonormal bases of S and T respectively. Since any $\mathbf{x} \in S$ and $\mathbf{y} \in T$ can be represented as $\mathbf{x} = \mathbf{S}\cdot\mathbf{u}$ and $\mathbf{y} = \mathbf{T}\cdot\mathbf{v}$ respectively, where $\mathbf{u}$ and $\mathbf{v}$ are unit vectors in $\mathbb{R}^K$, it follows that

$$\cos\theta_k = \sigma_k\left(\mathbf{S}'\cdot\mathbf{T}\right) \quad \text{for } 1 \le k \le K, \tag{3.13}$$

where $\sigma_k(\cdot)$ denotes the k-th largest singular value of the matrix.

We present the geometric interpretation in Lemma 3.2.2 and prove it in Appendix A.4.

Lemma 3.2.2. *Let $0 \le \theta_1 \le \theta_2 \le \cdots \le \theta_K \le \pi/2$ be the K principal angles between the two subspaces $E_K(\mathbf{Z})$ and $E_K(\widetilde{\mathbf{Z}})$. Then*

$$\operatorname{dist}^2(E_K(\mathbf{Z}), E_K(\widetilde{\mathbf{Z}})) = 2\cdot\sum_{k=1}^{K}\sin^2\theta_k. \tag{3.14}$$

At last, we claim that the above two ways of measuring the goodness of clustering of TSCC are equivalent in the following sense (see proof in Appendix A.2).

Lemma 3.2.3.

$$\mathrm{dist}^2(E_K(\mathbf{Z}), E_K(\widetilde{\mathbf{Z}})) = 2 \cdot \mathrm{TV}(\mathbf{U})\,. \tag{3.15}$$

3.2.2 The Perturbation Result

Given a general affinity tensor $\mathcal{A}$ we quantify its deviation from the perfect tensor $\widetilde{\mathcal{A}}$ by the difference

$$\mathcal{E} := \mathcal{A} - \widetilde{\mathcal{A}}. \tag{3.16}$$

Our main result shows that the magnitude of this perturbation controls the goodness of clustering of the TSCC algorithm.

Theorem 3.2.4. *Let $\mathcal{A}$ be any affinity tensor satisfying Assumption 1 and $\mathcal{E}$ its deviation from the perfect tensor. There exists a constant $C_1 = C_1(K, d, \varepsilon_1, \varepsilon_2)$ (estimated in equation* (A.49) *of Appendix A.5) such that if*

$$N^{-(d+2)} \|\mathcal{E}\|_{\mathrm{F}}^2 \leq \frac{1}{8C_1}, \tag{3.17}$$

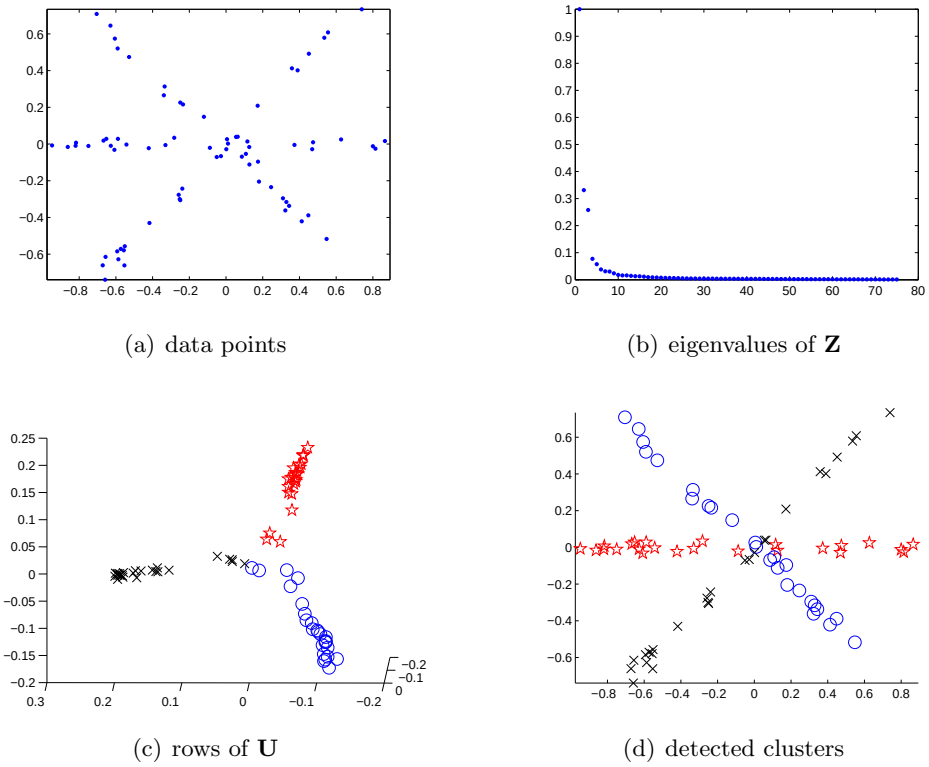
then

$$\mathrm{TV}(\mathbf{U}) \leq C_1 \cdot N^{-(d+2)} \|\mathcal{E}\|_{\mathrm{F}}^2\,. \tag{3.18}$$

Remark 3.2.4. For the TLSCC algorithm, Theorem 3.2.4 holds with d replaced by $d-1$.

Example 3.2.5. Illustration of the perturbation analysis: We corrupt the data in Figure 3.1 with 2.5% additive Gaussian noise (see Figure 3.2(a)), and apply TSCC with the polar tensor of equation (1.11) and $\sigma = 0.1840$. In this case of moderate noise, the top three eigenvalues are still clearly separated from the rest, even though two of them deviate from 1 (see Figure 3.2(b)). The rows of $\mathbf{U}$ still form three well separated clusters, but they deviate from concentrating at exactly three orthogonal vectors (see

Figure 3.2(c)). The underlying clusters are detected correctly, except possibly for a few points at their intersection (see Figure 3.2(d)).



(a) data points

(b) eigenvalues of $\mathbf{Z}$

(c) rows of $\mathbf{U}$

(d) detected clusters

Figure 3.2: Illustration of the perturbation analysis

3.3 The Effects of the Normalizations in TSCC

3.3.1 Possible Normalizations of U and Their Effects on Clustering

The analysis of the previous sections uses the embedding represented by the rows of $\mathbf{U}$. It is possible to normalize these rows (e.g., by their lengths as in [29]) before applying K-means. In the following we consider two normalized versions of the rows of $\mathbf{U}$, and

analyze their effects on the TSCC algorithm (in comparison with the rows of $\mathbf{U}$).

Using the cluster sizes, or the row lengths, one could normalize the matrix $\mathbf{U}$ and obtain two matrices $\mathbf{T}, \mathbf{V}$ whose rows are defined as follows:

$$\mathbf{t}^{(i)} = \sqrt{N_k} \cdot \mathbf{u}^{(i)}, \quad i \in \mathrm{I}_k, 1 \leq k \leq K; \tag{3.19}$$

$$\mathbf{v}^{(i)} = \frac{1}{\left\|\mathbf{u}^{(i)}\right\|_2} \cdot \mathbf{u}^{(i)}, \quad 1 \leq i \leq N. \tag{3.20}$$

These two normalizations are explained as follows. The $\mathbf{V}$ normalization discards all the magnitude information of the rows of $\mathbf{U}$ to contain only the angular information between them. The $\mathbf{T}$ normalization, containing the same angular information, reduces to $\mathbf{U}$ when $N_1 = \cdots = N_K = N/K$, and otherwise tries to further separate the underlying clusters by scaling the rows using the cluster sizes. See Figure 3.3(a) for an illustration of the $\mathbf{U}$, $\mathbf{T}$, $\mathbf{V}$ spaces.

Remark 3.3.1. The normalization $\mathbf{T}$ assumes knowledge of the underlying cluster sizes, but can be effectively approximated without this knowledge when using the practical version of TSCC, SCC (see Algorithm 2). The SCC algorithm employs an iterative sampling procedure which converges quickly, thus it can estimate $\mathbf{T}$ in the current iteration by using the clusters obtained in the previous iteration.

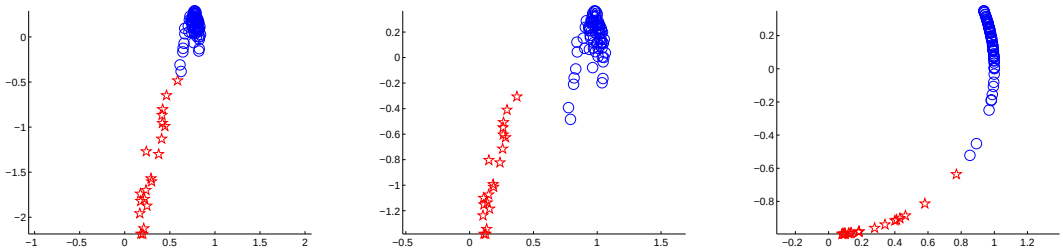
We view the matrix $\mathbf{V}$ as a weak approximation to $\mathbf{T}$. Indeed, in the ideal case they coincide, since

$$\left\|\widetilde{\mathbf{u}}^{(i)}\right\|_2 = \frac{1}{\sqrt{N_k}}, \quad i \in \mathrm{I}_k, 1 \leq k \leq K \tag{3.21}$$

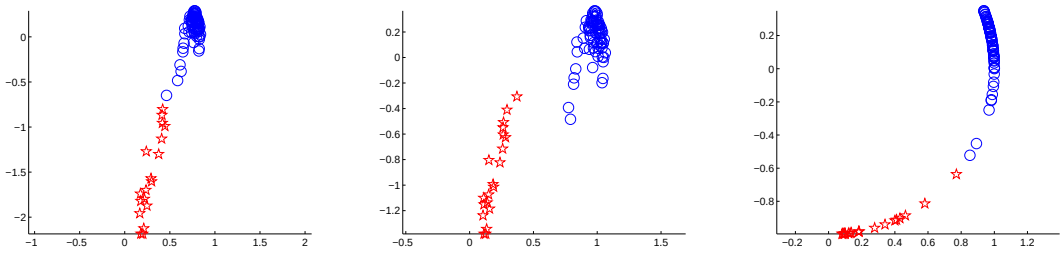
(see equation (A.10)). In the general case, the above equality only holds on average. More precisely, the orthonormality of $\mathbf{U}$ implies that

$$\sum_{k=1}^{K} \sum_{i \in \mathrm{I}_k} \left\|\mathbf{u}^{(i)}\right\|_2^2 = \|\mathbf{U}\|_{\mathrm{F}}^2 = \sum_{j=1}^{K} \|\mathbf{u}_j\|_2^2 = K. \tag{3.22}$$

We next define two criterions for analyzing the performance of $\mathbf{U}$, $\mathbf{T}$ and $\mathbf{V}$ when directly applying K-means to them.



(a) The underlying clusters in the $\mathbf{U}, \mathbf{T}, \mathbf{V}$ spaces respectively



(b) The clusters found by K-means in the $\mathbf{U}, \mathbf{T}, \mathbf{V}$ spaces

Figure 3.3: The underlying clusters and those found by K-means in the $\mathbf{U}$, $\mathbf{T}$, $\mathbf{V}$ spaces. The given data consists of 80 and 20 points on two lines in $\mathbb{R}^2$. We note that, in order for the rows of $\mathbf{U}$ to have similar magnitudes to those of $\mathbf{T}$ and $\mathbf{V}$, we have scaled each row of $\mathbf{U}$ by the square root of the average cluster size $\sqrt{N/K}$.

First, we define a notion of *the separation factor* for the centers of the underlying clusters in each of the $\mathbf{U}$, $\mathbf{T}$ and $\mathbf{V}$ spaces. The separation factor of the centers in the $\mathbf{U}$ space is defined as follows:

$$\beta(\mathbf{U}) := \frac{\sum_{1\le i<j\le K}\langle \mathbf{c}^{(i)}, \mathbf{c}^{(j)}\rangle^2}{\left(\sum_{1\le k\le K}\left\|\mathbf{c}^{(k)}\right\|_2^2\right)^2}. \tag{3.23}$$

The separation factors $\beta(\mathbf{T}), \beta(\mathbf{V})$ are defined similarly. The smaller β is, the more separated in $\mathbb{R}^K$ the centers of the underlying clusters are. Lemma 3.2.1 directly implies that $\beta(\mathbf{T})$ is controlled by $\mathrm{TV}(\mathbf{U})$ as follows.

Lemma 3.3.1.

$$\beta(\mathbf{T}) \le \frac{\mathrm{TV}(\mathbf{U})}{(K-\mathrm{TV}(\mathbf{U}))^2}. \tag{3.24}$$

We note that $\beta(\mathbf{U}) = \beta(\mathbf{T})$ when $N_k = N/K, k = 1, \ldots, K$. In general, we observe that $\beta(\mathbf{U}) \le \beta(\mathbf{T}) \le \beta(\mathbf{V})$, with the former two being fairly close. For example, $\beta(\mathbf{U}) = .0004, \beta(\mathbf{T}) = .0006, \beta(\mathbf{V}) = .0032$ in Figure 3.3(a). In practice, however, we have found that the underlying clusters in the $\mathbf{U}, \mathbf{T}, \mathbf{V}$ spaces are usually not closely concentrated around their centers, thus this criterion is not sufficient.

Second, we define a notion of the *clustering identification error* in the $\mathbf{U}$, $\mathbf{T}$ and $\mathbf{V}$ spaces respectively. For ease of discussion, we suppose that $K = 2$. In the $\mathbf{U}$ space, the corresponding error has the form:

$$e_{\mathrm{id}}(\mathbf{U}) := \frac{1}{N}\cdot\sum_{k=1,2}\#\left\{i\in \mathrm{I}_k \mid \left\|\mathbf{u}^{(i)}-\mathbf{c}^{(k)}\right\|_2 \ge 1/2\cdot\left\|\mathbf{c}^{(1)}-\mathbf{c}^{(2)}\right\|_2\right\} \tag{3.25}$$

The errors in the $\mathbf{T}, \mathbf{V}$ spaces are defined similarly. The following lemma (proved in Appendix A.6) shows that both $e_{\mathrm{id}}(\mathbf{T})$ and $e_{\mathrm{id}}(\mathbf{U})$ can be controlled by $\mathrm{TV}(\mathbf{U})$, with the former having a smaller upper bound.

Lemma 3.3.2. *Suppose that $K = 2$. If*

$$\mathrm{TV}(\mathbf{U}) < \left(\sqrt{3}-1\right)^2, \tag{3.26}$$

then the identification error in the $\mathbf{T}$ *space is bounded above as follows:*

$$e_{\mathrm{id}}(\mathbf{T}) \leq \frac{4 \cdot \mathrm{TV}(\mathbf{U})}{2 - \mathrm{TV}(\mathbf{U}) - 2\sqrt{\mathrm{TV}(\mathbf{U})}}. \tag{3.27}$$

If

$$\mathrm{TV}(\mathbf{U}) < \left(\sqrt{2 + \frac{4}{\varepsilon_1^2}} - \frac{2}{\varepsilon_1}\right)^2, \tag{3.28}$$

then the identification error in the $\mathbf{U}$ *space is bounded above as follows:*

$$e_{\mathrm{id}}(\mathbf{U}) \leq \frac{4 \cdot \mathrm{TV}(\mathbf{U})}{2 - \mathrm{TV}(\mathbf{U}) - 4/\varepsilon_1 \cdot \sqrt{\mathrm{TV}(\mathbf{U})}}, \tag{3.29}$$

where the constant ε_1 *is defined in equation* (3.4).

We remark that the clustering identification errors $e_{\mathrm{id}}(\mathbf{U}), e_{\mathrm{id}}(\mathbf{T}), e_{\mathrm{id}}(\mathbf{V})$ have only theoretical meanings. However, they can be used to estimate the clustering errors of K-means when applied in the $\mathbf{U}, \mathbf{T}, \mathbf{V}$ spaces respectively. We observed in practice that $e_{\mathrm{id}}(\mathbf{T})$ and $e_{\mathrm{id}}(\mathbf{V})$ are often very close.

Following the above discussion we think that $\mathbf{T}$ is probably the right normalization to be used in TSCC. Its practical implementation should follow Remark 3.3.1. We note that the application of this normalization in Lemma 3.2.1 results in analogous estimates for the $\mathbf{T}$ space which are independent of the sizes of clusters. Indeed, this normalization seems to outperform $\mathbf{U}$ when $N_1, \ldots, N_K$ vary widely (this claim is supported in practice by numerical experiments and in theory by Lemma 3.3.2). Another reason for our preference of $\mathbf{T}$ is that performing K-means in the $\mathbf{T}$ space is equivalent to performing weighted K-means (with weights $N_k/N, 1 \leq k \leq K$) in the $\mathbf{U}$ space, which allows small clusters to have relatively larger variances (see e.g., Figure 3.3(a)).

The $\mathbf{V}$ normalization is another possibility to use in TSCC. On one hand, it is a weak approximation to $\mathbf{T}$; on the other hand, it contains only the angular information of the rows of $\mathbf{U}$. The use of only angular information for K-means clustering, partly supported by the polarization theorem in [38], seems to also separate the underlying

clusters further. However, we need to understand this normalization more thoroughly, i.e., in terms of theoretical analysis.

In Chapters 5 and 6 we will use $\mathbf{U}$ to demonstrate our numerical strategies, though they also apply to $\mathbf{T}$ and $\mathbf{V}$.

3.3.2 TSCC Without Normalizing W

We analyze here the TSCC algorithm when the matrix $\mathbf{W}$ is not normalized, i.e., skipping Step 3 of Algorithm 1 and letting $\mathbf{Z} := \mathbf{W}$. We refer to the corresponding variant of TSCC as TSCC-UN, and formulate below analogous results of Proposition 3.1.1 and Theorem 3.2.4. The proof of Proposition 3.3.3 directly follows that of Proposition 3.1.1 in Appendix A.1 (in particular, equations (A.2) and (A.3)). Theorem 3.3.4 is proved in Appendix A.7.

Proposition 3.3.3. *Suppose that the TSCC-UN algorithm is applied with the perfect tensor $\widetilde{\mathbf{A}}$. Then*

1. *The eigenvalues of $\widetilde{\mathbf{W}}$ are $\widetilde{d}_K \geq \cdots \geq \widetilde{d}_2 \geq \widetilde{d}_1$ (each of multiplicity 1), and $\widetilde{\nu}_K \geq \cdots \geq \widetilde{\nu}_2 \geq \widetilde{\nu}_1$ (of multiplicity $N_K, \ldots, N_2, N_1$ respectively), where*

$$\widetilde{d}_k := (N_k - d - 1) \cdot \mathrm{P}(N_k - 1, d + 1), \tag{3.30}$$

$$\widetilde{\nu}_k := (d + 1) \cdot \mathrm{P}(N_k - 2, d). \tag{3.31}$$

2. *If $\widetilde{d}_1 > \widetilde{\nu}_K$, the rows of $\widetilde{\mathbf{U}}$ are exactly K mutually orthogonal vectors, each representing a distinct underlying cluster.*

Theorem 3.3.4. *Suppose that TSCC-UN is applied with a general affinity tensor $\mathcal{A}$, and that*

$$N \geq \sqrt{2(d+1)\left(1 - \frac{K-1}{K}\varepsilon_1\right)^d \left(\frac{2K}{\varepsilon_1}\right)^{d+2}}, \tag{3.32}$$

Let

$$C_2(K, d, \varepsilon_1, \varepsilon_2) := 32 \left(\frac{2K}{\varepsilon_1} \right)^{2(d+2)}. \tag{3.33}$$

If

$$N^{-(d+2)} \|\mathcal{E}\|_{\mathrm{F}}^2 \leq \frac{1}{8C_2}, \tag{3.34}$$

then

$$\mathrm{TV}(\mathbf{U}) \leq C_2 \cdot N^{-(d+2)} \|\mathcal{E}\|_{\mathrm{F}}^2 . \tag{3.35}$$

In view of equation (3.32), the TSCC-UN algorithm seems to require large data size in order to work well. Numerical experiments also indicate that this approach is very sensitive to the variation of cluster sizes, and works consistently worse than the normalized approach, i.e., TSCC. Our current analysis, however, does not manifest the significant advantage of the normalized approach. We thus leave the related exploration to later research.

Von Luxburg et al. [39] have shown that in the framework of kernel spectral clustering, the normalized method is consistent under very general conditions. On the other hand, the unnormalized method is only consistent under very specific conditions that are rarely met in practice. Since $\mathbf{W}$ can be seen as a kernel matrix, [39] provides another evidence for our preference of the normalized approach.

Chapter 4

Probabilistic Analysis of TSCC

In this chapter we analyze the performance of the TSCC algorithm with its own affinity tensor, i.e., the polar tensor of equation (1.11). We control with high sampling probability the goodness of clustering of TSCC when applied to the data generated in Problem 1.

4.1 Basic Setting and Definitions

We follow the setting of hybrid linear modeling described in Problem 1 together with the assumptions of regularity and possibly d-separation of $\{\mu_i\}_{i=1}^K$ (see Remark 1.2.5) as well as the restriction imposed by equation (3.4). We denote the corresponding N random variables by $\mathfrak{X}_1, \ldots, \mathfrak{X}_N \in \mathbb{R}^D$ and maintain the previous notation for their sampled values $\mathbf{x}_1, \ldots, \mathbf{x}_N$. The joint sample space is $(\mathbb{R}^D)^N$, and the corresponding joint probability measure is

$$\mu_{\mathrm{p}} := \mu_1^{N_1} \times \cdots \times \mu_K^{N_K}. \tag{4.1}$$

We introduce an incidence constant reflecting the separation between the measures $\mu_1, \ldots, \mu_K$ in regard to the polar curvature c_{p} and the tuning parameter σ. We first

define the following sets

$$\mathrm{S}_k := (\mathrm{supp}(\mu_k))^{d+2}\,,\ 1 \leq k \leq K. \tag{4.2}$$

Then, given a constant $\sigma > 0$, the incidence constant has the form:

$$C_{\mathrm{in}}(\mu_1, \ldots, \mu_K; \sigma) := \max_{\substack{1 \leq k_1, \ldots, k_{d+2} \leq K \\ \text{not all equal}}} \int_{\mathrm{S}_{k_1}} \cdots \int_{\mathrm{S}_{k_{d+2}}} e^{\frac{-c_{\mathrm{p}}(\mathbf{z}_1, \ldots, \mathbf{z}_{d+2})}{\sigma}} \, \mathrm{d}\mu_{k_1}(\mathbf{z}_1) \ldots \, \mathrm{d}\mu_{k_{d+2}}(\mathbf{z}_{d+2}), \tag{4.3}$$

where the maximum is taken over all $1 \leq k_1, \ldots, k_{d+2} \leq K$ except $k_1 = k_2 = \cdots = k_{d+2}$.

Remark 4.1.1. For TLSCC, the incidence constant is defined as follows:

$$C_{\mathrm{in,L}}(\mu_1, \ldots, \mu_K; \sigma) := \max_{\substack{1 \leq k_1, \ldots, k_{d+1} \leq K \\ \text{not all equal}}} \int_{\mathrm{S}_{k_1}} \cdots \int_{\mathrm{S}_{k_{d+1}}} e^{\frac{-c_{\mathrm{p}}(\mathbf{0}, \mathbf{z}_1, \ldots, \mathbf{z}_{d+1})}{\sigma}} \, \mathrm{d}\mu_{k_1}(\mathbf{z}_1) \ldots \, \mathrm{d}\mu_{k_{d+1}}(\mathbf{z}_{d+1}). \tag{4.4}$$

We note that for both TSCC and TLSCC, the incidence constant is between 0 and 1. The smaller the incidence constant is, the more separated (in terms of the polar curvature and the tuning parameter) the measures are. In Section 4.3 we estimate the incidence constant in a few special instances of hybrid linear modeling.

4.2 The Probabilistic Result

The following theorem (proved in Appendix A.10) shows that, when the underlying measures are sufficiently flat and well separated from each other, with high probability (with respect to the sampling of Problem 1) the TSCC algorithm segments the K underlying clusters well.

Theorem 4.2.1. *Suppose that the TSCC algorithm is applied to the data generated in Problem 1 with a tuning parameter $\sigma > 0$. Let*

$$\alpha := \frac{1}{\sigma^2} \sum_{k=1}^{K} c_{\mathrm{p}}^2(\mu_k) + C_{\mathrm{in}}(\mu_1, \ldots, \mu_K; \sigma/2), \tag{4.5}$$

and $C_1 = C_1(K, d, \varepsilon_1, \varepsilon_2)$ *be the constant defined in Theorem 3.2.4. If*

$$\alpha < \frac{1}{16 C_1}, \tag{4.6}$$

then

$$\mu_{\mathrm{p}}\left(\mathrm{TV}(\mathbf{U}) \le 2\alpha \cdot C_1 \mid \text{Assumption } 1 \text{ holds}\right) \ge 1 - e^{-2N\alpha^2/(d+2)^2}. \tag{4.7}$$

Remark 4.2.1. Theorem 4.2.1 also holds for the TLSCC algorithm, but with d replaced by $d-1$, and the constant α by

$$\alpha_{\mathrm{L}} := \frac{1}{\sigma^2} \sum_{k=1}^{K} c_{\mathrm{p,L}}^2(\mu_k) + C_{\mathrm{in,L}}(\mu_1, \ldots, \mu_K; \sigma/2), \tag{4.8}$$

where for any Borel probability measure μ,

$$c_{\mathrm{p,L}}(\mu) := \sqrt{\int c_{\mathrm{p}}^2(\mathbf{0}, \mathbf{z}_1, \ldots, \mathbf{z}_{d+1})\, \mathrm{d}\mu(\mathbf{z}_1) \ldots \mathrm{d}\mu(\mathbf{z}_{d+1})}. \tag{4.9}$$

Remark 4.2.2. A similar version of Theorem 4.2.1 holds for general affinity tensors of the form $\{e^{-c(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{d+2}})/\sigma}\}_{1 \le i_1, \ldots, i_{d+2} \le N}$, where c is a nonnegative, symmetric function defined on $\mathbb{R}^{d+2}$. The significance of using the polar curvature, or any other curvature satisfying Theorem 1.2.1, is explained in Section 4.3.

We showed in Lemma 3.3.2 that the clustering identification errors $e_{\mathrm{id}}(\mathbf{U})$ and $e_{\mathrm{id}}(\mathbf{T})$ can be controlled by $\mathrm{TV}(\mathbf{U})$ when $K = 2$. Combining Lemma 3.3.2 and Theorem 4.2.1 yields the following probabilistic statement.

Corollary 4.2.2. *Suppose that* $K = 2$*, and that* α, C_1 *are the constants defined in Theorem 4.2.1. If*

$$\alpha < \frac{1}{16 C_1}, \tag{4.10}$$

then

$$\begin{aligned} \mu_{\mathrm{p}}\left(e_{\mathrm{id}}(\mathbf{T}) \le \frac{4\,\alpha\, C_1}{1 - \alpha\, C_1 - \sqrt{2\,\alpha\, C_1}} \mid \text{Assumption } 1 \text{ holds}\right) \\ \ge 1 - e^{-2N\alpha^2/(d+2)^2}. \end{aligned} \tag{4.11}$$

If

$$\alpha < \frac{1}{2C_1} \cdot \min\left(\frac{1}{8}, \left(\sqrt{2+\frac{4}{\varepsilon_1^2}} - \frac{2}{\varepsilon_1}\right)^2\right), \tag{4.12}$$

then

$$\begin{aligned}\mu_{\mathrm{p}}\left(e_{\mathrm{id}}(\mathbf{U}) \leq \frac{4\,\alpha\,C_1}{1-\alpha\,C_1 - 2/\varepsilon_1 \cdot \sqrt{2\,\alpha\,C_1}} \mid \text{Assumption } \textit{1} \text{ holds}\right) \\ \geq 1 - e^{-2N\alpha^2/(d+2)^2}.\end{aligned} \tag{4.13}$$

4.3 Interpretation of the Constant α

Theorem 4.2.1 shows the strong effect of the constant α on the goodness of clustering of the TSCC algorithm. This constant has two parts, which are explained respectively as follows.

Theorem 1.2.1 implies that the first part of α is comparable to

$$\frac{1}{\sigma^2} \cdot \sum_{k=1}^{K} e_2^2(\mu_k). \tag{4.14}$$

We thus view the first part as the sum of the within-cluster errors of the model scaled by σ^2.

Remark 4.3.1. A similar interpretation applies to the tensors defined in equation (2.3). In this case, for any $q \geq 1$, the first term of α is replaced by

$$\frac{1}{\sigma^2} \sum_{k=1}^{K} c_{\mathrm{p}}^{(2q)}(\mu_k), \tag{4.15}$$

where for any Borel probability measure μ,

$$c_{\mathrm{p}}^{(2q)}(\mu) := \int c_{\mathrm{p}}^{2q}(\mathbf{z}_1, \ldots, \mathbf{z}_{d+2})\ \mathrm{d}\mu(\mathbf{z}_1) \ldots \mathrm{d}\mu(\mathbf{z}_{d+2}). \tag{4.16}$$

The above sum is then comparable to

$$\frac{1}{\sigma^2} \cdot \sum_{k=1}^{K} e_{2q}^{2q}(\mu_k), \tag{4.17}$$

where $e_{2q}(\mu_k)$ is the error of approximating μ_k by a d-flat while minimizing the L_{2q} norm [31].

We interpret the second part of α, i.e., the incidence constant, as the between-clusters interaction of the model. Unlike the first part, we do not have a theoretical result that fully establishes this interpretation. We show in a few special cases (with underlying linear subspaces) how to control this constant.

In the first example (Example 4.3.2) we estimate the incidence constant for two orthogonal line segments when using TSCC. The next three examples assume the use of the TLSCC algorithm. In Example 4.3.3 the model includes distributions along two clean line segments with an arbitrary angle θ between them. We establish the dependence of the incidence constant on θ and σ. In Example 4.3.4 we consider two orthogonal lines with uniform noise around them, and demonstrate the dependence of the incidence constant on the level of the noise and σ. Example 4.3.5 considers two clean orthogonal planes in $\mathbb{R}^3$.

Example 4.3.2. (**TSCC: two orthogonal clean lines**). We consider the following two orthogonal line segments in $\mathbb{R}^2$:

$$\mathrm{L1}: \ y = 0, \quad 0 \le x \le L,$$

and

$$\mathrm{L2}: \ x = 0, \quad 0 \le y \le L,$$

in which $L > 0$ is a fixed constant. We assume arclength measures $\mu_1 = \frac{\mathrm{d}x}{L}, \mu_2 = \frac{\mathrm{d}y}{L}$ supported on L1 and L2 respectively. For any $\sigma > 0$, the incidence constant for TSCC is bounded as follows (see Appendix A.11):

$$C_{\mathrm{in}}(\mu_1, \mu_2; \sigma) \le \frac{\sigma}{\sqrt{2}L} \left(1 - e^{-\sqrt{2}L/\sigma}\right). \tag{4.18}$$

Example 4.3.3. (**TLSCC: two intersecting clean lines**). We consider the following two lines in $\mathbb{R}^2$:

$$\mathrm{L1}: \ y = 0, \quad 0 \le x \le L,$$

and

$$\mathrm{L2}: \ y = r\sin\theta,\, x = r\cos\theta, \quad 0 \le r \le L,$$

in which $L > 0$ and $0 < \theta \le \pi/2$ are fixed constants. We assume arclength measures $\mu_1 = \frac{\mathrm{d}x}{L}, \mu_2 = \frac{\mathrm{d}r}{L}$ supported on L1 and L2 respectively. For any $\sigma > 0$, the incidence constant for TLSCC is bounded as follows (see Appendix A.12):

$$C_{\mathrm{in,L}}(\mu_1, \mu_2; \sigma) \le 2\left(\frac{\sigma}{L\sin\theta}\right)^2 \cdot \left(1 - e^{-\frac{L\sin\theta}{\sigma}}\left(1 + \frac{L\sin\theta}{\sigma}\right)\right). \tag{4.19}$$

We note that when $\theta = \pi/2$, $C_{\mathrm{in,L}}$ has a faster decay rate than C_{in} (see Example 4.3.2).

Example 4.3.4. (**TLSCC: two orthogonal rectangles**). We consider two rectangular strips in $\mathbb{R}^2$ determined by the following vertices respectively:

$$\mathrm{R1}: \ (\epsilon, 0), (L+\epsilon, 0), (\epsilon, \epsilon), (L+\epsilon, \epsilon),$$

and

$$\mathrm{R2}: \ (0, \epsilon), (0, L+\epsilon), (\epsilon, \epsilon), (\epsilon, L+\epsilon),$$

in which $0 < \epsilon < L$. We assume uniform measures $\mu_i = \frac{1}{L\epsilon}\mathcal{L}_2$ restricted to Ri, $i = 1, 2$. We view R1 and R2 as two lines surrounded by uniform noise. Let $\omega := L/\epsilon$. For any $\sigma > 0$, the incidence constant for TLSCC has the following upper bound (see Appendix A.13)

$$C_{\mathrm{in,L}}(\mu_1, \mu_2; \sigma) \le \frac{\sqrt{\sigma}}{\omega^2} + \frac{2\sqrt[4]{\sigma}}{\omega} \cdot e^{-1/\left(2\sigma^{3/4}\right)} + e^{-1/\sigma^{3/4}}. \tag{4.20}$$

In the limiting case of $\epsilon \to 0+$, i.e., when having two orthogonal lines with practically no noise, the above estimate decays faster to zero than the one in Example 4.3.3 with

$\theta = \pi/2$. This is due to the fact that in the current example we exclude the intersection of the two lines for any $\epsilon > 0$. As it turned out, the limit of the corresponding integral (as $\epsilon \to 0+$) is not the same as the full integral of this limit.

Example 4.3.5. (**TLSCC: two perpendicular clean half-disks**). We consider the following portions of two unit disks (in polar coordinates) in $\mathbb{R}^3$:

$$\mathrm{D1}: \ x = 0,\, y = \rho\cos\varphi,\, z = \rho\sin\varphi, \quad 0 \le \rho \le 1, 0 \le \varphi \le \pi,$$

and

$$\mathrm{D2}: \ x = r\cos\theta,\, y = r\sin\theta,\, z = 0, \quad 0 \le r \le 1, -\pi/2 \le \theta \le \pi/2.$$

We also assume uniform measures $\mu_i = \frac{2}{\pi}\mathcal{L}_2$ restricted on Di, $i = 1, 2$. In this case, the incidence constant for TLSCC is bounded above by the following quantity (see Appendix A.14)

$$C_{\mathrm{in,L}}(\mu_1, \mu_2; \sigma) \le \frac{8\sqrt{\sigma}}{\pi^2} + \frac{8\sqrt[4]{\sigma}}{\pi} + \frac{4\sigma^2}{(\sin\sqrt[4]{\sigma})^4}. \tag{4.21}$$

4.4 On the Existence of Assumption 1

The theory developed in this paper assumes that all affinity tensors used with TSCC, in particular the polar tensor, satisfy Assumption 1. We present some partial results regarding the existence of this assumption for the polar tensor while taking into account the restrictions on the size of σ imposed by Theorem 4.2.1. We remark that those results also extend to some other tensors.

We first show in the following lemma (proved in Appendix A.8) that if a data set is sampled from a hybrid linear model without noise, then Assumption 1 is always satisfied with the constant $\varepsilon_2 = 1$.

Lemma 4.4.1. *If the TSCC is applied to data sampled from a mixture of clean d-flats, then*

$$\mathbf{D} \ge \widetilde{\mathbf{D}}. \tag{4.22}$$

For more general data sampled from a hybrid linear model, we obtain the following estimate in expectation (see proof in Appendix A.9).

Lemma 4.4.2. *If the TSCC is applied to data sampled according to Problem 1, then Assumption 1 holds in expectation in the following sense:*

$$E_{\mu_{\mathrm{p}}}(\mathbf{D}) \geq \varepsilon_2 \cdot \widetilde{\mathbf{D}}, \tag{4.23}$$

where

$$\varepsilon_2 = e^{-\frac{2}{\sigma} \cdot \max_{1 \leq k \leq K} c_{\mathrm{p}}(\mu_k)}. \tag{4.24}$$

Remark 4.4.1. We do not expect Assumption 1 to hold with high probability (i.e., having the μ_{p} measure close to one) while maintaining the constant ε_2 formulated in Lemma 4.4.2. However, it seems reasonable to have a statement in high probability when replacing the polar curvature $c_{\mathrm{p}}(\mu_k)$ used in defining this constant with their following upper bounds:

$$\widehat{c}^{\,2}_{\mathrm{p}}(\mu_k) = \max_{\mathbf{z}_1 \in \mathrm{supp}(\mu_k)} \int c^2_{\mathrm{p}}(\mathbf{z_1}, \mathbf{z}_2, \ldots, \mathbf{z}_{d+2}) \, \mathrm{d}\mu_k(\mathbf{z}_2) \ldots \mathrm{d}\mu_k(\mathbf{z}_{d+2}) \, . \tag{4.25}$$

We leave the investigation of such a statement and the effect of using $\widehat{c}^{\,2}_{\mathrm{p}}(\mu)$ instead of $c^2_{\mathrm{p}}(\mu)$ to future research.

Chapter 5

The SCC Algorithm

The TSCC algorithm cannot be directly performed in practice due to its high complexity. In this chapter we first introduce several numerical techniques (in Section 5.1) to make the TSCC algorithm practical and then form the SCC algorithm (in Section 5.2). We next analyze the complexity of the SCC algorithm in terms of both storage and running time (in Section 5.3), and finally propose two more strategies: one for isolating outliers (in Section 5.4), and the other for segmenting flats of mixed dimensions (in Section 5.5).

5.1 The Novel Methods of SCC

5.1.1 Iterative Sampling

The TSCC algorithm is not applicable in practice for two reasons: First, the amount of space for storing the affinity matrix $\mathbf{A} \in \mathbb{R}^{N \times N^{d+1}}$ can be huge ($O(N^{d+2})$); Second, full computation of $\mathbf{A}$ and multiplication of this large matrix and its transpose (to produce $\mathbf{W}$) can be computationally prohibitive. One solution might be to use uniform sampling, i.e., randomly select and compute a small subset of the columns of $\mathbf{A}$, to produce an

estimate of $\mathbf{W}$ [40, 19] [1] , which is stated below.

Denoting by $\mathbf{A}(:, j)$ the j-th column of $\mathbf{A}$, we compute $\mathbf{W}$ in the following way:

$$\mathbf{W} = \sum_{j=1}^{N^{d+1}} \mathbf{A}(:, j) \cdot \mathbf{A}(:, j)'. \tag{5.1}$$

Consequently, $\mathbf{W}$ is a sum of N^{d+1} rank-1 matrices, i.e., the products of the columns of $\mathbf{A}$ and their transposes. Let $j_1, \ldots, j_c$ be c integers that are randomly selected between 1 and N^{d+1}. Then $\mathbf{W}$ can be approximated as follows [40] [2] :

$$\mathbf{W} \approx \sum_{t=1}^{c} \mathbf{A}(:, j_t) \cdot \mathbf{A}(:, j_t)'. \tag{5.2}$$

In practice, in order to have at most quadratic complexity, we expect the maximum possible c to be an absolute constant or a small number times N, resulting in $c/N^{d+1} \leq O(N^{-d})$. We thus conclude that uniform sampling (maintaining quadratic complexity) is almost surely not able to capture the column space of $\mathbf{A}$ when N is large and d is moderate. Indeed, this is demonstrated in Figure 5.1(a): In the two cases where $d > 2$, the error e_{OLS} does not get close to the model error even with $c = 100 \cdot N$. This illustrates a fundamental limitation of uniform sampling. In the following we explain our strategy to resolve this issue.

We note that each column j of $\mathbf{A}$ uniquely corresponds to an ordered list of $d + 1$ points $(\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \ldots, \mathbf{x}_{j_{d+1}})$, and moreover, repeated points lead to a zero column (see equation (1.11)). Thus, we will select only tuples of $d + 1$ distinct points in X when sampling columns of $\mathbf{A}$.

We say that an n-tuple of points is *pure* if these n points are in the same underlying cluster, and that it is *mixed* otherwise. Similarly, a column of the matrix $\mathbf{A}$ is said to

[1] In [40] a more accurate sampling scheme according to the magnitudes of the columns is also suggested. Nevertheless, since we do not have the full affinity matrix $\mathbf{A}$, this technique can not be applied in our setting.

[2] More precisely, a scaling constant needs to be used in front of the sum in order to have the right magnitude (see [40, Section 4]). However, since we are only interested in the eigen-structure of $\mathbf{W}$, this constant is omitted.

be pure if it corresponds to a pure $(d+1)$-tuple, and mixed otherwise. We use these two categories of columns of $\mathbf{A}$ to explain our sampling strategy.

In the ideal case (see Section 3.1), any mixed column of $\mathbf{A}$ is identically zero and thus makes no contribution to computing the matrix $\mathbf{W}$. On the other hand, the pure columns lead to a block diagonal structure of $\mathbf{W}$, which guarantees a perfect segmentation (see Proposition 3.1.1). In practice the mixed columns are typically not all zero. Since the percentage of the mixed columns in $\mathbf{A}$ is high, the matrix $\mathbf{W}$ loses the desired block diagonal structure. If we only use the pure columns of $\mathbf{A}$, then we can expect $\mathbf{W}$ to be nearly block diagonal.

The iterative sampling scheme is motivated by the above observations and works as follows. We fix c to be some constant, e.g., $c = 100 \cdot K$. Initially, c columns of $\mathbf{A}$ are randomly selected and computed so as to produce $\mathbf{W}$, and then an initial segmentation of X into K clusters is obtained with this $\mathbf{W}$ (we call this initial step *the zeroth iteration*). We then re-sample c columns of $\mathbf{A}$ by selecting c/K columns from within each of the K initially found clusters, or from the points within a small strip around the OLS d-flat of each such cluster, and obtain K newer clusters. In order to achieve the best segmentation, one can iterate this process a few times, as the newer clusters are expected to be closer to the underlying clusters.

We demonstrate the strength of this sampling strategy by repeating the experiments in Figure 5.1(a), but with iterative sampling replacing uniform sampling. Due to the randomness of sampling, we compute both the mean and the standard deviation of the errors e_{OLS} in the 500 experiments in each of the intermediate steps of iterative sampling (see Figure 5.1(b)). In all cases, the mean drops rapidly below the model error when iterating, and the standard deviation also decays quickly.

We remark that as d increases, we should also use larger c in the zeroth iteration in order to capture "enough" pure columns. Indeed, in order to have (on average) c_0

pure columns sampled from each underlying cluster in the zeroth iteration, we need to have $c \approx c_0 \cdot K^{d+2}$. Afterwards, we may still reduce c to a constant multiple of K in the subsequent iterations. We plan to study more carefully the required magnitudes of c (for the zeroth iteration and the subsequent iterations respectively) to ensure convergence. When the theoretical value of c is unrealistically large, we can sample columns in other ways, e.g., from the output of other d-flats clustering algorithms (such as K-Subspaces) to initialize SCC.

5.1.2 Estimation of the Tuning Parameter σ

The choice of the tuning parameter σ is crucial to the performance of any algorithm involving Gaussian-kernel affinities. However, selecting its optimal value is not an easy task, and also is insufficiently investigated in the literature. Common practice is to manually select a small set of values and choose the one that works the best (e.g., [29]). Since the optimal value of σ should depend on the scale of data, subjective choices may work poorly (see Figure 5.2). We develop an automatic scheme to infer the optimal value of σ (or an interval containing it) from the data itself.

We start by assuming that all curvatures are computed (which is unrealistic when d is large). In this case, we estimate the correct choice of σ, starting with the clean case and then corrupting it by noise. We follow by examining the practical setting of c sampled columns, i.e., when only a fraction of the curvatures are computed.

In the clean case, the polar curvatures of all pure $(d+2)$-tuples are zero. In contrast, (almost) all mixed $(d+2)$-tuples have positive curvatures [3] . By taking a sufficiently small $\sigma > 0$ the resulting affinity tensor can closely approximate the perfect tensor (see Definition 3.1.1), thus an accurate segmentation is guaranteed. When the data is corrupted with moderate noise, we still expect the curvatures of most pure $(d+2)$-tuples

[3] When a mixed $(d+2)$-tuple happen to be lying on a d-flat, the polar curvature will be correspondingly zero. However, such mixed tuples should be rare in most cases.

to be small, and those of most mixed $(d+2)$-tuples to be large. The optimal value of σ, σ_{opt}, is the maximum of the small curvatures corresponding to pure tuples (up to a scaling constant). Indeed, transforming the curvatures by $\exp(-\cdot/(2\sigma_{\text{opt}}^2))$ will produce affinities that are close to zero (for mixed tuples) and one (for pure tuples). In other words, this transformation serves like a "low-pass filter": It "passes" smaller curvatures by producing large affinities toward 1, and "blocks" bigger curvatures toward zero.

Therefore, in the case of small within-cluster curvatures and large between-cluster curvatures, one can compute all the curvatures, have them sorted in an increasing order, estimate the number of small curvatures corresponding to pure tuples, and take as σ the curvature value at that particular index in the sorted vector. The key step is determining the index of that curvature value. For this reason we refer to our approach as *index estimation*.

We next obtain this index in two cases. First, we suppose that all N_j are known. Then the proportion of pure $(d+2)$-tuples to all $(d+2)$-tuples equals:

$$\gamma = \frac{\sum_{1\le j\le K} \mathrm{P}(N_j, d+2)}{\mathrm{P}(N, d+2)} \approx \sum_{j=1}^{K} \left(\frac{N_j}{N}\right)^{d+2}. \tag{5.3}$$

That is, the curvature value at the index of $\gamma \cdot \mathrm{P}(N, d+2)$ can be used as the best estimate for the optimal σ. Second, when N_j are unknown, we work out the absolute minimum[4] of the last quantity in equation (5.3) and use it as a lower bound for the fraction γ:

$$\gamma \gtrapprox K \cdot \left(\frac{1}{K}\right)^{d+2} = \frac{1}{K^{d+1}}. \tag{5.4}$$

We note that if all N_j are equal to N/K, then this lower bound coincides with its tighter estimate provided in equation (5.3). The following example demonstrates this strategy.

[4] The absolute minimum can be obtained by solving a constrained optimization problem:

$$\min_{\gamma_1,\ldots,\gamma_K>0} \sum_{j=1}^{K} \gamma_j^{d+2} \qquad \text{subject to} \quad \sum_{j=1}^{K} \gamma_j = 1.$$

The minimum is attained when $\gamma_j = 1/K,\ j = 1, \ldots, K$.

Example 5.1.1. We take the data in Figure 5.2 which consists of three lines in $\mathbb{R}^2$, each having 25 points. This data set has a relatively small size, so we are able to compute all the polar curvatures. We apply equation (5.3) (or (5.4)) and obtain that $\gamma \approx 1/9$. Thus, we use the $1/9 \cdot \mathrm{P}(75, 2) = 617^{\text{th}}$ smallest curvature as the optimal value of the tuning parameter: $\sigma = 1.5111$. We also remark that the optimal value $\sigma = 0.1840$ in Example 3.2.5 was obtained similarly.

We now go to our practical setting (Section 5.1.1) where we iteratively sample only c columns of $\mathbf{A}$ and thus do not have all the curvatures. We assume convergence of the iterative sampling so that the proportion of pure columns (in the c sampled columns) increases with the iterations. Consequently, we obtain a lower bound for σ from the zeroth iteration, and an upper bound from the last iteration.

In the zeroth iteration (uniform sampling), c columns of $\mathbf{A}$ are randomly selected. We expect to have the same lower bound as in equation (5.4) for the proportion of pure $(d+2)$-tuples in these c columns. We note that there are exactly $N - d - 1$ elements corresponding to tuples of $d + 2$ distinct points in each of these c columns. Denoting by $\mathbf{c}$ the vector of the $(N - d - 1) \cdot c$ corresponding curvatures sorted in an increasing order, we write a lower bound for σ as follows:

$$\sigma_{\min} = \mathbf{c}\left((N - d - 1) \cdot c/K^{d+1}\right). \tag{5.5}$$

In the last iteration (when the scheme converges to finding the true clusters), c/K columns are sampled from each of the K underlying clusters, thus all the c columns are pure. In this case, the number of pure $(d+2)$-tuples in the c columns attains the following maximum possible value:

$$\sum_{j=1}^{K}(N_j - d - 1) \cdot \frac{c}{K} = N \cdot c/K - (d+1) \cdot c. \tag{5.6}$$

Therefore, we have the following upper bound for σ:

$$\sigma_{\max} = \mathbf{c}\left((N/K - d - 1) \cdot c\right). \tag{5.7}$$

We present two practical ways of searching the interval $[\sigma_{\min}, \sigma_{\max}]$ for the optimal value of σ. First, one can start with the upper bound $\sigma_{\max}$ and divide it by a constant (e.g., 2) each time until it falls below the lower bound $\sigma_{\min}$. Second, one can search by the index of the vector $\mathbf{c}$, i.e., choose the optimal value from a subset of $\mathbf{c}$:

$$\{\mathbf{c}\,(N \cdot c/K^q) \mid q = 1, \ldots, d+1\}. \tag{5.8}$$

We remark that the second strategy always requires $d+1$ searches for σ, thus one can have control over the total number of iterations. We have found in experiments that this search strategy works sufficiently well. To further improve efficiency, we can gradually raise the lower bound (i.e., $\sigma_{\min}$) in the subsequent iterations.

5.1.3 Initialization of K-means

The clustering step in the TSCC algorithm applies K-means to the rows of $\mathbf{U}$. In the ideal case, these rows coincide with K mutually orthogonal vectors (the "seeds") in $\mathbb{R}^K$ (see Proposition 3.1.1); in the case of noise, the rows of $\mathbf{U}$ correspond to more than K points that originate from those seeds and possibly overlap in between. See Figure 5.3 for an illustration. We locate these seeds by maximizing the variance among all possible combinations of K rows of $\mathbf{U}$, and then use them to initialize K-means.

Formally, the indices of these seeds can be found by solving the following optimization problem:

$$\{s_1, \ldots, s_K\} = \arg \max_{1 \le n_1 < \cdots < n_K \le N} \sum_{i=1}^{K} \|\mathbf{U}(n_i, :) - \frac{1}{K} \cdot \sum_{j=1}^{K} \mathbf{U}(n_j, :)\|^2. \tag{5.9}$$

With a little algebra we obtain an equivalent representation [5] :

$$\{s_1, \ldots, s_K\} = \arg \max_{1 \le n_1 < \cdots < n_K \le N} \sum_{1 \le i < j \le K} \|\mathbf{U}(n_i, :) - \mathbf{U}(n_j, :)\|^2. \tag{5.10}$$

[5] When all the rows of $\mathbf{U}$ have unit length, this criterion reduces to minimizing the total sum of inner products among all possible combinations of K rows of $\mathbf{U}$. With this normalization, our strategy (equations (5.11) and (5.12)) still differentiates from that of Ng et al. [29].

We thus apply an inductive scheme (via equation (5.10)) to solve the above maximization problem. The first index s_1 is chosen to be that of the row farthest from the center of all N rows. That is,

$$s_1 = \arg \max_{1 \leq n \leq N} \|\mathbf{U}(n,:) - \frac{1}{N} \sum_{i=1}^{N} \mathbf{U}(i,:)\|. \tag{5.11}$$

Suppose now that $1 \leq k < K$ seeds have been chosen, then the index of the $(k+1)$-st seed is determined by

$$s_{k+1} = \arg \max_{\substack{1 \leq n \leq N \\ n \neq s_1, \ldots, s_k}} \sum_{i=1}^{k} \|\mathbf{U}(s_i,:) - \mathbf{U}(n,:)\|^2. \tag{5.12}$$

5.2 The SCC Algorithm

We combine together the theoretical algorithm and all the techniques introduced in the previous section to form a comprehensive Spectral Curvature Clustering (SCC) algorithm for practical use (Algorithm 2).

Algorithm 2: Spectral Curvature Clustering (SCC)

input : Data set X, intrinsic dimension d, number of d-flats K (*required*); number of sampled columns c (*default* $= 100 \cdot K$)

output: K disjoint clusters $\mathrm{C}_1, \ldots, \mathrm{C}_K$ and error e_{OLS}

begin

Sample randomly c subsets of X, each containing exactly $d+1$ distinct points.

repeat

Compute the polar curvature of any subset and each of the rest of points in X by equation (1.2), and sort increasingly into a vector $\mathbf{c}$ those $(N-d-1) \cdot c$ curvatures.

for $q = 1$ **to** $d+1$ **do**

Use equation (2.3) with $q = 2$ and $\sigma = \mathbf{c}\,(N \cdot c/K^q)$ to compute the c selected columns of $\mathbf{A}$. Form a matrix $\mathbf{A}_c \in \mathbb{R}^{N \times c}$ using these c columns.

Compute $\mathbf{D} = \mathrm{diag}\{\mathbf{A}_c \cdot (\mathbf{A}_c' \cdot \mathbf{1})\}$ and use it to normalize $\mathbf{A}_c$: $\mathbf{A}_c^* = \mathbf{D}^{-1/2} \cdot \mathbf{A}_c$.

Stack in columns the top K left singular vectors of $\mathbf{A}_c^*$ to form $\mathbf{U}$.

Apply K-means, initialized according to equations (5.11) and (5.12), to the rows of $\mathbf{U}$ [a] and separate them into K clusters.

Use these detected clusters to group the points of X into K subsets, and compute the corresponding error e_{OLS} using equation (2.1).

end

Record the K subsets $\mathrm{C}_1, \ldots, \mathrm{C}_K$ of X that correspond to the smallest error e_{OLS} in the above loop. Sample c/K $(d+1)$-tuples from each C_j (or the points within a small strip around each of their OLS d-flats).

until e_{OLS} *converges*

end

[a] The reader might want to apply the $\mathbf{V}$ normalization (equation (3.20)) to the rows of $\mathbf{U}$ before K-means in order to obtain better results. See Section 3.3.1 for relevant discussions.

5.3 Complexity of the SCC Algorithm

The implementation of the SCC algorithm is mainly through standard matrix operations, such as element-wise manipulation, matrix-vector multiplication, Singular Value Decomposition (SVD), etc.. Consequently, the complexity of SCC is completely determined by the sizes of the matrices used in the algorithm and the types of operations between them.

The storage requirement of the algorithm is $O(N \cdot (D + c))$. Indeed, the biggest matrices are X (when considered as a matrix), whose size is $N \times D$, and $\mathbf{A}_c, \mathbf{A}_c^*$ (defined in Algorithm 2), which have size $N \times c$. In order to estimate the running time, we first note that it takes $O((d + 1) \cdot D \cdot N \cdot c))$ time to compute $\mathbf{A}_c$ by using matrix manipulations (see code at *http://www.math.umn.edu/∼lerman/scc/*). Also, it takes $O(N \cdot c)$ time to compute $\mathbf{D}, \widetilde{\mathbf{A}}_c$, and $O((N + c) \cdot K^2)$ time to calculate $\mathbf{U}$ by fast SVD algorithms (e.g. [41]). Thus, each iteration takes $O((d + 1)^2 \cdot D \cdot N \cdot c)$ time (the computation is repeated $d+1$ times in Step 2 of Algorithm 2). Let n_{s} denote the number of sampling iterations performed. We then obtain that the total running time of the SCC algorithm is $O(n_{\mathrm{s}} \cdot (d + 1)^2 \cdot D \cdot N \cdot c)$.

5.4 Outliers Detection

We detect outliers according to the degrees of the data points, i.e., the diagonal elements of the matrix $\mathbf{D}$ in Algorithm 2. We assume that the percentage of outliers is known. In each sampling iteration, after the degrees $\mathbf{D}$ have been computed, we isolate the percentage of points with the smallest degrees as intermediate outliers, and remove the corresponding rows from the matrix $\mathbf{A}_c$. We then re-compute $\mathbf{D}$ from the reduced matrix $\mathbf{A}_c$ and follow the subsequent steps of SCC to obtain K clusters. In the next iteration, we will sample c/K columns only from each of the previously detected clusters

to form $\mathbf{A}_c$ (thus excluding the previous outliers). Those outliers isolated in the final sampling iteration will be the ultimate outliers.

To evaluate the performance of this outliers detection strategy associated with SCC, we plot in Figure 5.4 a *Receiver Operating Characteristic (ROC) curve* in the case of lines contaminated with outliers. An ROC curve is the plot of the *true positive rates* (TPR) against the *false positive rates* (FPR). The TPR is the percentage of correctly detected outliers; while the FPR is the percentage of data points in the stable distribution which are falsely detected as outliers. A large area under the ROC curve is indication of good performance in outliers detection for a wide rage of FPRs. The area of the region under the ROC curve corresponding to SCC is 0.8105. In comparison, the Robust GPCA algorithm (RGPCA) [2] has an area of 0.7613 under its ROC curve. The figure also emphasizes the fact that SCC has a better performance than RGPCA at low FPRs which are practically more important.

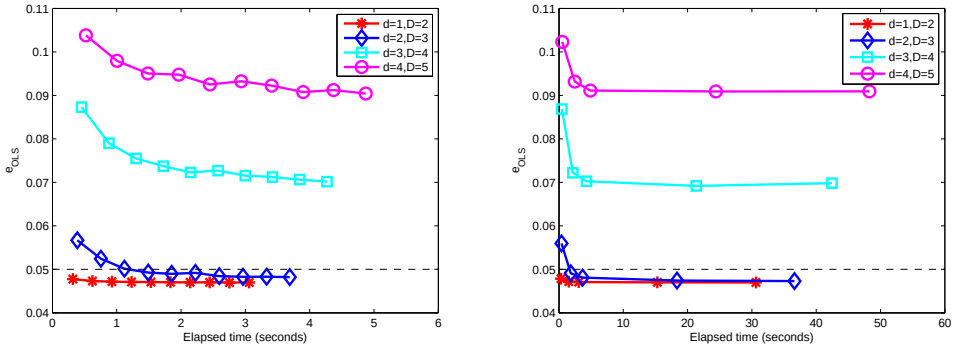
5.5 Mixed Dimensions

The SCC algorithm is formulated in the setting of data sampled from flats of the same dimension d. In fact, it can be easily adapted to cluster flats of mixed dimensions, i.e., when the dimensions $d_1, d_2, \ldots, d_K$ are not necessarily the same.

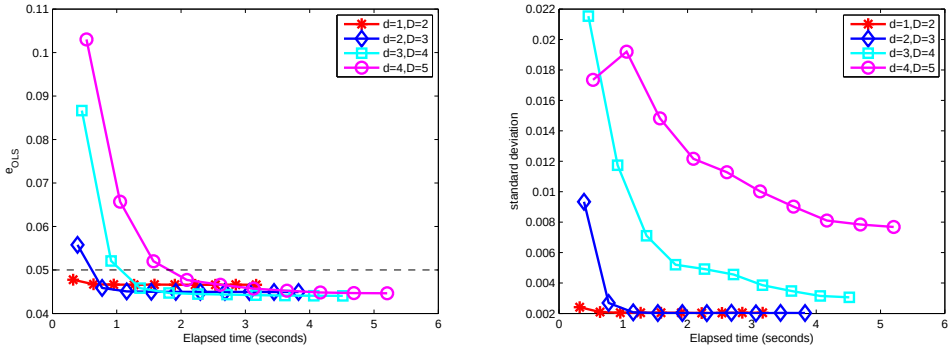
Our strategy is to use the maximum of the dimensions

$$d_{\max} = \max_{1 \leq j \leq K} d_j, \tag{5.13}$$

and apply SCC to segment K $d_{\max}$-flats. We find in experiments that this technique often results in small segmentation errors $e_{\%}$. At this stage we cannot compute e_{OLS} due to not knowing the intrinsic dimensions of the detected clusters. We will try to resolve this issue in later research.

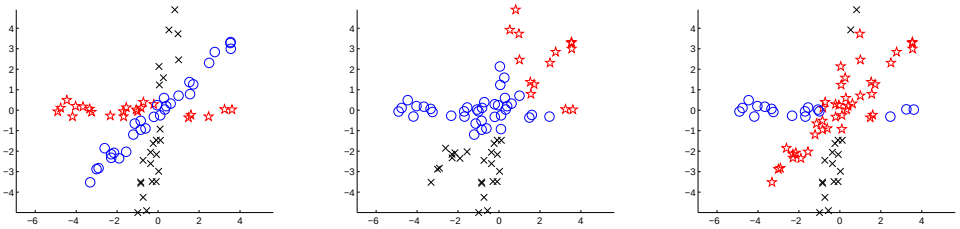


(a) Uniform Sampling: The errors obtained using different choices of c. On each curve a symbol represents a distinct value of c. Left: c is taken to be $N, 2N, \ldots, 10N$ respectively; Right: $c = N, 5N, 10N, 50N, 100N$.



(b) Iterative Sampling: The mean (left) and standard deviation (right) of the errors obtained in the initial step (uniform sampling) and the first 9 updates using iterative sampling with $c = N = 100 \cdot K$ always fixed.

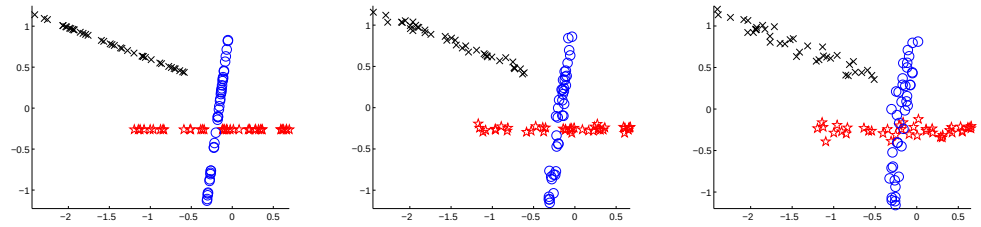
Figure 5.1: Plots of the errors (e_{OLS}) using different sampling strategies against time. In each experiment we randomly generate $K = 3$ d-dimensional linear subspaces in $\mathbb{R}^D$. Each subspace contains 100 points, so $N = 100 \cdot K$. The model error is 0.05 in all situations (indicated by the dash lines). We repeat this experiment 500 times (for each fixed pair (d, D)) in order to compute an average of e_{OLS} for each iteration.



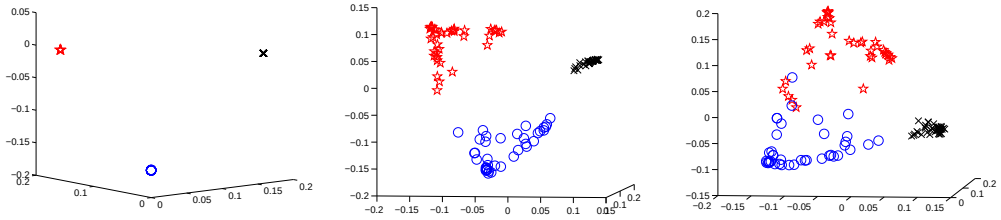
(a) $\sigma = 1.5111$ (b) $\sigma = 100$ (c) $\sigma = 10$

(d) $\sigma = 0.1$ (e) $\sigma = 0.01$ (f) $\sigma = 0.001$

Figure 5.2: Segmentation results with different choices of σ. The value 1.5111 is inferred from data using our strategy (explained in Example 5.1.1); the other values are manually selected.



(a) Data and clusters



(b) New coordinates in $\mathbb{R}^3$ (i.e., the rows of $\mathbf{U}$)

Figure 5.3: Three data sets of the same model but with increasing levels of noise, and their images in the embedded space.

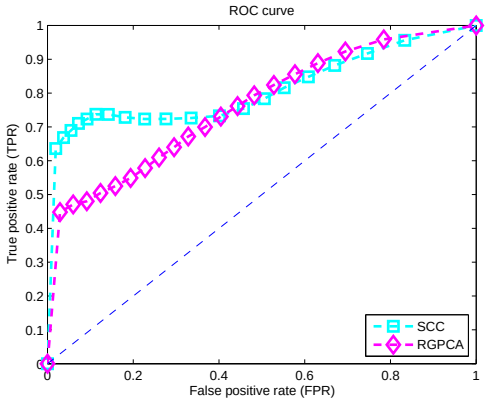


Figure 5.4: ROC curves corresponding to SCC and RGPCA. We randomly generate $K = 3$ linear lines in $\mathbb{R}^2$, and sample 100 points from each line. The samples are then corrupted with 5% Gaussian noise and further contaminated with some percentage of outliers. The percentages used are $5\%, 10\%, 15\%, \ldots, 95\%$ respectively, as indicated by the symbols on each curve. For each fixed percentage, 500 experiments are repeated in order to compute an average for each of the two rates.

Chapter 6

Experiments

In this chapter we test the performance of the SCC algorithm on both synthetic data and real-world applications.

6.1 Simulations

We compare the SCC algorithm (and also LSCC when applicable) with other competing methods on a number of artificial data sets in the setting of hybrid linear modeling.

The three methods that we compare with are the Mixtures of Probabilistic PCA algorithm (MoPPCA) [15], the K-Subspaces algorithm (KS) [9], and the GPCA algorithm with voting (GPCA) [2]. We use the Matlab codes of the GPCA algorithm that are readily available at *http://perception.csl.uiuc.edu/gpca/*. We also borrow from that web site the Matlab code that generates various data sets. MoPPCA and KS are implemented by Stefan Atev and ourselves (see codes at *http://www.math.umn.edu/∼lerman/scc/*). These two methods are always initialized with a random guess of the membership of the data points. Due to the randomness in the initialization, multiple restarts are used and the best segmentation result is recorded.

The three multi-way clustering algorithms [18, 19, 20] seem highly related and should have been included for comparison. However, they mainly focus on how to process a given affinity tensor; many practical and sensitive issues are not fully discussed in the context of hybrid linear modeling, and are also missing from their implementation. In fact, we have compared with [19] (in Figures 5.1 and 5.2) regarding random sampling and choices of the tuning parameter σ. We also tried to compare with k-Manifolds [22]. However, this method tends to find curves/surfaces instead of straight lines/flat planes, so it performs poorly in this context and is also not included.

In the following we conduct experiments in the cases of linear/affine subspaces of the same dimension/mixed dimensions to compare the performance of the four algorithms, namely MoPPCA, KS, GPCA, and SCC. The simulations were performed on a compute server with two dual-core AMD Opteron 64-bit 280 processors (2.4 GHz) with 8 GB of RAM. We remark that when applying SCC (Algorithm 2) we fix $c = 100 \cdot K$.

We first randomly generate K *linear* subspaces of a fixed dimension d in some Euclidean space $\mathbb{R}^D$, which we write $d^K \in \mathbb{R}^D$ for short. We follow [2] to mandate the angles between these subspaces to be at least 30 degrees in order to ensure enough separation. Also, the diameter of each subspace is fixed to be 1. We then randomly draw 100 samples from each of the subspaces, and corrupt them with 5% Gaussian noise. We apply the four algorithms to the data and record both types of errors e_{OLS} and $e_{\%}$ as well as the computation time t. This experiment is repeated 500 times and the averaged errors and time are shown in Table 6.1. In all the three scenarios, MoPPCA, KS and SCC have comparable performance, but they all outperform GPCA at $1 - 10^{-7}$ confidence level using paired t-tests.

We also note that LSCC has a slightly better segmentation result than SCC. The reasons are explained as follows: (1) The new matrix $\mathbf{A}$ (in full form) has less columns than before by one order of N, so the same number of sampled columns can be a better

Table 6.1: The two types of errors $e_{\text{OLS}}, e_{\%}$ and computation time t (in seconds) of the four methods when clustering *linear* subspaces. The number of subspaces K and the intrinsic dimension d are given to all algorithms. The MoPPCA and KS algorithms are randomly initialized. Ten restarts are used for each of them, and the smallest error is used.

	$2^4 \in \mathbb{R}^3$			$3^3 \in \mathbb{R}^4$			$4^3 \in \mathbb{R}^6$		
	e_{OLS}	$e_{\%}$	t	e_{OLS}	$e_{\%}$	t	e_{OLS}	$e_{\%}$	t
MoPPCA	.042	19.2%	0.7	.043	16.8%	0.4	.048	3.2%	0.4
KS	.043	19.5%	0.2	.043	16.3%	0.2	.048	3.1%	0.2
LSCC	.043	19.8%	1.8	.044	17.3%	1.5	.048	3.4%	1.8
SCC	.048	23.1%	3.0	.044	18.2%	2.0	.048	3.6%	2.1
GPCA	.088	39.5%	1.5	.077	32.5%	1.3	.126	31.7%	3.1

representative of the column space of $\mathbf{A}$; (2) With $d+1$ points and the origin, a small curvature always implies that the $d+1$ points are close to being on some underlying linear subspace. This excludes the unfavorable small curvatures for $d+2$ points lying around an affine subspace (which is the case for SCC).

We next compare the SCC algorithm with the other methods on clustering *affine* subspaces. We generate affine subspaces with the same controlling parameters as in the linear case. We remark that the software borrowed from *http://perception.csl.uiuc.edu/gpca/* tries to avoid intersection of these affine subspaces, or more precisely, of the sampled clusters. We note that, since SCC does not distinguish between linear and affine subspaces, its performance in the case of intersecting affine subspaces can be reflected in Table 6.1 (where we have intersecting linear subspaces). The two types of errors due to all four methods and their computation time are recorded in Table 6.2. The results of paired t-tests between SCC and the other three methods show that SCC performs better at $1 - 10^{-7}$ confidence level in terms of both errors.

We finally compare all the algorithms on clustering linear/affine subspaces of *mixed* dimensions in order to further evaluate their performance. We follow the notation in [2] to denote data sampled from subspaces of mixed dimensions by $(d_1, \ldots, d_K) \in \mathbb{R}^D$. All

Table 6.2: The two types of errors $e_{\text{OLS}}, e_{\%}$ and computation time t (in seconds) of the four algorithms when clustering *affine* subspaces. The number of subspaces K and the intrinsic dimension d are given to all algorithms. The MoPPCA and KS algorithms are randomly initialized. Ten restarts are used for each of them, and the smallest error is used.

	$1^4 \in \mathbb{R}^2$			$2^3 \in \mathbb{R}^3$			$4^3 \in \mathbb{R}^5$		
	e_{OLS}	$e_{\%}$	t	e_{OLS}	$e_{\%}$	t	e_{OLS}	$e_{\%}$	t
GPCA	.174	29.1%	1.3	.116	20.1%	1.0	.138	30.2%	1.5
MoPPCA	.110	35.4%	0.7	.115	47.6%	0.6	.089	49.0%	0.9
KS	.089	25.5%	0.2	.113	45.4%	0.1	.090	49.3%	0.2
SCC	.049	4.2%	1.3	.049	2.8%	1.0	.048	1.4%	2.1

the parameters used in generating data are the same as above, except that the noise level becomes 3%. Table 6.3 shows the percentage of misclassified points (i.e., $e_{\%}$) and elapsed time by each of the four algorithms in eight scenarios. Without further processing, the LSCC (resp. SCC) algorithm in the case of linear (resp. affine) subspaces still exhibits better performance in terms of $e_{\%}$ than its competitors at $1 - 10^{-7}$ confidence level (using paired t-tests).

6.2 Applications

Hybrid linear modeling has broad applications in many areas, such as computer vision, image processing, pattern recognition, and system identification. We exemplify below the application of the SCC algorithm to a few real-world problems that are studied in [1, 2].

6.2.1 Motion Segmentation under Affine Camera Models

Suppose that a video sequence consists of F frames of images of several objects that are moving independently against the background, and that N feature points $\mathbf{y}_1, \ldots, \mathbf{y}_N \in \mathbb{R}^3$ are detected on the objects and the background. Let $\mathbf{z}_{ij} \in \mathbb{R}^2$ be the coordinates of

Table 6.3: The percentage of misclassified points $e_{\%}$ and elapsed time t (in seconds) by all algorithms when clustering hybrid data sampled from linear(L)/affine(A) subspaces of *mixed* dimensions. The dimensions of the subspaces are given to all the algorithms. The MoPPCA and K-Subspaces algorithms are randomly initialized. Ten restarts are used for each of them, and the smallest error is used.

	$(1,2,2) \in \mathbb{R}^3$		$(1,1,2) \in \mathbb{R}^3$		$(1,1,2,2) \in \mathbb{R}^3$		$(1,2,3) \in \mathbb{R}^4$	
$e_{\%} \cdot 100$	L	A	L	A	L	A	L	A
KS	10.6	34.1	11.2	26.9	21.8	36.6	19.5	38.6
MoPPCA	8.0	41.4	24.0	37.6	20.4	44.0	24.0	31.8
GPCA	7.3	11.7	17.8	18.1	25.2	24.7	13.2	17.4
SCC	7.2	1.0	9.2	0.5	18.6	1.4	8.4	0.3
LSCC	6.1		7.1		10.8		6.6	
t								
KS	0.1	0.1	0.1	0.1	0.2	0.2	0.1	0.1
MoPPCA	0.4	0.6	0.4	0.6	0.8	1.0	0.6	0.7
GPCA	2.2	2.5	2.1	1.9	3.5	3.6	6.4	4.5
SCC	1.2	0.9	1.1	0.9	2.2	1.6	1.6	1.4
LSCC	0.7		0.8		1.3		1.5	

the feature point $\mathbf{y}_j$ in the i-th image frame for every $1 \leq i \leq F$ and $1 \leq j \leq N$. Then $\mathbf{z}_j = [\mathbf{z}'_{1j}\ \mathbf{z}'_{2j}\ \ldots\ \mathbf{z}'_{Fj}]' \in \mathbb{R}^{2F}$ represents the trajectory of the j-th feature point across the F frames. The problem is how to separate these trajectory vectors $\mathbf{z}_1, \ldots, \mathbf{z}_N$ into independent motions undertaken by those objects and the background.

It has been shown (e.g., in [2]) that, under affine camera models and with some mild conditions, the trajectory vectors corresponding to different moving objects and the background across the F image frames live in distinct linear subspaces of dimension at most four in $\mathbb{R}^{2F}$, or affine subspaces of dimension at most three within those linear subspaces.

We borrow the data from [42], which are also used in [2]. This data consist of two outdoor sequences taken by a moving camera tracking a car moving in front of a parking lot and a building (Sequences A and B), and one indoor sequence taken by a moving camera tracking a person moving his head (Sequence C), as shown in [42, Figure 7].

Following the above theory, we first apply SCC (Algorithm 2) as well as LSCC to segment two 4-dimensional linear subspaces in $\mathbb{R}^{2F}$ for each of the three sequences. We also apply SCC to each sequence and segment 3-dimensional affine subspaces in $\mathbb{R}^{2F}$. In all these cases, SCC obtains 100% accuracy. In contrast, GPCA cannot be applied directly to the original trajectories in Sequences A and C, as it is computationally too expensive to find all the normal vectors of these low dimensional linear subspaces within a high dimensional ambient space. Even in Sequence B where we could apply GPCA (where $2F = 34$ is small), it produces varying errors, which are sometimes nearly 40% (see Table 6.4).

Table 6.4: Percentage of misclassified points $e_{\%}$ by SCC and GPCA respectively using different combinations (d, D). Here d is the dimension of the subspaces, and D is the ambient dimension. Both algorithms are without post-optimization. In the table below N/A represents *Not Applicable*, while VE is short for *Varying Errors*.

Sequence		A	B	C
Number of points N		136	63	73
Number of frames F		30	17	100
SCC/LSCC	$d = 4, D = 2F$	0%	0%	0%
SCC	$d = 3, D = 2F$	0%	0%	0%
PCA+SCC	$d = 3, D = 4$	0%	0%	0%
GPCA	$d = 3/4, D = 2F$	N/A	VE	N/A
SVD+GPCA	$d = 4, D = 5$	0%	0%	40%

To further evaluate the performance of the two algorithms, we have also applied GPCA and SCC to the three sequences after reducing the ambient dimensions. We first project the trajectories onto a 5-dimensional space by direct SVD (to maintain the linear structure), and apply GPCA to segment 4-dimensional linear subspaces in $\mathbb{R}^5$ as suggested in [2], but without post-optimization by KS. A segmentation error as large as 40% is obtained for Sequence C (see Table 6.4). The equivalent way of applying SCC is to first project the data onto the first four principal components by PCA, and then segment 3-flats in $\mathbb{R}^4$. Again, SCC achieves zero error (see Table 6.4).

6.2.2 Face Clustering under Varying Lighting Conditions

We study the problem of clustering a given collection of images of human faces in fixed pose under varying illumination conditions. It has been proved that the set of all images of a Lambertian object under a variety of lighting conditions form a convex polyhedral cone in the image space, and this cone can be accurately approximated by a low-dimensional linear subspace (of dimension at most 9) [9, 43, 44]. If we assume that images of different faces lie in different subspaces, then we can cluster these images by segmenting an arrangement of linear subspaces using SCC (and also LSCC).

Figure 6.1: The ten subjects in the Yale Face Database B. First row: subjects 1 through 5; second row: subjects 6 to 10.

Following Vidal et al. [1] we use a subset of the Yale Face Database B [45] consisting of the frontal face images of three subjects (numbered by 5, 8, and 10) of the ten (see Figure 6.1) under 64 varying lighting conditions. There are $N = 64 \times 3$ images in total. For computational efficiency, we have downsampled each image to 120×160 pixels, so the dimension of the image space is $D' = 120 \times 160$. We then stack these images (after vectorized) into a $D' \times N$ matrix $\mathbf{X}$ and apply SVD to reduce the ambient dimension to $D \ll D'$, forming a new matrix $\mathbf{Y} \in \mathbb{R}^{D \times N}$.

We apply SCC to the columns of $\mathbf{Y}$ and cluster three d-dimensional linear subspaces in $\mathbb{R}^D$. The above theory indicates that d should be at most 9. We have tried all the

possible combinations $0 \leq d < D \leq 10$. The pairs (d, D) with which SCC and LSCC give a perfect segmentation are listed in Table 6.5. In comparison, we have also applied the GPCA-voting algorithm to the columns of $\mathbf{Y}$ with $0 \leq d < D \leq 10$. There are many situations where GPCA does not give 100% accuracy but SCC does (see Table 6.5).

Table 6.5: Combinations (d, D) with which SCC and GPCA achieves a perfect segmentation respectively. Here d is the dimension of the subspaces while D is the ambient dimension.

Methods	(d, D)
SVD+SCC	$(0, 2 \leq D \leq 4), (1, 3/4), 2 \leq d < D \leq 10$
SVD+LSCC	$(1, 3/4/5/7/8), 2 \leq d < D \leq 10$
SVD+GPCA	$(3, 5), (4, 6), (4, 7), (5, 7), (4, 8), (6, 8)$

Vidal et al. [1] suggest to first project the data onto the top three principal components and then apply GPCA to the data in homogeneous coordinates by fitting three linear subspaces of dimensions 3, 2, and 2 in $\mathbb{R}^4$. They obtain zero error in this case. However, we are not aware of the reason of using mixed dimensions. We follow their strategy but instead we apply GPCA using the same dimension 3 for each linear subspace. Then a segmentation error of about 4% is obtained. We note that applying GPCA with $d = 3$ for each linear subspace (in homogeneous coordinates) in $\mathbb{R}^4$ is equivalent to applying SCC with $D = 3$ and $d = 2$. In this case, SCC achieves a perfect segmentation.

6.2.3 Temporal Segmentation of Video Sequences

We consider the problem of partitioning a long video sequence into multiple short segments corresponding to different scenes. We assume that all the image frames having the same scene live in a low dimensional subspace of the image space and that different scenes correspond to different subspaces. We show that the SCC and LSCC algorithms

can be applied to solve this problem.

Figure 6.2: The first, 56th and last (135th) frames of the Fox video sequence.

The video sequence that we received from Rene Vidal is about an interview at Fox TV (Figure 6.2). It is also used in [1]. It consists of 135 images of size 294×413, each containing either the interviewer alone, or the interviewee alone, or both. We would like to segment these images into the three scenes. We view each image frame as a sample point in $\mathbb{R}^{D'}$, where $D' = 294 \times 413$. We first apply SVD to reduce the ambient dimension from D' to $D \leq 10$, and then apply SCC to segment three d-dimensional linear subspaces within $\mathbb{R}^D$. The combinations (d, D) with which SCC/LSCC obtains 100% accuracy are reported in Table 6.6.

Table 6.6: The pairs (d, D) with which each algorithm obtains 100% accuracy. Here D is the ambient dimension while d is the dimension of the subspaces.

Method	(d, D)
SVD+SCC	$(0, 1/2/3/4), (1, 3/4), (2, 3/4/5)$
SVD+LSCC	$(1, 3), (2, 3/4), (3, 4)$
SVD+GPCA	NONE

Vidal et al. [1] applied GPCA to solve this problem and obtained 100% accuracy. We do not know what dimensions of the ambient space and the subspaces they used. We also apply GPCA to segment d-dimensional linear subspaces in the projected space $\mathbb{R}^D$, where $1 \leq d < D \leq 10$. However, we did not find any combination that leads to a perfect segmentation.

Chapter 7

Conclusion and Future Work

We first proposed the Theoretical Spectral Curvature Clustering (TSCC) algorithm (Algorithm 1) for solving the problem of hybrid linear modeling, and then analyzed the theoretical performance of SCC in the setting of Problem 1. We showed that the TSCC algorithm could precisely cluster the underlying components knowing the perfect tensor (Proposition 3.1.1), and established good performance in the case of reasonable deviation from the perfect case (Theorem 3.2.4). Using this result, we proved that if a data set is sampled independently and identically according to the setting of Problem 1, then with high sampling probability the TSCC algorithm will perform well as long as the underlying distributions are sufficiently flat and separated (Theorem 4.2.1).

We next introduced various techniques to make the algorithm practical, forming the Spectral Curvature Clustering (SCC) algorithm (Algorithm 2). The complexity of SCC, i.e., the storage and running time, depends linearly on both the size of the data and the ambient dimension. We performed extensive simulations to compare our algorithm with a few other standard methods. It seemed that our algorithm is at least comparable to its competitors. It has a marked advantage in the case of affine subspaces and in certain instances of mixed dimensions. We also applied our algorithm to several real-world

problems, and obtained satisfactory results in all cases. Our algorithm performed well even in relatively high dimensional projected spaces, sometimes including the full space, and thus did not require aggressive dimensionality reduction as other algorithms.

We conclude this paper by discussing both the open directions and the possible extensions of this work.

Further understanding of the two normalizations discussed in Section 3.3.1: We first explored in Section 3.3.1 possible normalizations of the matrix $\mathbf{U}$, and analyzed (to some extent) the performance of TSCC with and without them. We concluded that the normalization suggested by the matrix $\mathbf{T}$ is probably the right one to apply in TSCC. It will be interesting to test our practical strategy when applying such a normalization (see Remark 3.3.1) on both artificial and practical data sets with varying numbers of points within each cluster. Also, we wish to study more carefully the possible advantages of the normalization suggested by the matrix $\mathbf{V}$.

At last, Section 3.3.1 analyzed the TSCC algorithm when applied without the unnormalized matrix $\mathbf{Z}$. The perturbation results there were practically comparable to those obtained when applying TSCC with the normalized matrix $\mathbf{Z}$. It thus did not reveal the significant advantage of using $\mathbf{Z}$. In future investigations we would like to improve the current estimates so that they emphasize this significant advantage.

Further interpretation of the incidence constant: Currently we have described the behavior of the incidence constant in a few typical examples of two intersecting linear subspaces. We ask about characterization of this constant for general mixtures of flats, and its dependence on the separation between the subspaces, the magnitude of noise as well as the tuning parameter.

Estimation of the clustering identification error: We showed in Section 3.3.1 that when $K = 2$ and $\mathrm{TV}(\mathbf{U})$ is sufficiently small, then a large percentage of the points can

be clustered correctly. We would like to extend the corresponding analysis to the case where $K > 2$.

Further investigation of Assumption 1: Assumption 1 is a crucial condition for Algorithm 1 to work well. Our partial results (i.e., Lemmas 4.4.1 and 4.4.2) showed that this assumption holds at least in expectation. We would like to explore the existence in high probability of Assumption 1 with a constant $\varepsilon_2 > 0$ that does not contradict the bounds imposed by Theorem 4.2.1 (see discussion in Section 4.4, in particular, Remark 4.4.1).

Justification of Iterative Sampling: Our heuristic idea of iterative sampling seems to work well in all cases and thus results in a fast and accurate algorithm. We are interested in a more rigorous foundation of this procedure, in particular, finding conditions under which it converges (e.g., how large c should be to ensure convergence).

Thorough Study of Robustness: Numerical experiments indicate that the SCC algorithm (without isolating outliers in each iteration) is robust to outliers. We would like to pursue a theoretical justification of robustness of the SCC algorithm (or TSCC). We are also interested in improving the strategy for detecting outliers, especially when the outlier percentage is not given.

Improving the Case of Mixed Dimensions: Currently, when dealing with mixed dimensions, we use the highest dimension. This strategy works well in terms of $e_{\%}$. To improve the performance of SCC in this case, and consequently to more accurately evaluate the other error e_{OLS}, we plan to explore estimation of the true dimensions of the detected flats. Another strategy might be to hierarchically perform SCC according to different intrinsic dimensions.

Determining the Number of Flats and Their Dimensions: Throughout this

paper we have assumed that K and d_k are given. In many cases prior knowledge of these parameters may not be available. We thus need to develop techniques and criterions to select an optimal model.

References

[1] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12), 2005.

[2] Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50(3):413–458, 2008.

[3] A. Szlam. Modifications on k q-flats for supervised learning. http://www.math.ucla.edu/∼aszlam/kplanes.pdf, 2008.

[4] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Proc. CVPR*, Alaska, June 2008.

[5] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, June 1981.

[6] P. H. S. Torr. Geometric motion segmentation and model selection. *Phil. Trans. R. Soc. Lond. A*, 356:1321–1340, 1998.

[7] A. Y. Yang, S. R. Rao, and Y. Ma. Robust statistical estimation and segmentation of multiple subspaces. In *Computer Vision and Pattern Recognition Workshop*, June 2006.

[8] A. Kambhatla and T. Leen. Fast non-linear dimension reduction. In *Advances in Neural Information Processing Systems 6*, pages 152–159, 1994.

[9] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 11–18, 2003.

[10] P. Bradley and O. Mangasarian. k-plane clustering. *J. Global optim.*, 16(1):23–32, 2000.

[11] P. Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, April 2000.

[12] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[13] K. Kanatani. Motion segmentation by subspace separation and model selection. In *Proc. of 8th ICCV*, volume 3, pages 586–591. Vancouver, Canada, 2001.

[14] K. Kanatani. Evaluation and selection of models for motion segmentation. In *7th ECCV*, volume 3, pages 335–349, May 2002.

[15] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[16] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Netw.*, 13(4-5):411–430, 2000.

[17] G. Medioni, M.-S. Lee, and C.-K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.

[18] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie. Beyond pairwise clustering. In *Proceedings of the 2005 IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 2, pages 838–845, 2005.

[19] V. Govindu. A tensor decomposition for geometric grouping and segmentation. In *CVPR*, volume 1, pages 1150–1157, June 2005.

[20] A. Shashua, R. Zass, and T. Hazan. Multi-way clustering using super-symmetric non-negative tensor factorization. In *ECCV06*, volume IV, pages 595–608, 2006.

[21] S. Agarwal, K. Branson, and S. Belongie. Higher order learning with graphs. In *Proceedings of the 23rd International Conference on Machine learning*, volume 148, pages 17–24, 2006.

[22] R. Souvenir and R. Pless. Manifold clustering. In *the 10th International Conference on Computer Vision (ICCV 2005)*, 2005.

[23] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and nondegenerate. In *ECCV*, volume 4, pages 94–106, 2006.

[24] P. Gruber and F. Theis. Grassmann clustering. In *Proc. EUSIPCO 2006*, Florence, Italy, 2006.

[25] D. Kushnir, M. Galun, and A. Brandt. Fast multiscale clustering and manifold identification. *Pattern Recognition*, 39(10):1876–1891, October 2006.

[26] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1546–1562, September 2007.

[27] G. Haro, G. Randall, and G. Sapiro. Translated Poisson mixture model for stratification learning. *Int. J. Comput. Vision*, 80(3):358–374, 2008.

[28] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[29] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856, 2001.

[30] G. Lerman and J. T. Whitehouse. High-dimensional Menger-type curvatures - part I: Geometric multipoles and multiscale inequalities. Available from http://arxiv.org/abs/0805.1425v1.

[31] G. Lerman and J. T. Whitehouse. Least squares approximations for probability distributions via multi-way curvatures. In preparation.

[32] G. Lerman and J. T. Whitehouse. On d-dimensional d-semimetrics and simplex-type inequalities for high-dimensional sine functions. *Journal of Approximation Theory*, 2008. Also available from http://dx.doi.org/10.1016/j.jat.2008.03.005.

[33] G. Lerman and J. T. Whitehouse. High-dimensional Menger-type curvatures - part II: d-separation and a menagerie of curvatures. Accepted for publication in the Journal of Constructive Approximation. Available from http://arxiv.org/abs/0809.0137v1.

[34] B. Bader and T. Kolda. Matlab tensor classes for fast algorithm prototyping. Technical Report SAND2004-5187, Sandia National Laboratories, October 2004.

[35] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. A.*, 21(4):1253–1278, 2000.

[36] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics*

and Probability, volume 1, pages 281–297. University of California Press, Berkeley, CA, 1967.

[37] G. Golub and C. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, Maryland, 1996.

[38] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, January 2003.

[39] U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, 36(2):555–586, 2008.

[40] P. Drineas, R. Kannan, and M. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM J. Comput.*, 36(1):132–157, 2006.

[41] M. Brand. Fast online SVD revisions for lightweight recommender systems. In *Proc. SIAM International Conference on Data Mining*, 2003.

[42] Y. Sugaya and K. Kanatani. Multi-stage unsupervised learning for multi-body motion segmentation. *IEICE Transactions on Information and Systems*, E87-D(7):1935–1942, 2004.

[43] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, February 2003.

[44] R. Epstein, P. Hallinan, and A. Yuille. 5 ± 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models. In *IEEE Workshop on Physics-based Modeling in Computer Vision*, pages 108–116, June 1995.

[45] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[46] L. Zwald and G. Blanchard. On the convergence of eigenspaces in kernel principal components analysis. In *Advances in Neural Information Processing Systems 18*, pages 1649–1656, 2005.

[47] C. McDiarmid. On the method of bounded differences. In *Surveys in combinatorics*, pages 148–188. Cambridge University Press, 1989.

Appendix A

Proofs

A.1 Proof of Proposition 3.1.1

The affinity matrix $\widetilde{\mathbf{A}}$, the matricized version of $\widetilde{\mathcal{A}}$, is a 0/1 matrix of size $N \times N^{d+1}$. We identify the unit entries in each row as follows. For any fixed $1 \leq i \leq N_1$, the entries of the i-th row of $\widetilde{\mathbf{A}}$ are of the form $\widetilde{\mathcal{A}}(i, i_2, \ldots, i_{d+2}), 1 \leq i_2, \ldots, i_{d+2} \leq N$. These entries will be 1 if they represent affinities of distinct $d+2$ points in $\widetilde{\mathrm{C}}_1$, that is, the indices $i, i_2, \ldots, i_{d+2}$ are distinct and between 1 and N_1. Therefore, the i-th row has exactly $\mathrm{P}(N_1 - 1, d+1)$ entries filled by a 1, which is exactly the number of permutations of size $d+1$ out of the first N_1 points excluding i. Similarly, each of the subsequent N_2 rows has $\mathrm{P}(N_2 - 1, d+1)$ ones, and each of the next N_3 rows has $\mathrm{P}(N_3 - 1, d+1)$ ones, etc..

The weight matrix $\widetilde{\mathbf{W}} = \widetilde{\mathbf{A}}\widetilde{\mathbf{A}}'$ can be expressed in terms of the tensor $\widetilde{\mathcal{A}}$ in the following way:

$$\widetilde{W}_{ij} = \sum_{1 \leq i_2, \ldots, i_{d+2} \leq N} \widetilde{\mathcal{A}}(i, i_2, \ldots, i_{d+2}) \widetilde{\mathcal{A}}(j, i_2, \ldots, i_{d+2}), \quad 1 \leq i, j \leq N. \tag{A.1}$$

If $\mathbf{x}_i$ and $\mathbf{x}_j$ are not in the same underlying cluster, then all the products are zero.

Therefore, $\widetilde{\mathbf{W}}$ is block diagonal:

$$\widetilde{\mathbf{W}} = \mathrm{diag}\{\widetilde{\mathbf{W}}^{(1)}, \widetilde{\mathbf{W}}^{(2)}, \ldots, \widetilde{\mathbf{W}}^{(K)}\}, \tag{A.2}$$

where $\widetilde{\mathbf{W}}^{(k)} \in \mathbb{R}^{N_k \times N_k}$, corresponding to the underlying cluster $\widetilde{\mathrm{C}}_k$, has the following form:

$$\widetilde{W}_{ij}^{(k)} = \begin{cases} \mathrm{P}(N_k - 1, d+1), & \text{if } i = j; \\ \mathrm{P}(N_k - 2, d+1), & \text{otherwise.} \end{cases} \tag{A.3}$$

Indeed, the diagonal elements of $\widetilde{\mathbf{W}}^{(k)}$ are simply the number of ones in the corresponding rows of $\widetilde{\mathbf{A}}$, and the off-diagonal elements are the number of ones that appear at the intersection of the corresponding pair of rows.

It then follows that

$$\widetilde{\mathbf{D}} = \mathrm{diag}\{\widetilde{\mathbf{W}} \cdot \mathbf{1}\} = \mathrm{diag}\{\widetilde{d}_1 \mathbf{I}_{N_1}, \widetilde{d}_2 \mathbf{I}_{N_2}, \ldots, \widetilde{d}_K \mathbf{I}_{N_K}\}, \tag{A.4}$$

where

$$\begin{aligned} \widetilde{d}_k &= \mathrm{P}(N_k - 1, d+1) + (N_k - 1) \cdot \mathrm{P}(N_k - 2, d+1) \\ &= (N_k - d - 1) \cdot \mathrm{P}(N_k - 1, d+1). \end{aligned} \tag{A.5}$$

The normalized matrix $\widetilde{\mathbf{Z}} = \widetilde{\mathbf{D}}^{-1/2} \widetilde{\mathbf{W}} \widetilde{\mathbf{D}}^{-1/2}$ is also block diagonal:

$$\widetilde{\mathbf{Z}} = \mathrm{diag}\{\widetilde{\mathbf{Z}}^{(1)}, \widetilde{\mathbf{Z}}^{(2)}, \ldots, \widetilde{\mathbf{Z}}^{(K)}\}, \tag{A.6}$$

where each block has the form $\widetilde{\mathbf{Z}}^{(k)} = \widetilde{\mathbf{W}}^{(k)} / \widetilde{d}_k, 1 \le k \le K$. The (i,j)-element of $\widetilde{\mathbf{Z}}^{(k)}$, for all $1 \le i, j \le N_k$, is

$$\widetilde{Z}_{ij}^{(k)} = \begin{cases} \frac{1}{N_k - d - 1}, & \text{if } i = j; \\ \frac{N_k - d - 2}{(N_k - 1)(N_k - d - 1)}, & \text{otherwise.} \end{cases} \tag{A.7}$$

Straightforward calculation shows that each block matrix $\widetilde{\mathbf{Z}}^{(k)}$ has two distinct eigenvalues:

$$\widetilde{\lambda}_n^{(k)} = \begin{cases} 1, & \text{if } n = 1; \\ \frac{d+1}{(N_k-1)(N_k-d-1)}, & \text{if } 2 \le n \le N_k. \end{cases} \tag{A.8}$$

The eigenspace associated with the single eigenvalue 1 for $\widetilde{\mathbf{Z}}^{(k)}$ is spanned by $\mathbf{1}_{N_k}$, the N_k-dimensional column vector of all ones. Since the eigenvalues and eigenvectors of a block diagonal matrix are essentially the union of those of its blocks (for eigenvectors we need to append zeros in an appropriate way), we conclude that $\widetilde{\mathbf{Z}}$ has the largest eigenvalue 1 of multiplicity K with associated eigenspace spanned by the following K orthonormal vectors:

$$\frac{1}{\sqrt{N_1}} \begin{pmatrix} \mathbf{1}_{N_1} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \frac{1}{\sqrt{N_2}} \begin{pmatrix} \mathbf{0} \\ \mathbf{1}_{N_2} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \ldots, \frac{1}{\sqrt{N_K}} \begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{1}_{N_K} \end{pmatrix} \in \mathbb{R}^N. \tag{A.9}$$

We note that the K eigenvectors associated with the eigenvalue 1 can only be determined up to an orthonormal transformation. That is,

$$\widetilde{\mathbf{U}} = \begin{pmatrix} \frac{1}{\sqrt{N_1}}\mathbf{1}_{N_1} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{N_2}}\mathbf{1}_{N_2} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \frac{1}{\sqrt{N_K}}\mathbf{1}_{N_K} \end{pmatrix} \mathbf{Q} \in \mathbb{R}^{N \times K}, \tag{A.10}$$

where $\mathbf{Q}$ is a $K \times K$ orthonormal matrix.

If we write $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_K)'$, where $\mathbf{q}_k$ is the k-th column of $\mathbf{Q}'$, then equation (A.10) implies that the K clusters are mapped one-to-one to the K mutually orthogonal vectors $\frac{1}{\sqrt{N_1}} \cdot \mathbf{q}_1, \ldots, \frac{1}{\sqrt{N_K}} \cdot \mathbf{q}_K \in \mathbb{R}^K$.

A.2 Proof of Lemma 3.2.3

We first note that $P^K(\mathbf{Z}) = \mathbf{U}\mathbf{U}'$ and $P^K(\widetilde{\mathbf{Z}}) = \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}'$, due to the fact that both $\mathbf{U}$ and $\widetilde{\mathbf{U}}$ are composed of orthonormal columns. Therefore,

$$\begin{aligned}\left\|P^K(\mathbf{Z}) - P^K(\widetilde{\mathbf{Z}})\right\|_{\mathrm{F}}^2 &= \left\|\mathbf{U}\mathbf{U}' - \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}'\right\|_{\mathrm{F}}^2 = \operatorname{trace}\left(\left(\mathbf{U}\mathbf{U}' - \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}'\right)^2\right) \\ &= \operatorname{trace}\left(\mathbf{U}\mathbf{U}' - \mathbf{U}\mathbf{U}'\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}' - \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}'\mathbf{U}\mathbf{U}' + \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}'\right). \qquad \text{(A.11)}\end{aligned}$$

Since

$$\operatorname{trace}\left(\mathbf{U}\mathbf{U}'\right) = \operatorname{trace}\left(\mathbf{U}'\mathbf{U}\right) = \operatorname{trace}(\mathbf{I}_K) = K, \qquad \text{(A.12)}$$

and similarly,

$$\operatorname{trace}\left(\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}'\right) = K, \qquad \text{(A.13)}$$

we have

$$\left\|P^K(\mathbf{Z}) - P^K(\widetilde{\mathbf{Z}})\right\|_{\mathrm{F}}^2 = 2K - 2 \cdot \operatorname{trace}\left(\mathbf{U}\mathbf{U}'\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}'\right). \qquad \text{(A.14)}$$

In the formula of the matrix $\widetilde{\mathbf{U}}$ (see equation (A.10)), there is an arbitrary orthonormal matrix $\mathbf{Q}$. However, the product $\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}'$ does not depend on $\mathbf{Q}$. Hence, we can use a representation of $\widetilde{\mathbf{U}}$ where $\mathbf{Q}$ is the identity matrix, and proceed as follows:

$$\begin{aligned}\left\|P^K(\mathbf{Z}) - P^K(\widetilde{\mathbf{Z}})\right\|_{\mathrm{F}}^2 &= 2K - 2 \cdot \left\|\mathbf{U}'\widetilde{\mathbf{U}}\right\|_{\mathrm{F}}^2 \\ &= 2K - 2 \cdot \left\|\left[\sum_{i\in \mathrm{I}_1} \frac{1}{\sqrt{N_1}}\left(\mathbf{u}^{(i)}\right)' \cdots \sum_{i\in \mathrm{I}_K} \frac{1}{\sqrt{N_K}}\left(\mathbf{u}^{(i)}\right)'\right]\right\|_{\mathrm{F}}^2 \\ &= 2K - 2 \cdot \sum_{k=1}^{K} \frac{1}{N_k}\left\|\sum_{i\in \mathrm{I}_k} \mathbf{u}^{(i)}\right\|_2^2 \\ &= 2K - 2 \cdot \sum_{k=1}^{K} N_k \left\|\mathbf{c}^{(k)}\right\|_2^2. \qquad \text{(A.15)}\end{aligned}$$

Since the columns of the matrix $\mathbf{U}$ are unit vectors, we have

$$\sum_{i=1}^{N} \left\| \mathbf{u}^{(i)} \right\|_2^2 = \|\mathbf{U}\|_{\mathrm{F}}^2 = \sum_{k=1}^{K} \|\mathbf{u}_k\|_2^2 = K. \tag{A.16}$$

Combining the last two equations we get that

$$\begin{aligned} \left\| P^K(\mathbf{Z}) - P^K(\widetilde{\mathbf{Z}}) \right\|_{\mathrm{F}}^2 &= 2 \cdot \left(\sum_{i=1}^{N} \left\| \mathbf{u}^{(i)} \right\|_2^2 - \sum_{k=1}^{K} N_k \cdot \left\| \mathbf{c}^{(k)} \right\|_2^2 \right) \\ &= 2 \cdot \sum_{k=1}^{K} \left(\sum_{i \in \mathrm{I}_k} \left\| \mathbf{u}^{(i)} \right\|_2^2 - N_k \cdot \left\| \mathbf{c}^{(k)} \right\|_2^2 \right) \\ &= 2 \cdot \sum_{k=1}^{K} \sum_{i \in \mathrm{I}_k} \left\| \mathbf{u}^{(i)} - \mathbf{c}^{(k)} \right\|_2^2. \end{aligned} \tag{A.17}$$

A.3 Proof of Lemma 3.2.1

Equation (3.7) is a direct consequence of combining equation (A.15) and Lemma 3.2.3.

To show equation (3.8), we first expand the following two products

$$\mathbf{U}\mathbf{U}' = \left(\langle \mathbf{u}^{(i)}, \mathbf{u}^{(j)} \rangle \right)_{1 \le i,j \le N}, \tag{A.18}$$

$$\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}' = \operatorname{diag}\left\{ \frac{1}{N_1} \mathbf{1}_{N_1 \times N_1}, \ldots, \frac{1}{N_K} \mathbf{1}_{N_K \times N_K} \right\}. \tag{A.19}$$

Then

$$\begin{aligned} &\left\| P^K(\mathbf{Z}) - P^K(\widetilde{\mathbf{Z}}) \right\|_{\mathrm{F}}^2 = \left\| \mathbf{U}\mathbf{U}' - \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}' \right\|_{\mathrm{F}}^2 \\ &= \sum_{1 \le k \le K} \sum_{i,j \in \mathrm{I}_k} \left(\langle \mathbf{u}^{(i)}, \mathbf{u}^{(j)} \rangle - \frac{1}{N_k} \right)^2 + \sum_{1 \le k \ne \ell \le K} \sum_{i \in \mathrm{I}_k, j \in I_\ell} \left(\langle \mathbf{u}^{(i)}, \mathbf{u}^{(j)} \rangle \right)^2 \\ &\ge \sum_{1 \le k \ne \ell \le K} \sum_{i \in \mathrm{I}_k, j \in I_\ell} \left(\langle \mathbf{u}^{(i)}, \mathbf{u}^{(j)} \rangle \right)^2. \end{aligned} \tag{A.20}$$

We next apply the inequality $\left(\sum_{i=1}^m a_i\right)^2 \leq m \cdot \sum_{i=1}^m a_i^2$ and conclude that

$$\begin{aligned}\left\|P^K(\mathbf{Z}) - P^K(\widetilde{\mathbf{Z}})\right\|_{\mathrm{F}}^2 &\geq \sum_{1\leq k\neq \ell\leq K} \frac{1}{N_k N_\ell} \cdot \left(\sum_{i\in \mathrm{I}_k, j\in I_\ell} \langle \mathbf{u}^{(i)}, \mathbf{u}^{(j)}\rangle\right)^2 \\ &= \sum_{1\leq k\neq \ell\leq K} N_k N_\ell \cdot \langle \mathbf{c}^{(k)}, \mathbf{c}^{(\ell)}\rangle^2. \end{aligned} \tag{A.21}$$

Finally, combining the last equation and Lemma 3.2.3 completes the proof.

A.4 Proof of Lemma 3.2.2

From the proof of Lemma 3.2.3 we have that

$$\begin{aligned}\left\|P^K(\mathbf{Z}) - P^K(\widetilde{\mathbf{Z}})\right\|_{\mathrm{F}}^2 &= 2K - 2\left\|\mathbf{U}'\widetilde{\mathbf{U}}\right\|_{\mathrm{F}}^2 = 2K - 2\sum_{k=1}^K \sigma_k^2\left(\mathbf{U}'\widetilde{\mathbf{U}}\right) \\ &= 2K - 2\sum_{k=1}^K \cos^2\theta_k = 2\sum_{k=1}^K \sin^2\theta_k. \end{aligned} \tag{A.22}$$

A.5 Proof of Theorem 3.2.4

The proof is based on a perturbation result by Zwald and Blanchard [46, Theorem 3]. In fact, we only need a special case of it which is formulated below.

Theorem A.5.1 (Matrix version of Theorem 3 in Zwald and Blanchard [46])**.** *Let* $\mathbf{S}$ *be a symmetric positive square matrix with nonzero eigenvalues* $\lambda_1 \geq \cdots \geq \lambda_K > \lambda_{K+1} \geq \cdots \geq 0$*, where* $K > 0$ *is an integer. Define* $\delta_K = \lambda_K - \lambda_{K+1} > 0$*, which denotes the* K^{th} *eigengap of* $\mathbf{S}$*. Let* $\mathbf{B}$ *be another symmetric matrix such that* $\|\mathbf{B}\|_{\mathrm{F}} < \delta_K/4$ *and* $\mathbf{S}+\mathbf{B}$ *is still a positive matrix. Then*

$$\left\|P^K(\mathbf{S}+\mathbf{B}) - P^K(\mathbf{S})\right\|_{\mathrm{F}} \leq 2\|\mathbf{B}\|_{\mathrm{F}}/\delta_K. \tag{A.23}$$

In order to apply the above theorem to the quantity $\left\|P^K(\mathbf{Z}) - P^K(\widetilde{\mathbf{Z}})\right\|_{\mathrm{F}}$, we need a lower bound on $\widetilde{\delta}_K$, the K^{th} eigengap of $\widetilde{\mathbf{Z}}$, and an upper bound on the Frobenius

norm of the difference $\mathbf{B} := \mathbf{Z} - \widetilde{\mathbf{Z}}$. While the former bound is immediate, we find the latter bound somewhat challenging.

First, equation (A.8), together with $N_1 = \min_{1 \le k \le K} N_k$, implies that:

$$\widetilde{\delta}_K = 1 - \frac{d+1}{(N_1 - 1)(N_1 - d - 1)}. \tag{A.24}$$

Since $N_1 \ge 2(d+1) + 1$ by equation (3.4), we then obtain that

$$\widetilde{\delta}_K \ge \frac{2d+3}{2d+4} \ge \frac{3}{4}. \tag{A.25}$$

Next, we estimate the Frobenius norm of the perturbation $\mathbf{B}$ as follows. Using the definitions of the matrices $\mathbf{Z}$ and $\mathbf{W}$, we rewrite $\mathbf{B}$ in the following way:

$$\mathbf{B} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{A}'\mathbf{D}^{-1/2} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}'\widetilde{\mathbf{D}}^{-1/2}. \tag{A.26}$$

Regrouping terms gives that

$$\begin{aligned}\mathbf{B} = &\left(\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right)\left(\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right)' \\ &+ \left(\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right)\widetilde{\mathbf{A}}'\widetilde{\mathbf{D}}^{-1/2} + \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\left(\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right)'. \end{aligned} \tag{A.27}$$

We thus get an initial upper bound on its Frobenius norm:

$$\|\mathbf{B}\|_{\mathrm{F}} \le \left\|\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}^2 + 2\left\|\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}\left\|\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}. \tag{A.28}$$

By using equations (A.6) and (A.7), we get that

$$\left\|\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}^2 = \operatorname{trace}\left(\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{W}}\widetilde{\mathbf{D}}^{-1/2}\right) = \operatorname{trace}\left(\widetilde{\mathbf{Z}}\right) = \sum_{k=1}^{K} \frac{N_k}{N_k - d - 1}. \tag{A.29}$$

Equation (3.4) implies that

$$\frac{N_k}{N_k - d - 1} < 2, \quad 1 \le k \le K. \tag{A.30}$$

Consequently, we have

$$\left\|\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}^2 < 2K, \tag{A.31}$$

and thus equation (A.28) becomes

$$\|\mathbf{B}\|_{\mathrm{F}} < \left\|\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}^2 + 2\sqrt{2K}\cdot\left\|\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}. \tag{A.32}$$

Therefore, in order to control $\|\mathbf{B}\|_{\mathrm{F}}$, we only need to bound $\left\|\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}$.

Let

$$\mathbf{E} := \mathbf{A} - \widetilde{\mathbf{A}}. \tag{A.33}$$

Replacing $\mathbf{A}$ with $\widetilde{\mathbf{A}} + \mathbf{E}$ yields that

$$\begin{aligned}\left\|\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}} &= \left\|\left(\mathbf{D}^{-1/2} - \widetilde{\mathbf{D}}^{-1/2}\right)\widetilde{\mathbf{A}} + \mathbf{D}^{-1/2}\mathbf{E}\right\|_{\mathrm{F}} \\ &\le \left\|\left(\mathbf{D}^{-1/2} - \widetilde{\mathbf{D}}^{-1/2}\right)\widetilde{\mathbf{A}}\right\|_{\mathrm{F}} + \left\|\mathbf{D}^{-1/2}\mathbf{E}\right\|_{\mathrm{F}}. \end{aligned} \tag{A.34}$$

The second term on the right hand side of equation (A.34) is bounded as follows

$$\begin{aligned}\left\|\mathbf{D}^{-1/2}\mathbf{E}\right\|_{\mathrm{F}} &\le \left\|\mathbf{D}^{-1/2}\right\|_2 \cdot \|\mathbf{E}\|_{\mathrm{F}} \le \left\|(\varepsilon_2\widetilde{\mathbf{D}})^{-1/2}\right\|_2 \cdot \|\mathbf{E}\|_{\mathrm{F}} \\ &= \left(\varepsilon_2\widetilde{d}_1\right)^{-1/2} \cdot \|\mathbf{E}\|_{\mathrm{F}}, \end{aligned} \tag{A.35}$$

in which the second inequality follows from Assumption 1 ($\mathbf{D} \ge \varepsilon_2\widetilde{\mathbf{D}} > 0$), and the last equality is due to our convention: $N_1 = \min_{1\le k\le K} N_k$ (which implies that $\widetilde{d}_1 = \min_{1\le k\le K} \widetilde{d}_k$).

Bounding the first term of the right hand side of equation (A.34) requires more work. We estimate it as follows:

$$\begin{aligned}\left\|\left(\mathbf{D}^{-1/2} - \widetilde{\mathbf{D}}^{-1/2}\right)\cdot\widetilde{\mathbf{A}}\right\|_{\mathrm{F}} &= \left\|\widetilde{\mathbf{D}}^{-1/2}\mathbf{D}^{-1/2}\left(\mathbf{D}^{1/2} + \widetilde{\mathbf{D}}^{1/2}\right)^{-1}\left(\mathbf{D} - \widetilde{\mathbf{D}}\right)\cdot\widetilde{\mathbf{A}}\right\|_{\mathrm{F}} \\ &\le \left\|\widetilde{\mathbf{D}}^{-1/2}\left(\varepsilon_2\widetilde{\mathbf{D}}\right)^{-1/2}\left(\widetilde{\mathbf{D}}^{1/2}\right)^{-1}\left(\mathbf{D} - \widetilde{\mathbf{D}}\right)\cdot\widetilde{\mathbf{A}}\right\|_{\mathrm{F}} \\ &= \varepsilon_2^{-1/2}\left\|\widetilde{\mathbf{D}}^{-3/2}\left(\mathbf{D} - \widetilde{\mathbf{D}}\right)\cdot\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}. \end{aligned} \tag{A.36}$$

We proceed by using the index sets $\mathrm{I}_1, \ldots, \mathrm{I}_K$ (see equation (3.1)) to expand the last

equation:

$$\left\|\left(\mathbf{D}^{-1/2}-\widetilde{\mathbf{D}}^{-1/2}\right)\cdot\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}\leq\varepsilon_2^{-1/2}\sqrt{\sum_{1\leq k\leq K}\sum_{i\in\mathrm{I}_k}\left(D_{ii}-\widetilde{d}_k\right)^2\widetilde{d}_k^{-3}\cdot\left\|\widetilde{\mathbf{A}}(i,:)\right\|_2^2}$$
$$=\varepsilon_2^{-1/2}\sqrt{\sum_{1\leq k\leq K}\sum_{i\in\mathrm{I}_k}\frac{\left(D_{ii}-\widetilde{d}_k\right)^2}{(N_k-d-1)\cdot\widetilde{d}_k^2}}$$
$$\leq\varepsilon_2^{-1/2}\widetilde{d}_1^{-1}(N_1-d-1)^{-1/2}\cdot\left\|\mathbf{D}-\widetilde{\mathbf{D}}\right\|_{\mathrm{F}}. \tag{A.37}$$

Using the definitions of $\mathbf{D}$ and $\widetilde{\mathbf{D}}$, we obtain that

$$\left\|\mathbf{D}-\widetilde{\mathbf{D}}\right\|_{\mathrm{F}}=\left\|\left(\mathbf{W}-\widetilde{\mathbf{W}}\right)\cdot\mathbf{1}_N\right\|_2\leq\left\|\mathbf{W}-\widetilde{\mathbf{W}}\right\|_{\mathrm{F}}\cdot\|\mathbf{1}_N\|_2$$
$$=N^{1/2}\cdot\left\|\widetilde{\mathbf{A}}\mathbf{E}'+\mathbf{E}\widetilde{\mathbf{A}}'+\mathbf{E}\mathbf{E}'\right\|_{\mathrm{F}}$$
$$\leq N^{1/2}\cdot\left(2\left\|\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}\|\mathbf{E}\|_{\mathrm{F}}+\|\mathbf{E}\|_{\mathrm{F}}^2\right). \tag{A.38}$$

Combining equations (A.37) and (A.38) and applying $N_1-d-1>\frac{N_1}{2}\geq\frac{\varepsilon_1 N}{2K}$ (following equation (3.4)) gives that

$$\left\|\left(\mathbf{D}^{-1/2}-\widetilde{\mathbf{D}}^{-1/2}\right)\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}\leq\left(\frac{2K}{\varepsilon_1\varepsilon_2}\right)^{1/2}\widetilde{d}_1^{-1}\left(2\left\|\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}\|\mathbf{E}\|_{\mathrm{F}}+\|\mathbf{E}\|_{\mathrm{F}}^2\right). \tag{A.39}$$

By substituting equations (A.35) and (A.39) into equation (A.34), we arrive at

$$\left\|\mathbf{D}^{-1/2}\mathbf{A}-\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}\leq\left(\frac{2K}{\varepsilon_1\varepsilon_2}\right)^{1/2}\widetilde{d}_1^{-1}\left(2\left\|\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}\|\mathbf{E}\|_{\mathrm{F}}+\|\mathbf{E}\|_{\mathrm{F}}^2\right)$$
$$+\varepsilon_2^{-1/2}\widetilde{d}_1^{-1/2}\|\mathbf{E}\|_{\mathrm{F}}. \tag{A.40}$$

In order to complete the above estimate for $\left\|\mathbf{D}^{-1/2}\mathbf{A}-\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}$, we need to estimate $\left\|\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}$ from above

$$\left\|\widetilde{\mathbf{A}}\right\|_{\mathrm{F}}<\sqrt{N^{d+2}}=N^{(d+2)/2}, \tag{A.41}$$

and $\widetilde{d}_1$ from below

$$\widetilde{d}_1=(N_1-d-1)\cdot\mathrm{P}(N_1-1,d+1)\geq(N_1/2)^{d+2}\geq\left(\frac{\varepsilon_1 N}{2K}\right)^{d+2}. \tag{A.42}$$

We also note that all the elements of the matrix $\mathbf{E}$ are between -1 and 1, and thus

$$\|\mathbf{E}\|_{\mathrm{F}} \leq N^{(d+2)/2}. \tag{A.43}$$

We then continue from equation (A.40), together with the last three estimates, and get that

$$\begin{aligned}
&\left\|\mathbf{D}^{-1/2}\mathbf{A} - \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\right\|_{\mathrm{F}} \\
&\leq \left(\frac{2K}{\varepsilon_1\varepsilon_2}\right)^{1/2} \left(\frac{\varepsilon_1 N}{2K}\right)^{-(d+2)} 3N^{(d+2)/2} \|\mathbf{E}\|_{\mathrm{F}} + \varepsilon_2^{-1/2} \left(\frac{\varepsilon_1 N}{2K}\right)^{-(d+2)/2} \|\mathbf{E}\|_{\mathrm{F}} \\
&\leq 4\varepsilon_2^{-1/2} \left(\frac{2K}{\varepsilon_1}\right)^{d+5/2} N^{-(d+2)/2} \|\mathbf{E}\|_{\mathrm{F}}\,.
\end{aligned} \tag{A.44}$$

Finally, it follows from equations (A.32) and (A.44) that

$$\|\mathbf{B}\|_{\mathrm{F}} \leq C_0(K, d, \varepsilon_1, \varepsilon_2) \cdot N^{-(d+2)/2} \|\mathbf{E}\|_{\mathrm{F}}\,, \tag{A.45}$$

where

$$C_0(K, d, \varepsilon_1, \varepsilon_2) := 16\varepsilon_2^{-1} \left(\frac{2K}{\varepsilon_1}\right)^{2d+5} + 2\sqrt{2K} \cdot 4\varepsilon_2^{-1/2} \left(\frac{2K}{\varepsilon_1}\right)^{d+5/2}. \tag{A.46}$$

By combining Theorem A.5.1 with equations (A.25) and (A.45), we obtain that when

$$C_0(K, d, \varepsilon_1, \varepsilon_2) \cdot N^{-(d+2)/2} \|\mathbf{E}\|_{\mathrm{F}} < 3/16, \tag{A.47}$$

then

$$\left\|P^K(\mathbf{Z}) - P^K(\widetilde{\mathbf{Z}})\right\|_{\mathrm{F}} \leq 8/3 \cdot C_0(K, d, \varepsilon_1, \varepsilon_2) \cdot N^{-(d+2)/2} \|\mathbf{E}\|_{\mathrm{F}}\,. \tag{A.48}$$

Letting

$$C_1(K, d, \varepsilon_1, \varepsilon_2) := 32/9 \cdot C_0^2(K, d, \varepsilon_1, \varepsilon_2), \tag{A.49}$$

and noting

$$\|\mathbf{E}\|_{\mathrm{F}} \equiv \|\mathcal{E}\|_{\mathrm{F}}\,, \tag{A.50}$$

we complete the proof by combining Lemma 3.2.3 and equations (A.48) and (A.49).

A.6 Proof of Lemma 3.3.2

In the $\mathbf{T}$ space the centers of the underlying clusters are

$$\mathbf{c}_{\mathbf{T}}^{(k)} := \sqrt{N_k} \cdot \mathbf{c}^{(k)}, \quad 1 \le k \le K. \tag{A.51}$$

Applying Lemma 3.2.1 with $K = 2$ gives that

$$\begin{aligned} \left\| \mathbf{c}_{\mathbf{T}}^{(1)} - \mathbf{c}_{\mathbf{T}}^{(2)} \right\|_2^2 &= N_1 \cdot \left\| \mathbf{c}^{(1)} \right\|_2^2 + N_2 \cdot \left\| \mathbf{c}^{(2)} \right\|_2^2 - 2\sqrt{N_1 N_2} \cdot \langle \mathbf{c}^{(1)}, \mathbf{c}^{(2)} \rangle \\ &\ge 2 - \mathrm{TV}(\mathbf{U}) - 2\sqrt{\mathrm{TV}(\mathbf{U})}. \end{aligned} \tag{A.52}$$

When

$$\mathrm{TV}(\mathbf{U}) < \left(\sqrt{3} - 1 \right)^2, \tag{A.53}$$

we can let

$$\tau := \sqrt{2 - \mathrm{TV}(\mathbf{U}) - 2\sqrt{\mathrm{TV}(\mathbf{U})}}. \tag{A.54}$$

Then the clustering identification error of TSCC in the $\mathbf{T}$ space is bounded as follows:

$$e_{\mathrm{id}}(\mathbf{T}) \le \frac{1}{N} \cdot \sum_{k=1}^{2} \# \left\{ i \in \mathrm{I}_k \mid \left\| \mathbf{t}^{(i)} - \mathbf{c}_{\mathbf{T}}^{(k)} \right\|_2 \ge \tau/2 \right\}. \tag{A.55}$$

For each $k = 1, 2$, we apply Chebyshev's inequality and obtain that

$$\# \left\{ i \in \mathrm{I}_k \mid \left\| \mathbf{t}^{(i)} - \mathbf{c}_{\mathbf{T}}^{(k)} \right\|_2 \ge \tau/2 \right\} \le \frac{4}{\tau^2} \sum_{i \in \mathrm{I}_k} \left\| \mathbf{t}^{(i)} - \mathbf{c}_{\mathbf{T}}^{(k)} \right\|_2^2. \tag{A.56}$$

Thus,

$$\begin{aligned} e_{\mathrm{id}}(\mathbf{T}) &\le \frac{1}{N} \cdot \sum_{k=1,2} \frac{4}{\tau^2} \sum_{i \in \mathrm{I}_k} \left\| \mathbf{t}^{(i)} - \mathbf{c}_{\mathbf{T}}^{(k)} \right\|_2^2 \\ &\le \frac{4}{\tau^2} \sum_{k=1}^{2} \frac{N_k}{N} \sum_{i \in \mathrm{I}_k} \left\| \mathbf{u}^{(i)} - \mathbf{c}^{(k)} \right\|_2^2 \\ &\le \frac{4}{\tau^2} \cdot \mathrm{TV}(\mathbf{U}). \end{aligned} \tag{A.57}$$

In the $\mathbf{U}$ space, we also apply Lemma 3.2.1 with $K = 2$, together with the assumptions $N_2 \geq N_1 \geq \varepsilon_1 \cdot N/2$, and obtain that

$$\begin{aligned}
\left\|\mathbf{c}^{(1)} - \mathbf{c}^{(2)}\right\|_2^2 &= \left\|\mathbf{c}^{(1)}\right\|_2^2 + \left\|\mathbf{c}^{(2)}\right\|_2^2 - 2 \cdot \langle \mathbf{c}^{(1)}, \mathbf{c}^{(2)} \rangle \\
&\geq \frac{1}{N_2} \cdot \left(N_1 \ \left\|\mathbf{c}^{(1)}\right\|_2^2 + N_2 \ \left\|\mathbf{c}^{(2)}\right\|_2^2 \right) - \frac{2}{\sqrt{N_1 N_2}} \cdot \sqrt{N_1 N_2} \ \langle \mathbf{c}^{(1)}, \mathbf{c}^{(2)} \rangle \\
&\geq \frac{1}{N_2} \cdot (2 - \mathrm{TV}(\mathbf{U})) - \frac{2}{N_1} \sqrt{\mathrm{TV}(\mathbf{U})} \\
&\geq \frac{1}{N} \cdot \left(2 - \mathrm{TV}(\mathbf{U}) - 4/\varepsilon_1 \cdot \sqrt{\mathrm{TV}(\mathbf{U})} \right). \qquad \text{(A.58)}
\end{aligned}$$

When

$$\mathrm{TV}(\mathbf{U}) < \left(\sqrt{2 + \frac{4}{\varepsilon_1^2}} - \frac{2}{\varepsilon_1} \right)^2, \qquad \text{(A.59)}$$

we can apply similar steps as above to obtain that

$$e_{\mathrm{id}}(\mathbf{U}) \leq \frac{4 \, \mathrm{TV}(\mathbf{U})}{2 - \mathrm{TV}(\mathbf{U}) - 4/\varepsilon_1 \cdot \sqrt{\mathrm{TV}(\mathbf{U})}}. \qquad \text{(A.60)}$$

A.7 Proof of Theorem 3.3.4

The proof proceeds in parallel to that of Theorem 3.2.4. That is, we bound from below the K^{th} eigengap $\widetilde{\delta}_K$ of $\widetilde{\mathbf{W}}$, estimate from above the Frobenius norm of the perturbation $\mathbf{B} := \mathbf{W} - \widetilde{\mathbf{W}}$, and then conclude the theorem by combining these two bounds with Theorem A.5.1.

Straightforward calculation shows that the matrix $\widetilde{\mathbf{W}}$ (see formula in Equation (A.3)) has the following eigenvalues:

$$\widetilde{d}_K \geq \cdots \geq \widetilde{d}_2 \geq \widetilde{d}_1 \text{ and } \nu_K \geq \cdots \geq \nu_2 \geq \nu_1, \qquad \text{(A.61)}$$

where $\widetilde{d}_k, 1 \leq k \leq K$, are defined in equation (3.30), and

$$\nu_k := (d+1) \cdot \mathrm{P}(N_k - 2, d), \quad k = 1, \ldots, K. \qquad \text{(A.62)}$$

Using equation (3.4) we obtain that

$$N_K = N - \sum_{k=1}^{K-1} N_k \leq N - (K-1) \cdot \frac{\varepsilon_1 N}{K} = \left(1 - \frac{K-1}{K}\varepsilon_1\right) \cdot N. \tag{A.63}$$

The above equation together with equations (3.4) and (3.32) implies that

$$\begin{aligned}
\widetilde{\delta}_K &= \widetilde{d}_1 - \nu_K \\
&\geq \left(\frac{N_1}{2}\right)^{d+2} - \quad (d+1) \cdot {N_K}^d \\
&\geq \left(\frac{\varepsilon_1 N}{2K}\right)^{d+2} - (d+1) \cdot \left(1 - \frac{K-1}{K}\varepsilon_1\right)^d N^d \\
&\geq \frac{1}{2}\left(\frac{\varepsilon_1 N}{2K}\right)^{d+2}.
\end{aligned} \tag{A.64}$$

We follow by bounding the magnitude of the perturbation $\mathbf{B} = \widetilde{\mathbf{W}} - \mathbf{W}$:

$$\|\mathbf{B}\|_{\mathrm{F}} = \left\|\mathbf{A}\mathbf{E}' + \mathbf{E}\widetilde{\mathbf{A}}'\right\|_{\mathrm{F}} \leq \|\mathbf{A}\|_{\mathrm{F}} \|\mathbf{E}\|_{\mathrm{F}} + \|\mathbf{E}\|_{\mathrm{F}} \left\|\widetilde{\mathbf{A}}\right\|_{\mathrm{F}} \leq 2N^{(d+2)/2} \|\mathbf{E}\|_{\mathrm{F}}. \tag{A.65}$$

Therefore, by combining equations (A.64) and (A.65) with Theorem A.5.1 we conclude that when

$$N^{-(d+2)/2} \|\mathbf{E}\|_{\mathrm{F}} \leq \frac{1}{16}\left(\frac{\varepsilon_1}{2K}\right)^{d+2}, \tag{A.66}$$

we have

$$\left\|P^K(\mathbf{W}) - P^K(\widetilde{\mathbf{W}})\right\|_{\mathrm{F}} \leq 8\left(\frac{2K}{\varepsilon_1}\right)^{d+2} N^{-(d+2)/2} \|\mathbf{E}\|_{\mathrm{F}}. \tag{A.67}$$

Theorem 3.3.4 is then a direct consequence of combining the above equation and Lemma 3.2.3.

A.8 Proof of Lemma 4.4.1

For any $1 \leq k \leq K$ and $i \in \mathrm{I}_k$, we have

$$\begin{aligned}
D_{ii} &\geq \sum_{j \in \mathrm{I}_k} W_{ij} \geq \sum_{j \in \mathrm{I}_k} \sum_{i_2, \ldots, i_{d+2} \in \mathrm{I}_k} \mathcal{A}(i, i_2, \ldots, i_{d+2}) \mathcal{A}(j, i_2, \ldots, i_{d+2}) \\
&= \sum_{j \in \mathrm{I}_k} \sum_{\substack{i_2, \ldots, i_{d+2} \in \mathrm{I}_k \setminus \{i,j\} \\ \text{and are distinct}}} e^{-\frac{c_{\mathrm{p}}\left(\mathbf{x}_i, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_{d+2}}\right) + c_{\mathrm{p}}\left(\mathbf{x}_j, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_{d+2}}\right)}{\sigma}}.
\end{aligned} \tag{A.68}$$

When the given data is noiseless, the polar curvature of any distinct $d+2$ points in $\widetilde{\mathrm{C}}_k$ is zero. Hence,

$$D_{ii} \geq \sum_{j \in \mathrm{I}_k} \sum_{\substack{i_2,\ldots,i_{d+2} \in \mathrm{I}_k \setminus \{i,j\} \\ \text{and are distinct}}} 1 = \widetilde{d}_k, \tag{A.69}$$

where $\widetilde{d}_k$ (replicated N_k times), $1 \leq k \leq K$, are the diagonal elements of $\widetilde{\mathbf{D}}$ (see equation (3.30)). We have thus proved that $\mathbf{D} \geq \widetilde{\mathbf{D}}$.

A.9 Proof of Lemma 4.4.2

We take the expectation of each side of equation (A.68) with respect to the measure μ_{p} (defined in equation (4.1)), and proceed using Jensen's inequality (twice) as follows:

$$\begin{aligned} E_{\mu_{\mathrm{p}}}(D_{ii}) &\geq \sum_{j \in \mathrm{I}_k} \sum_{\substack{i_2,\ldots,i_{d+2} \in \mathrm{I}_k \setminus \{i,j\} \\ \text{and are distinct}}} e^{-\frac{2}{\sigma} \cdot E_{\mu_k^{d+2}} c_{\mathrm{p}}\left(\mathfrak{x}_i, \mathfrak{x}_{i_2}, \ldots, \mathfrak{x}_{i_{d+2}}\right)} \\ &\geq \sum_{j \in \mathrm{I}_k} \sum_{\substack{i_2,\ldots,i_{d+2} \in \mathrm{I}_k \setminus \{i,j\} \\ \text{and are distinct}}} e^{-\frac{2}{\sigma} \cdot \sqrt{E_{\mu_k^{d+2}} c_{\mathrm{p}}^2\left(\mathfrak{x}_i, \mathfrak{x}_{i_2}, \ldots, \mathfrak{x}_{i_{d+2}}\right)}} \\ &= e^{-\frac{2}{\sigma} \cdot c_{\mathrm{p}}(\mu_k)} \cdot \widetilde{d}_k, \end{aligned} \tag{A.70}$$

where in the last step we have used equation (1.4). Letting

$$\varepsilon_2 := \min_{1 \leq k \leq K} e^{-\frac{2}{\sigma} \cdot c_{\mathrm{p}}(\mu_k)} = e^{-\frac{2}{\sigma} \cdot \max_{1 \leq k \leq K} c_{\mathrm{p}}(\mu_k)}, \tag{A.71}$$

we have that

$$E_{\mu_{\mathrm{p}}}(D_{ii}) \geq \varepsilon_2 \cdot \widetilde{d}_k, \quad i \in \mathrm{I}_k, 1 \leq k \leq K. \tag{A.72}$$

Equivalently,

$$E_{\mu_{\mathrm{p}}}(\mathbf{D}) \geq \varepsilon_2 \cdot \widetilde{\mathbf{D}}. \tag{A.73}$$

A.10 Proof of Theorem 4.2.1

We first bound the expectation of the perturbation $\|\mathcal{E}_{\rm p}\|_{\rm F}^2$, where $\mathcal{E}_{\rm p} = \mathcal{A}_{\rm p} - \widetilde{\mathcal{A}}$, and then apply McDiarmid's inequality [47] to obtain a probabilistic estimate for $\|\mathcal{E}_{\rm p}\|_{\rm F}^2$. Finally, we conclude the proof by combining the probabilistic estimate together with Theorem 3.2.4.

Using the definitions of the sets $\mathrm{I}_1, \ldots, \mathrm{I}_K$ and the tensors $\mathcal{A}_{\rm p}$ and $\widetilde{\mathcal{A}}$, we express $\|\mathcal{E}_{\rm p}\|_{\rm F}^2$ as a function of the random variables $\mathfrak{X}_1, \ldots, \mathfrak{X}_N$:

$$\|\mathcal{E}_{\rm p}\|_{\rm F}^2 = \sum_{k=1}^{K} \sum_{\mathrm{I}_k^{d+2}} \left(1 - e^{\frac{-c_{\rm p}(\mathfrak{X}_{i_1}, \ldots, \mathfrak{X}_{i_{d+2}})}{\sigma}}\right)^2 + \sum_{\left(\bigcup_{k=1}^{K} \mathrm{I}_k^{d+2}\right)^c} \left(e^{\frac{-c_{\rm p}(\mathfrak{X}_{i_1}, \ldots, \mathfrak{X}_{i_{d+2}})}{\sigma}}\right)^2. \quad \text{(A.74)}$$

By applying the inequality: $1 - e^{-|x|} \leq |x|$, we obtain that

$$\|\mathcal{E}_{\rm p}\|_{\rm F}^2 \leq \sum_{k=1}^{K} \sum_{\mathrm{I}_k^{d+2}} \frac{c_{\rm p}^2(\mathfrak{X}_{i_1}, \ldots, \mathfrak{X}_{i_{d+2}})}{\sigma^2} + \sum_{\left(\bigcup_{k=1}^{K} \mathrm{I}_k^{d+2}\right)^c} e^{\frac{-c_{\rm p}(\mathfrak{X}_{i_1}, \ldots, \mathfrak{X}_{i_{d+2}})}{\sigma/2}}. \quad \text{(A.75)}$$

We then take the expectation of $\|\mathcal{E}_{\rm p}\|_{\rm F}^2$ (with respect to $\mu_{\rm p}$) using equations (1.4) and (4.3) and have that

$$\begin{aligned} E_{\mu_{\rm p}}(\|\mathcal{E}_{\rm p}\|_{\rm F}^2) &\leq \frac{1}{\sigma^2} \sum_{k=1}^{K} N_k^{d+2} c_{\rm p}^2(\mu_k) + N^{d+2} C_{\rm in}(\mu_1, \ldots, \mu_K; \sigma/2) \\ &= N^{d+2} \cdot \left(\frac{1}{\sigma^2} \sum_{k=1}^{K} \left(\frac{N_k}{N}\right)^{d+2} c_{\rm p}^2(\mu_k) + C_{\rm in}(\mu_1, \ldots, \mu_K; \sigma/2) \right) \\ &\leq \alpha \cdot N^{d+2}, \end{aligned} \quad \text{(A.76)}$$

in which

$$\alpha := \frac{1}{\sigma^2} \cdot \sum_{k=1}^{K} c_{\rm p}^2(\mu_k) + C_{\rm in}(\mu_1, \ldots, \mu_K; \sigma/2). \quad \text{(A.77)}$$

We next note that for each fixed $1 \leq i \leq N$,

$$\sup_{\mathfrak{X}_1, \ldots, \mathfrak{X}_N, \widehat{\mathfrak{X}}_i} \left| \|\mathcal{E}_{\rm p}\|_{\rm F}^2 (\mathfrak{X}_1, \ldots, \mathfrak{X}_i, \ldots, \mathfrak{X}_N) - \|\mathcal{E}_{\rm p}\|_{\rm F}^2 (\mathfrak{X}_1, \ldots, \widehat{\mathfrak{X}}_i, \ldots, \mathfrak{X}_N) \right| \leq (d+2) \cdot N^{d+1}. \quad \text{(A.78)}$$

Indeed, the number of additive terms in $\|\mathcal{E}_{\mathrm{p}}\|_{\mathrm{F}}^2(\mathfrak{X}_1,\ldots,\mathfrak{X}_N)$ that contain $\mathfrak{X}_i$ is $(d+2)\cdot \mathrm{P}(N-1,d+1)$, and each of them is between 0 and 1.

The above property implies that $\|\mathcal{E}_{\mathrm{p}}\|_{\mathrm{F}}^2$ satisfies McDiarmid's inequality [47], that is,

$$\mu_{\mathrm{p}}\left(\|\mathcal{E}_{\mathrm{p}}\|_{\mathrm{F}}^2 - E_{\mu_{\mathrm{p}}}(\|\mathcal{E}_{\mathrm{p}}\|_{\mathrm{F}}^2) \geq \alpha N^{d+2}\right) \leq e^{-2N\alpha^2/(d+2)^2}. \tag{A.79}$$

Combining the last equation with equation (A.76) yields that

$$\mu_{\mathrm{p}}\left(\|\mathcal{E}\|_{\mathrm{F}}^2 \geq 2\alpha N^{d+2}\right) \leq e^{-2N\alpha^2/(d+2)^2}, \tag{A.80}$$

or equivalently,

$$\mu_{\mathrm{p}}\left(N^{-(d+2)}\|\mathcal{E}_{\mathrm{p}}\|_{\mathrm{F}}^2 < 2\alpha\right) \geq 1 - e^{-2N\alpha^2/(d+2)^2}. \tag{A.81}$$

Consequently, combining Theorem 3.2.4 and the last equation gives that, if

$$2\alpha \leq \frac{1}{8C_1}, \tag{A.82}$$

where $C_1 = C_1(K,d,\varepsilon_1,\varepsilon_2)$ is defined in equation (A.49), then

$$\begin{aligned}
&\mu_{\mathrm{p}}\left(\mathrm{TV}(\mathbf{U}) < 2\alpha\cdot C_1 \mid \text{Assumption 1 holds}\right)\\
&\geq \mu_{\mathrm{p}}\left(\mathrm{TV}(\mathbf{U}) < 2\alpha\cdot C_1 \mid \text{Assumption 1 holds, and } N^{-(d+2)}\|\mathcal{E}_{\mathrm{p}}\|_{\mathrm{F}}^2 < 2\alpha\right)\\
&\qquad\cdot \mu_{\mathrm{p}}\left(N^{-(d+2)}\|\mathcal{E}_{\mathrm{p}}\|_{\mathrm{F}}^2 < 2\alpha \mid \text{Assumption 1 holds}\right)\\
&= 1\cdot \mu_{\mathrm{p}}\left(N^{-(d+2)}\|\mathcal{E}_{\mathrm{p}}\|_{\mathrm{F}}^2 < 2\alpha\right)\\
&\geq 1 - e^{-2N\alpha^2/(d+2)^2}.
\end{aligned} \tag{A.83}$$

A.11 Proof of Equation (4.18)

For any three points $\mathbf{p}_1(x_1, 0), \mathbf{p}_2(x_2, 0) \in \mathrm{L1}$, and $\mathbf{q}(0, y) \in \mathrm{L2}$, their polar curvature is bounded below by

$$\begin{aligned} c_{\mathrm{p}}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{q}) &= \operatorname{diam}\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{q}\} \cdot \sqrt{\sin^2 \angle \mathbf{p}_1\mathbf{p}_2\mathbf{q} + \sin^2 \angle \mathbf{p}_2\mathbf{p}_1\mathbf{q} + \sin^2 \angle \mathbf{p}_1\mathbf{q}\mathbf{p}_2} \\ &\geq \max\left(\sqrt{x_1^2 + y^2}, \sqrt{x_2^2 + y^2}\right) \cdot \sqrt{\frac{y^2}{x_1^2 + y^2} + \frac{y^2}{x_2^2 + y^2}} \\ &\geq \sqrt{y^2 + y^2} = \sqrt{2} \cdot y. \end{aligned} \tag{A.84}$$

Thus, by using the symmetry of the lines, we obtain that

$$\begin{aligned} C_{\mathrm{in}}(\mu_1, \mu_2; \sigma) &= \int_{\mathrm{L1}} \int_{\mathrm{L1}} \int_{\mathrm{L2}} e^{-\frac{c_{\mathrm{p}}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{q})}{\sigma}} \, \mathrm{d}\mu_1(\mathbf{p}_1) \, \mathrm{d}\mu_1(\mathbf{p}_2) \, \mathrm{d}\mu_2(\mathbf{q}) \\ &\leq \int_0^L e^{-\frac{\sqrt{2}\, y}{\sigma}} \, \frac{\mathrm{d}y}{L} = \frac{\sigma}{\sqrt{2}L} \left(1 - e^{-\sqrt{2}L/\sigma}\right). \end{aligned} \tag{A.85}$$

A.12 Proof of Equation (4.19)

For any two points $\mathbf{p}(x, 0) \in \mathrm{L1}, \mathbf{q}(r\cos\theta, r\sin\theta) \in \mathrm{L2}$, the polar curvature of $\mathbf{p}, \mathbf{q}$ and the origin $\mathbf{o}$ is bounded below by

$$\begin{aligned} c_{\mathrm{p}}(\mathbf{o}, \mathbf{p}, \mathbf{q}) &= \operatorname{diam}\{\mathbf{o}, \mathbf{p}, \mathbf{q}\} \cdot \sqrt{\sin^2\theta + \sin^2 \angle \mathbf{o}\mathbf{p}\mathbf{q} + \sin^2 \angle \mathbf{o}\mathbf{q}\mathbf{p}} \\ &\geq \max(x, r) \cdot \sin\theta. \end{aligned} \tag{A.86}$$

Thus, the incidence constant is bounded above by

$$\begin{aligned} C_{\text{in,L}}(\mu_1,\mu_2;\sigma) &= \int_{\text{L1}}\int_{\text{L2}} e^{-\frac{c_{\text{p}}(\mathbf{o},\mathbf{p},\mathbf{q})}{\sigma}}\,\mathrm{d}\mu_1(\mathbf{p})\,\mathrm{d}\mu_2(\mathbf{q}) \\ &\le \int_0^L\int_0^L e^{-\frac{\max(x,r)\cdot\sin\theta}{\sigma}}\,\frac{\mathrm{d}x}{L}\frac{\mathrm{d}r}{L} \\ &= 2\iint_{0\le x\le r\le L} e^{-\frac{r\sin\theta}{\sigma}}\,\frac{\mathrm{d}x}{L}\frac{\mathrm{d}r}{L} \\ &= \frac{2}{L}\int_0^L r\cdot e^{-\frac{r\sin\theta}{\sigma}}\,\frac{\mathrm{d}r}{L} \\ &= 2\left(\frac{\sigma}{L\sin\theta}\right)^2\cdot\left(1-e^{-\frac{L\sin\theta}{\sigma}}\left(1+\frac{L\sin\theta}{\sigma}\right)\right). \end{aligned} \tag{A.87}$$

A.13 Proof of Equation (4.20)

For any $\mathbf{p}(x,y_2)\in\text{R1},\mathbf{q}(x_1,y)\in\text{R2}$, we define $\widetilde{\mathbf{p}}(x,\epsilon)\in\text{R1},\widetilde{\mathbf{q}}(\epsilon,y)\in\text{R2}$. The polar curvature of $\mathbf{p},\mathbf{q}$ and the origin $\mathbf{o}$ is bounded below by

$$\begin{aligned} c_{\text{p}}(\mathbf{o},\mathbf{p},\mathbf{q}) &\ge \max(\|\mathbf{op}\|,\|\mathbf{oq}\|)\cdot\sin\angle\mathbf{poq} \ge \max(x,y)\cdot\sin\angle\widetilde{\mathbf{p}}\mathbf{o}\widetilde{\mathbf{q}} \\ &= \frac{\max(x,y)\cdot(xy-\epsilon^2)}{\sqrt{(x^2+\epsilon^2)(y^2+\epsilon^2)}}. \end{aligned} \tag{A.88}$$

Thus, the incidence constant is

$$\begin{aligned} C_{\text{in,L}}(\mu_1,\mu_2;\sigma) &= \int_{\text{R1}}\int_{\text{R2}} e^{-\frac{c_{\text{p}}(\mathbf{o},\mathbf{p},\mathbf{q})}{\sigma}}\,\frac{\mathrm{d}x}{L}\frac{\mathrm{d}y_2}{\epsilon}\frac{\mathrm{d}x_1}{\epsilon}\frac{\mathrm{d}y}{L} \\ &\le \frac{1}{L^2}\cdot\int_\epsilon^{L+\epsilon}\int_\epsilon^{L+\epsilon} e^{-\frac{\max(x,y)\cdot(xy-\epsilon^2)}{\sigma\cdot\sqrt{(x^2+\epsilon^2)(y^2+\epsilon^2)}}}\,\mathrm{d}x\,\mathrm{d}y. \end{aligned} \tag{A.89}$$

Changing variables $x:=x/\epsilon, y:=y/\epsilon$ and setting $\omega:=L/\epsilon$ gives that

$$C_{\text{in}}(\mu_1,\mu_2;\sigma) \le \frac{1}{\omega^2}\cdot\int_1^{1+\omega}\int_1^{1+\omega} e^{-\frac{\max(x,y)\cdot(xy-1)}{\sigma\cdot\sqrt{(x^2+1)(y^2+1)}}}\,\mathrm{d}x\,\mathrm{d}y. \tag{A.90}$$

We observe that the integrand is bounded between 0 and 1, symmetric about x and

y, and decreasing in each of its arguments. We thus obtain that

$$\begin{aligned} C_{\mathrm{in,L}}(\mu_1,\mu_2;\sigma) &\le \frac{1}{\omega^2}\cdot\left(\int_1^{1+\sqrt[4]{\sigma}}\int_1^{1+\sqrt[4]{\sigma}} +2\int_1^{1+\sqrt[4]{\sigma}}\int_{1+\sqrt[4]{\sigma}}^{1+\omega} +\int_{1+\sqrt[4]{\sigma}}^{1+\omega}\int_{1+\sqrt[4]{\sigma}}^{1+\omega}\right) \\ &\qquad e^{-\frac{\max(x,y)\cdot(xy-1)}{\sigma\cdot\sqrt{(x^2+1)(y^2+1)}}}\,\mathrm{d}x\,\mathrm{d}y \\ &\le \frac{1}{\omega^2}\cdot\left(\left(\sqrt[4]{\sigma}\right)^2+2\cdot\sqrt[4]{\sigma}\cdot(\omega-\sqrt[4]{\sigma})\cdot e^{-\frac{\left(1+\sqrt[4]{\sigma}\right)\cdot\left(1\cdot\left(1+\sqrt[4]{\sigma}\right)-1\right)}{\sigma\cdot\sqrt{2\cdot\left(1+\left(1+\sqrt[4]{\sigma}\right)^2\right)}}}\right) \\ &\qquad +\frac{1}{\omega^2}\cdot\left(\omega-\sqrt[4]{\sigma}\right)^2\cdot e^{-\frac{\left(1+\sqrt[4]{\sigma}\right)\cdot\left(\left(1+\sqrt[4]{\sigma}\right)^2-1\right)}{\sigma\cdot\left(1+\left(1+\sqrt[4]{\sigma}\right)^2\right)}} \\ &\le \frac{\sqrt{\sigma}}{\omega^2}+\frac{2\sqrt[4]{\sigma}}{\omega}\cdot e^{-1/\left(2\sigma^{3/4}\right)}+e^{-1/\sigma^{3/4}}. \end{aligned} \tag{A.91}$$

A.14 Proof of Equation (4.21)

Let $\mathbf{p}(0,\rho\cos\varphi,\rho\sin\varphi)\in$ D1, and $\mathbf{q}_1(0,r_1\cos\theta_1,r_1\sin\theta_1),\mathbf{q}_2(0,r_2\cos\theta_2,r_2\sin\theta_2)\in$ D2. Then the polar curvature of these three points and the origin $\mathbf{o}$ has the following lower bound:

$$c_{\mathrm{p}}(\mathbf{o},\mathbf{p},\mathbf{q}_1,\mathbf{q}_2)\ge|\mathbf{op}|\cdot\mathrm{psin}_{\mathbf{o}}(\mathbf{p},\mathbf{q}_1,\mathbf{q}_2)=\rho\cdot\sin\varphi\,\sin|\theta_1-\theta_2|. \tag{A.92}$$

Due to the symmetry of the two disks, we have that

$$\begin{aligned} C_{\mathrm{in,L}}(\mu_1,\mu_2;\sigma) &= \int_{\mathrm{D1}}\int_{\mathrm{D2}}\int_{\mathrm{D2}} e^{-c_{\mathrm{p}}(\mathbf{o},\mathbf{p},\mathbf{q}_1,\mathbf{q}_2)/\sigma}\,\mathrm{d}\mu_1(\mathbf{p})\,\mathrm{d}\mu_2(\mathbf{q}_1)\,\mathrm{d}\mu_2(\mathbf{q}_2) \\ &\le \int_0^1\int_0^{\pi}\int_{-\pi/2}^{\pi/2}\int_{-\pi/2}^{\pi/2} e^{-\frac{\rho\sin\varphi\cdot\sin|\theta_1-\theta_2|}{\sigma}}\,\frac{\rho\,\mathrm{d}\rho\,\mathrm{d}\varphi}{\pi/2}\,\frac{\mathrm{d}\theta_1}{\pi}\,\frac{\mathrm{d}\theta_2}{\pi} \\ &= \frac{4}{\pi^3}\cdot\int_0^1\int_0^{\pi}\iint_{-\frac{\pi}{2}\le\theta_2\le\theta_1\le\frac{\pi}{2}} e^{\frac{-\rho\sin\varphi\cdot\sin(\theta_1-\theta_2)}{\sigma}}\,\rho\,\mathrm{d}\rho\,\mathrm{d}\varphi\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2. \end{aligned} \tag{A.93}$$

Changing variables $\theta:=\theta_1-\theta_2,\theta_2:=\theta_2$ and exchanging the corresponding double

integral, we obtain that

$$\begin{aligned} C_{\mathrm{in,L}}(\mu_1,\mu_2;\sigma) &\le \frac{4}{\pi^3}\cdot\int_0^1\int_0^\pi\int_0^\pi e^{-\frac{\rho\sin\varphi\cdot\sin\theta}{\sigma}}\,\rho\,\mathrm{d}\rho\,\mathrm{d}\varphi\,(\pi-\theta)\,\mathrm{d}\theta \\ &\le \frac{4}{\pi^2}\cdot\int_0^1\int_0^\pi\int_0^\pi e^{-\frac{\rho\sin\varphi\cdot\sin\theta}{\sigma}}\,\rho\,\mathrm{d}\rho\,\mathrm{d}\varphi\,\mathrm{d}\theta \\ &= \frac{16}{\pi^2}\cdot\int_0^1\int_0^{\pi/2}\int_0^{\pi/2} e^{-\frac{\rho\sin\varphi\cdot\sin\theta}{\sigma}}\,\rho\,\mathrm{d}\rho\,\mathrm{d}\varphi\,\mathrm{d}\theta. \end{aligned} \tag{A.94}$$

We observe that the integrand is bounded between 0 and 1, symmetric about φ and θ, and decreasing in each of them. Thus,

$$\begin{aligned} C_{\mathrm{in,L}}(\mu_1,\mu_2;\sigma) &\le \frac{16}{\pi^2}\cdot\int_0^1\left(\int_0^{\sqrt[4]{\sigma}}\int_0^{\sqrt[4]{\sigma}} +2\int_0^{\sqrt[4]{\sigma}}\int_{\sqrt[4]{\sigma}}^{\frac{\pi}{2}} +\int_{\sqrt[4]{\sigma}}^{\frac{\pi}{2}}\int_{\sqrt[4]{\sigma}}^{\frac{\pi}{2}}\right) \\ &\qquad e^{-\frac{\rho\sin\varphi\cdot\sin\theta}{\sigma}}\,\rho\,\mathrm{d}\rho\,\mathrm{d}\varphi\,\mathrm{d}\theta \\ &\le \frac{16}{\pi^2}\cdot\left(\left(\sqrt[4]{\sigma}\right)^2+2\cdot\sqrt[4]{\sigma}\cdot\left(\frac{\pi}{2}-\sqrt[4]{\sigma}\right)\right)\cdot\int_0^1\rho\,\mathrm{d}\rho \\ &\qquad +\frac{16}{\pi^2}\cdot\left(\frac{\pi}{2}-\sqrt[4]{\sigma}\right)^2\cdot\int_0^1 e^{-\frac{\rho\cdot\left(\sin\sqrt[4]{\sigma}\right)^2}{\sigma}}\,\rho\,\mathrm{d}\rho \\ &\le \frac{8\sqrt{\sigma}}{\pi^2}+\frac{8\sqrt[4]{\sigma}}{\pi}+\frac{4\sigma^2}{(\sin\sqrt[4]{\sigma})^4}. \end{aligned} \tag{A.95}$$