STATISTICAL LEARNING: GENERATIVE MODEL FOR DIMENSIONALITY REDUCTION, CLUSTERING, AND SHAPE ANALYSIS

by

Yen-Yun Yu

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computing

School of Computing

The University of Utah

August 2016

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Yen-Yun Yu**

has been approved by the following supervisory committee members:

Ross T Whitaker , Chair **03/30/2016**
Date Approved

Suyash Prakash Awate , Co-chair **04/05/2016**
Date Approved

Preston Thomas Fletcher , Member **03/25/2016**
Date Approved

Jeffrey Phillips , Member **03/25/2016**
Date Approved

Craig Carl Teerlink , Member **03/25/2016**
Date Approved

and by **Ross T Whitaker** , Chair/Dean of

the Department/College/School of **Computing**

and by David B. Kieda, Dean of The Graduate School.

ABSTRACT

Statistical learning theory has garnered attention during the last decade because it provides the theoretical and mathematical framework for solving pattern recognition problems, such as dimensionality reduction, clustering, and shape analysis. In statistical learning, a generative model is a fully probabilistic model of observed data and latent variables with important properties, including compact representations, parameter estimation and subsequent statistical analysis, and the ability to induce classifications on unseen data. This dissertation proposes new generative approaches for 1) learning in kernel feature space, 2) learning via the semisupervised approach, and 3) learning from shape data.

This dissertation first proposes a new statistical tool, *kernel principal geodesic analysis*, for hyperspherical statistical analysis in kernel feature space, including algorithms for computing the sample-weighted Karcher mean and eigenanalysis of the sample-weighted Karcher covariance. It then applies these tools to advance novel methods for dimensionality reduction and clustering. Second, this dissertation proposes a novel generative framework for multigroup shape analysis relying on a *hierarchical generative shape model* on shapes within a population. The framework represents individual shapes as point sets modulo translation, rotation, and scale, following the notion in Kendall shape space. Whereas individual shapes are derived from their group shape model, each group shape model is derived from a single population shape model. The hierarchical model follows the natural organization of population data, and the top level in the hierarchy provides a common frame of reference for multigroup shape analysis. Lastly, this dissertation proposes a principled generative approach, *generative model with pairwise relationships*, to probabilistically model the joint distribution given by user-defined pairwise relations between data points (e.g., pairs of data points belonging to the same class or different classes) to remarkably improve the performance of unsupervised clustering. The proposed model accounts for general underlying distributions without assuming a specific form.

CONTENTS

Chapters

LIST OF FIGURES

ACKNOWLEDGMENTS

My deepest gratitude is to my advisor, Dr. Ross Whitaker, and coadvisor Dr. Suyash Awate. They were always there to give advice. I would like to acknowledge Dr. Thomas Fletcher and Dr. Craig Teerlink for their friendship and assistance during my program at the University of Utah. I would like to thank Dr. Jeff Phillips for his input. I would also like to thank Karen, Ann, Christine, Danny, Shireen, and all my friends in the School of Computing and SCI. I appreciate the financial support from Exxon Mobile. Finally, and most importantly, I would like to thank my parents, parents-in-law, my dear wife Yu-Yin, and my dear son Luke.

CHAPTER 1

INTRODUCTION

The focus of this dissertation is the automatic recognition of patterns from data. Many types of algorithms have been proposed to understand the regularities in data, including nonparametric methods and neural networks. In this dissertation, I primarily focus on statistical learning theory and its subcategory, the **generative model** learning algorithms. Statistical learning-based methods perform consistently over time and provide a useful benchmark for measuring efficiency. Furthermore, statistical methods provide a theoretical approach to data analysis, where data/information is presented and organized in an analytical manner. The works in this dissertation use generative models for solving the pattern recognition problems: dimensionality reduction, clustering, and shape analysis.

1.1 Statistical Learning Theory Using Generative Models

Statistical learning theory has garnered attention in several research areas, such as computer vision [1], [2], biology [3], [4], financial analysis [5], and petroleum exploration [6]. It contributes the *probability theory* and the *statistical* aspects of pattern recognition, which aim to discover the structure of data while classifing the data into different categories. Probability theory and statistics are useful as a more theoretical approach to forecasting (i.e., prediction). They further facilitate the evaluation of work performance (e.g., null hypothesis testing). Statistical learning generally involves 1) learning patterns from data, 2) building the predictive function or statistical model based on the patterns, and 3) predicting unknown data via the model we have learned.

In statistical learning, a *generative model* is a fully probabilistic model of all variables. It is different from a *discriminative model*, which evaluates only the target value y (i.e., all possible outputs) conditional upon the observed variable x. Because discriminative models directly optimize the conditional prediction $P(y|x)$, they often outperform generative models in classification tasks. However, numerous studies have demonstrated that discriminative models that are not carefully regularized may be outperformed by simple generative models (e.g., naive Bayes) [7]–[11]. Further, if the size of the training data is *not* large enough, a good performance of the training data will *not* guarantee the recurrence of future testing data, i.e., learned discriminative models easily overfit the training data. Lastly, most discriminative models are supervised, which means that without labeled data, it is impossible to train the model.

Generative models define the joint probability density function $P(x, y)$ over both the observed variable x and *latent variable* y, which is not observed directly but can be inferred from the observed variable. By assuming a parametric form for the underlying distribution that generates the data in each class, a generative model has some advantages over a discriminative model. First, a generative model not only can *more precisely* describe the relation between the observed variable and the target value (i.e., the latent variable) but also sample *any* other variables in the model. Second, a generative model can often perform better than a discriminative model on a *small* set of training data, because the model trains the joint probability density function $P(x, y)$, in which the marginal step (i.e., marginal probability $P(x) = \sum_y P(x, y)$) is treated essentially as a regularizer and hence can avoid the overfitting problem [12]. Third, labeled data may be expensive or difficult to obtain in some research areas (e.g., medical imaging analysis). Unlike a discriminative model, a generative model can be used for unsupervised learning, i.e., labeled data are not required. Fourth, a generative model is an induction that learns a model based on data from the problem space. The model is further used to test new data, which can be any data point in that space (i.e., reasoning from training data to a *general* rule). Another advantage of a generative model is that it includes latent variables that allow more complex distributions to be formed from a simplistic model.

This dissertation proposes *generative models* for a more efficient and principled way to drive the *latent variable* during learning, which focuses on the applications in *clustering* and *statistical shape analysis*. This dissertation also proposes a statistical analysis tool for feature extraction: *dimensionality reduction*, which is a standard preprocessing step that extracts features of interest from the data and can be computationally efficient.

1.2 Motivation

This dissertation focuses on the problem of processing data and identifying data that belong to a set of categories, classified according to learned regularities in the data. Therefore, this dissertation deals with finding a predictive function (or model) and assigning a label to the data. For example, if I want to build a computer-aided diagnosis (CAD) tool to evaluate a patient suspected of having a tumor via an x-ray image, I first need a collection of x-ray images (i.e., training data) that have been labeled $+1$ (i.e., healthy) or -1 (i.e., has a tumor) by a medical doctor. The set of training data is used to build a statistical model that performs a predictive function through a *learning* phase. Once the system has been trained, it then is able to take any new patient's x-ray image as input x (i.e., testing data) and produce the target value $f(x) = y$, where $y = \{+1, -1\}$ through a *generalization* phase. In addition, a successful inference is based on a statistical *assumption.* The assumption from the x-ray images could be that both training data and testing data are drawn independently from an identical distribution (e.g., an identical population). In other words, all the x-ray images are collected from patients who show a common set of relevant characteristics (e.g., demographics).

We can consider again the above medical imaging analysis example to examine the problem in more detail. In order to learn the complex medical images more efficiently, we can first extract a distinct feature to represent the interesting parts of an image instead of using the entire image (a.k.a., dimensionality reduction). Afterwards, we can group the images into the different clusters that are used to recognize the similarity between the images within the same/different cluster (a.k.a., clustering). Last, because in this case the data are an image, the geometrical properties of the shape of the tumor can be used as landmarks for the analysis (a.k.a., statistical shape analysis). In the following subsections, I further discuss the motivation for 1) *dimensionality reduction*, 2) *clustering*, and 3) *statistical shape analysis.*

1.2.1 Dimensionality Reduction

Dimensionality reduction aims to transform the data from a high-dimensional space to a lower dimensional space, in which data have a less noisy and more interpretable representation; hence, learning problems can be solved more efficiently. Fig. 1.1 provides an example of why dimensional reduction is important in statistical learning. The grayscale face image always lies in the space in which the dimension is equal to a large number of pixels; however, the underlying structure of the face image could be related to only a small number of parameters, such as different camera angles, pose angles, lighting conditions, and facial expressions. Therefore, the dimensional reduction step captures the invariant data via fewer variables, and it makes the next step, image analysis, more efficient and tractable. In addition to the challenge of space complexity, time complexity can also be a problem in the analysis of data in high-dimensional space because of the greater cost of computation. Therefore, dimensionality reduction allows the inference to have a lower computation cost with fewer variables. Dimensional reduction is also helpful for utility

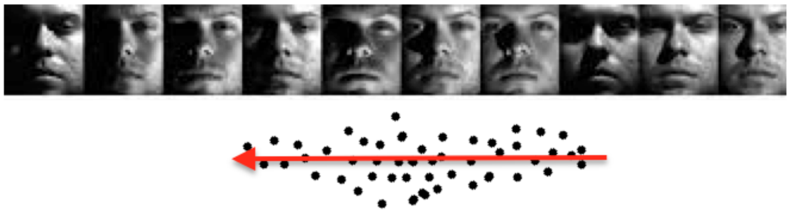


Fig. 1.1. **Illustration of dimensionality reduction:** Each point indicates a face image. The red line represents the lower-dimensional space that captures the different lighting conditions (YALE face image dataset [13]).

performance in statistical learning since it avoids the overfitting problem.

Principal component analysis (PCA) is one of the most popular dimensionality reduction algorithms. PCA is based on finding the principal components that are drawn on the sequentially orthogonal components with the largest variances, which are particularly useful in statistical learning. In 1998, an extension of linear principal component analysis was proposed: kernel principal component analysis (KPCA). KPCA is a nonlinear dimensionality reduction technique. The theory behind KPCA is to perform PCA in kernel feature space using the kernel method. Conventionally, KPCA relies on Euclidean statistics in kernel feature space. However, Euclidean analysis can make KPCA inefficient or incorrect for many popular kernels that map input points to a *hypersphere* in the kernel feature space. Thus, this dissertation proposes a novel adaptation of KPCA, namely *kernel principal geodesic analysis*, for *hyperspherical* statistical analysis in kernel feature space. The proposed kernel principal geodesic analysis is therefore able to describe the data more meaningfully and be used as a statistical tool for applications such as *dimensionality reduction* and *clustering*.

1.2.2 Clustering

In statistical learning, the two major subclasses are *supervised* learning (i.e., classification) and *unsupervised* learning (i.e., clustering). The goal of supervised classification is to learn directly from a set of training data (i.e., labeled data) and assign the testing data to a predefined class. Unsupervised clustering aims to group a set of data into different clusters based on the similarities of the data. Unlike the supervised classification problem, the clustering model is derived from data without labeled information. Thus, clustering groups the data instead of labeling the data. The unsupervised clustering algorithm can, therefore, serve as a tool to provide meaningful insights into the structure of the data. To explain the importance of clustering, I list some applications:

- **Image segmentation** [1], [2]: Image segmentation aims to divide an image into multiple segments that are more useful for further analysis, a very important preprocess for medical imaging analysis, image retrieval, or object tracking. The framework for clustering in image segmentation is to partition pixels into several clusters.

- **Biology** [3], [4]: Clustering is an important tool in bioinformatics. One of the applications of clustering in bioinformatics is to group homologous sequences into different clusters. This clustering is then used to understand the function or evolution of DNA or RNA.

- **Business** [5]: Business modeling uses clustering analysis to recognize, for example, groups of people with similar behaviors in order to conduct market segmentation or develop new products.

- **Sports analysis** [14], [15]: The goal of sports analysis is to evaluate the performance of the professional team or individual player. The sports analyst employs clustering as a tool to segment players into different clusters, which is useful information for negotiating a contract or designing tactics.

Clustering can also be categorized as a generative or a discriminative approach. Because a generative assumption is made to model the data and their underlying distribution, the goal of a generative approach is to find the optimal parameters to maximize the probability (i.e., likelihood) of the data given the model. Moreover, a generative clustering model can be viewed as distribution-based clustering, which provides probability information about the data. In contrast, in a discriminative clustering model (e.g., graph-based clustering), the algorithms try to learn from relationships (e.g., pairwise) over the set of input data, so they typically do not rely on the underlying parametric form of the distribution. In this dissertation, I focus on clustering using a generative model.

Recently, many studies have indicated that the performance of statistical learning significantly improves by incorporating *large* amounts of unlabeled data with a *small* amount of labeled data., e.g., semisupervised learning [16]–[27]. An example of such classification is a toddler using flash cards (i.e., unlabeled data) to learn words (see Fig. 1.2). The toddler might learn more efficiently if the parent gives the toddler a few examples (i.e., labeled data). However, the semisupervised approach using a generative model can be considered as a method to find the maximum likelihood, which combines the likelihood of the unsupervised and supervised approaches. Therefore, the performance of the semisupervised approach simply depends on the ratio of labeled data to unlabeled data [28]–[31]. To address this

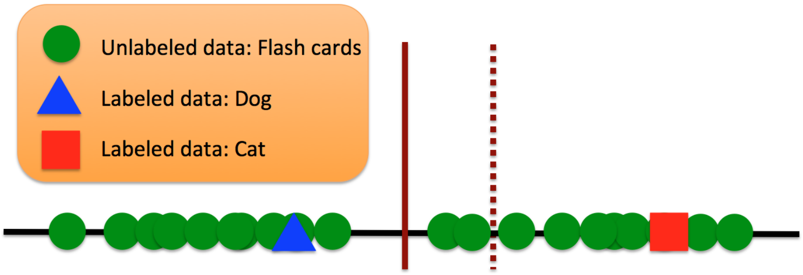


Fig. 1.2. **Example of semisupervised learning**: Learning with a large amount of unlabeled data and a small amount of labeled data. The dotted brown line is the decision boundary when only labeled data are available. However, the true decision boundary (i.e., solid brown line) will move to the left side of the dotted line when a large amount of unlabeled data is available simultaneously. In this case, two cards were incorrectly predicted to be cats when using only the labeled data.

issue, this dissertation proposes a generative approach by incorporating an instance-level pairwise relationship with the clustering problem in a precise manner.

1.2.3 Shape Analysis

Shape modeling and analysis is an important problem in a variety of fields, including biology, medical image analysis, and computer vision [32]–[34], and has received considerable attention over the last few decades. Objects in biological images or anatomical structures in medical images often possess *shape* as the sole identifying characteristic instead of color or texture. Applications of shape analysis beyond natural contexts include handwriting analysis and character recognition. In the medical context, shapes of anatomical structures can provide crucial cues in the diagnosis of pathologies or disorders. The key problems in this context lie in the fitting of shape models to population image data, followed by statistical analysis such as hypothesis testing to compare groups, classification of unseen shapes in one of the groups for which the shape model is learned, etc.

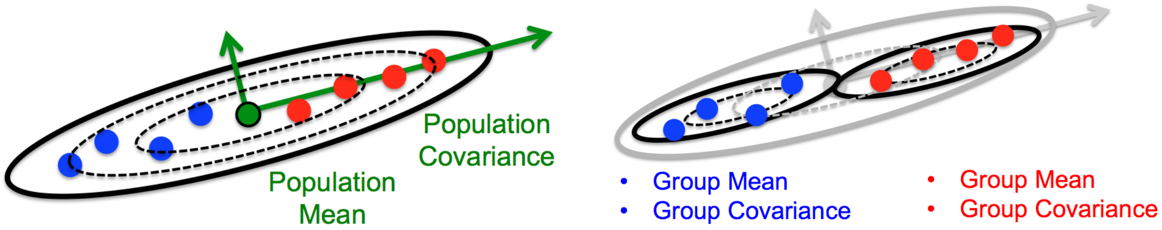
In section 1.1, I described how a generative model could perform well if the model assumption is correct, even if the size of the data is small. This is an especially attractive motivation to use a generative model for statistical shape analysis in medical imaging analysis where medical image data are not easily obtained for research due to governmental policies and/or personal privacy concerns. In order to build a reliable predictive model (e.g., computer-aided diagnosis) following the statistical approach, a generative model is clearly a better choice than a discriminative model. Further, the nature of a generative model is that any sample can be drawn when the model is given. This property is also extremely useful for medical imaging analysis. For example, one can predict and visualize the shape variation of a brain along with time from a patient who has Alzheimer's disease. For applications in image segmentation, a generative model provides prior information of shape variation; this information can significantly improve the performance of image segmentation. For applications in anatomical atlases, if the particular statistical shape analysis algorithm is based on a generative model, the joint distribution allows us to present a full range of geometric variabilities. In general, statistical shape analysis [32], [33] entails the inference of shape models from population data and associated statistical analyses. However, previous approaches [35]–[39] considered *only* the population mean and covariance for multigroup comparisons, such as hypothesis testing, as in Fig. 1.3-(a). This dissertation, on the other hand, proposes a generative latent variable model that entails a multigroup strategy (Fig. 1.3-(b)), a more natural way to solve statistical shape analysis.

1.3 Dissertation and Claims

In this section, I summarize the contributions of this dissertation.

- A novel method that extends KPCA, named *kernel principal geodesic analysis* (KPGA), to 1) define more meaningful modes of variation in kernel feature space by explicitly modeling the data on the *Hilbert sphere* in kernel feature space, 2) represent variability using fewer modes, and 3) reduce the curvature of distributions by modeling them explicitly on the Hilbert sphere, instead of modeling them in the ambient space, to avoid artificially large measurements of variability observed in the ambient space.

- A new method for clustering that uses a kernel trick embedded in the generative mixture model for the data on the unit *Hilbert sphere* in kernel feature space. The probability density distribution of the observed variable given the latent variable and



(a) Single group

(b) Multigroup

Fig. 1.3. **Multigroup strategy**: Each point indicates a shape: four shapes in the blue group and four shapes in the red group. (a) Single group: consider only population mean and covariance for two groups of shapes and (b) multigroup: also consider group mean and covariance for each group.

parameters is evaluated by geodesic Mahalanobis distance and performs model fitting using expectation-maximization.

- A *fully* generative approach for semisupervised clustering, rather than the heuristic approach or adding ad hoc penalties.
- A proposed generative semisupervised clustering model that reflects user preferences and maintains a probabilistic interpretation, which allows it to be generalized to take advantage of *alternative* density models or optimization algorithms.
- A proposed semisupervised clustering model that clearly deals with the must-link *and* cannot-link cases in a unified framework and demonstrates that solutions using must-link and cannot-link together or independently are tractable and effective.
- Instead of pairwise constraints, a statistical interpretation of pairwise relationships that allows the model estimation to converge to a distribution that follows user preferences with *less* domain knowledge.
- Within the content of the proposed semisupervised clustering algorithm, a parameter estimation that is very similar to the Gaussian mixture model using expectation-maximization in terms of ease of implementation and efficiency.
- Motivated by the natural organization of population data into multiple groups, the proposal of a novel *hierarchical generative* statistical model for shapes, which represents shapes using pointsets and defines a joint distribution on the population's 1) shape variables (i.e., latent variable) and 2) object-boundary data. The new method solves for optimal point locations, correspondences, and model-parameter values as a *single* optimization problem.
- A new method for maximizing the posterior of shapes using EM and relying on a novel Markov-chain Monte-Carlo algorithm for *sampling* in Kendall shape space.

1.4 Dissertation Organization

Chapter 2 includes the paper, *Kernel principal geodesic analysis*, which has been published in the *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD, 2014)*. The paper in Chapter 3, *Hierarchical Bayesian modeling, estimation, and sampling for multigroup shape analysis*, has been published in the *International Conference on the Medical Image Computing and Computer Assisted Intervention (MICCAI, 2014)*. Chapter 4 is the paper, *Clustering with pairwise relationships:*

a generative approach, which is under review by the *Journal of Machine Learning Research (JMRL)*. This dissertation is organized as follows:

- Chapter 2 provides background for generative models, the expectation-maximization algorithm, and the kernel method. This chapter also presents statistics for the Hilbert sphere in kernel feature space and dimensionality reduction and clustering on a hypersphere in kernel feature space using kernel principal geodesic analysis.

- Chapter 3 presents the hierarchical generative shape model and its application.

- Chapter 4 describes the generative model with pairwise relationships, which is in the general form of a probability distribution. This chapter also demonstrates that the semisupervised GMM is a special case of the proposed model.

CHAPTER 2

KERNEL PRINCIPAL GEODESIC ANALYSIS

2.1 Abstract

Kernel principal component analysis (kPCA) has been proposed as a dimensionality reduction technique that achieves nonlinear, low-dimensional representations of data via mapping to kernel feature space. Conventionally, kPCA relies on Euclidean statistics in kernel feature space. However, Euclidean analysis can make kPCA inefficient or incorrect for many popular kernels that map input points to a *hypersphere* in kernel feature space. To address this problem, this dissertation proposes a novel adaptation of kPCA, namely *kernel principal geodesic analysis* (kPGA), for hyperspherical statistical analysis in kernel feature space. This dissertation proposes tools for statistical analyses on the Riemannian manifold of the Hilbert sphere in reproducing kernel Hilbert space, including algorithms for computing the sample-weighted Karcher mean and eigenanalysis of the sample-weighted Karcher covariance. It then applies these tools to propose novel methods for 1) dimensionality reduction and 2) clustering using mixture-model fitting. The results, on simulated and real-world data, show that kPGA-based methods perform favorably relative to their kPCA-based analogs.

2.2 Introduction

Kernel principal component analysis (kPCA) [40] maps points in *input space* to a (high-dimensional) *kernel feature space* where it estimates a best fitting linear subspace via PCA. This mapping to the kernel feature space is typically denoted by $\Phi(\cdot)$. For many of the most

This chapter is a revised and adapted with permission (copyright by authors) from

Kernel Principal Geodesic Analysis. *Suyash P. Awate, Yen-Yun Yu, and Ross T. Whitaker.* Machine Learning and Knowledge Discovery in Databases Volume 8724 of the series Lecture Notes in Computer Science pages 82-98 (European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, Nancy, France, September 15-19, 2014).

useful and widely used kernels (e.g., Gaussian, exponential, Matern, spherical, circular, wave, power, log, rational quadratic), the input data x gets mapped to a *hypersphere*, or a *Hilbert sphere*, in the kernel feature space. Such a mapping also occurs when using 1) kernel normalization, which is common, e.g., in pyramid match kernel [41], and 2) polynomial and sigmoid kernels when the input points have constant l^2 norm, which is common in digit image analysis [10]. This special structure arises because for these kernels $k(\cdot, \cdot)$, the self-similarity of any data point x equals unity (or some constant), i.e., $k(x, x) = 1$. The kernel defines the inner product in the kernel feature space $\mathcal{F}$, and thus, $\langle \Phi(x), \Phi(x) \rangle_{\mathcal{F}} = 1$, which, in turn, equals the distance of the mapped point $\Phi(x)$ from the origin in $\mathcal{F}$. Thus, all mapped points $\Phi(x)$ lie on a Hilbert sphere in kernel feature space. Fig. 2.1 illustrates this behavior.

The literature shows that for many high-dimensional real-world datasets, where the data representation uses a large number of dimensions, the intrinsic dimension is often quite small, e.g., between 5–20 in [42]–[46]. The utility of kPCA lies in capturing the intrinsic dimension of the data through the few principal (linear) modes of variation in kernel feature space. This dissertation proposes a novel extension of kPCA to model distributions on the Hilbert sphere manifold in kernel feature space. Manifold-based statistical analysis explicitly models data to reside in a lower-dimensional subspace of the ambient space, representing variability in the data more efficiently (fewer degrees of freedom). In this way, the proposed method extends kPCA to 1) define more meaningful modes of variation in kernel feature space by explicitly modeling the data on the Hilbert sphere in kernel feature space, 2) represent variability using fewer modes, and 3) reduce curvature of distributions by modeling them explicitly on the Hilbert sphere, instead of modeling them in the ambient space, to avoid artificially large measurements of variability observed in the ambient space. Fig. 2.2 illustrates this idea.

Typically, Euclidean PCA of spherical data introduces one additional (unnecessary) component, aligned orthogonally to the sphere and proportional to the sectional curvature. In practice, however, PCA in high-dimensional spaces (e.g., kernel feature space) is known to be unstable and prone to error [47], which interacts with the curvature of the Hilbert sphere on which the data reside. Thus, our empirical results demonstrate that the actual gains in our hyperspherical analysis in kernel feature space surpass what we would expect for the low-dimensional case.

While several works in the literature [10], [48]–[51] address the properties and uses of kernel feature spaces, these works do *not* systematically explore this special structure of

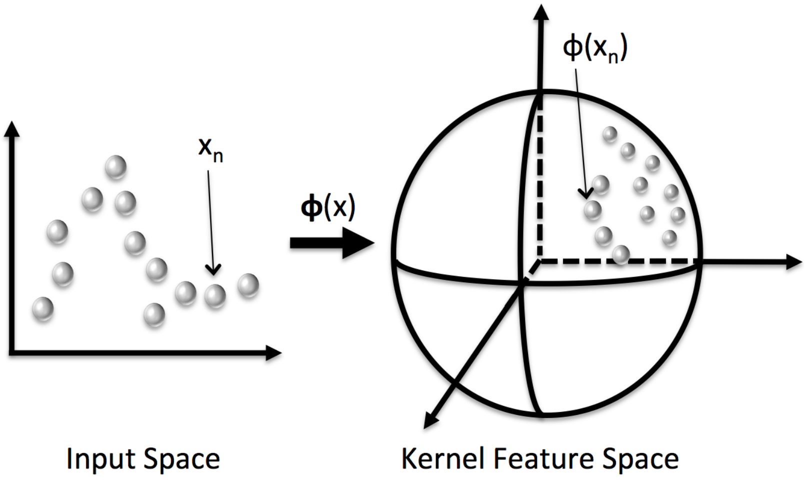


Fig. 2.1. **Map points to a single *orthant* of a Hilbert sphere:** Points in input space get mapped, via several popular Mercer kernels, to a hypersphere or a Hilbert sphere in kernel feature space.

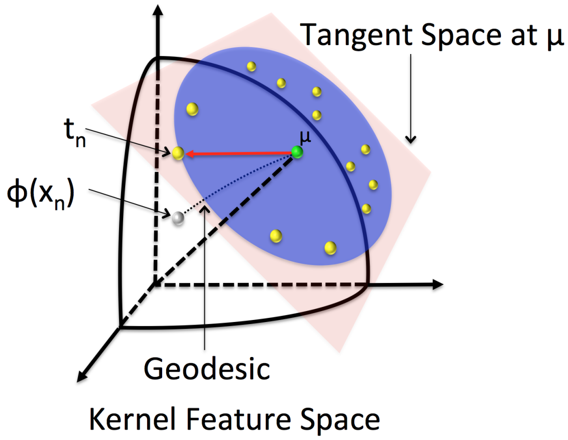


Fig. 2.2. **Principal geodesic analysis on the Hilbert sphere in kernel feature space:** The silver point $\Phi(x_n)$ indicates the data point on hypersphere and the yellow point t_n is the logarithmic map of $\Phi(x_n)$ that locates on the tangent space (red plane) with respect to the sample Karcher mean μ (green point). The purple circle on the tangent space visualizes the sample Karcher covariance.

kernel feature space and its implications for PCA in kernel feature space; that is the focus of this paper. Recently, [49] have, in an independent development, examined the use of the Karcher mean in kernel feature spaces, but they propose a different estimation strategy and they do *not* formulate, estimate, or demonstrate the use of principle components on the sphere, which is the main purpose of this work.

This dissertation makes several contributions. It proposes new formulations and algorithms for computing the sample Karcher mean on a Hilbert sphere in reproducing kernel Hilbert space (RKHS). To analyze sample Karcher covariance, this dissertation proposes a kernel-based PCA on the Hilbert sphere in RKHS, namely, *kernel principal geodesic analysis* (kPGA). It shows that just as kPCA leads to a standard eigenanalysis problem, kPGA leads to a generalized eigenanalysis problem. This dissertation evaluates the utility of kPGA for 1) nonlinear dimensionality reduction and 2) clustering with a Gaussian mixture model (GMM) and an associated expectation-maximization (EM) algorithm on the Hilbert sphere in RKHS. Results on simulated and real-world data show that kPGA-based methods perform favorably with their kPCA-based analogs.

2.3 Related Work

Several areas of related work inform the results in this dissertation. The Karcher mean and associated covariance have recently become important tools for statistical analysis [52]. The algorithm for the Karcher mean proposed in [53] is restricted to analyzing the intrinsic mean and does *not* address how to capture covariance for data lying on spheres, even in finite-dimensional spaces. Other algorithms for the Karcher mean exist and may be more efficient numerically [54]. To capture covariance structure on Riemannian manifolds, Fletcher et al. [55] propose PGA and an associated set of algorithms. Likewise, a small body of work relies on the local geometric structure of Riemannian spaces of covariance matrices for subsequent statistical analysis [56]–[58].

Because many RKHSs are infinite-dimensional, we must acknowledge the problem of modeling distributions in such spaces [59] and the corresponding theoretical problems [60]. Of course, these same theoretical concerns arise in kPCA, and other well-known kernel methods, and thus the justification for this work is similar. First, we may assume or assert that the covariance operator of the mapped data is of trace class or, even more strongly, restricted to a finite-dimensional manifold defined by the cardinality of the input data. Second, the proposed methods are intended primarily for data analysis rather than statistical estimation, and, thus, we intentionally work in the subspace defined by the data (which is limited by the data sample size).

In addition to the dimensionality structure, the Hilbert sphere imposes its own structure and has an associated geometry with underlying theoretical implications. The proposed approach in this dissertation extends PGA [55] to the Hilbert sphere in RKHS. The important geometrical properties of the sphere for the proposed extension concern 1) the geodesic distance between two points, which depends on the arc cosine of their dot product, and 2) the existence and formulation of tangent spaces [61]–[63].

The work in [49] is more directly related to the proposed method, because it uses logarithmic and exponential maps on the Hilbert sphere in RKHS for data analysis. However, [49] does *not* define a mean or a covariance on the Hilbert sphere in RKHS; it also requires the solution of the ill-posed preimage problem. Unlike [49], we define covariance and its low-dimensional approximations on the Hilbert sphere, represented in terms of the Gram matrix of the data, and incorporate this formulation directly into novel algorithms for dimensionality reduction and clustering via EM [64], including geodesic Mahalanobis distance on the Hilbert sphere in RKHS.

We apply the proposed method for 1) dimensionality reduction for machine-learning applications and 2) mixture modeling. This builds on the work in kPCA [40], and therefore represents an alternative to other nonlinear mapping methods, such as Sammon's nonlinear mapping [65], Isomap [66], and other kernel-based methods [67]–[71]. For applications to clustering, the proposed approach generalizes kernel k-means [40] and kernel GMMs [72], where we use formulations of means and/or covariances that respect the hyperspherical geometry of the mapped points in RKHS.

In this chapter, we will first review a generative model with latent variables using expectation-maximization and kernel method. We then introduce the proposed statistical tool, kPGA.

2.4 Background

2.4.1 Generative Model Using Expectation-Maximization

In this section, we review the generative model in subsection 2.4.1.1 and expectation-maximization in subsection 2.4.1.4.

2.4.1.1 Generative Model With Latent Variable

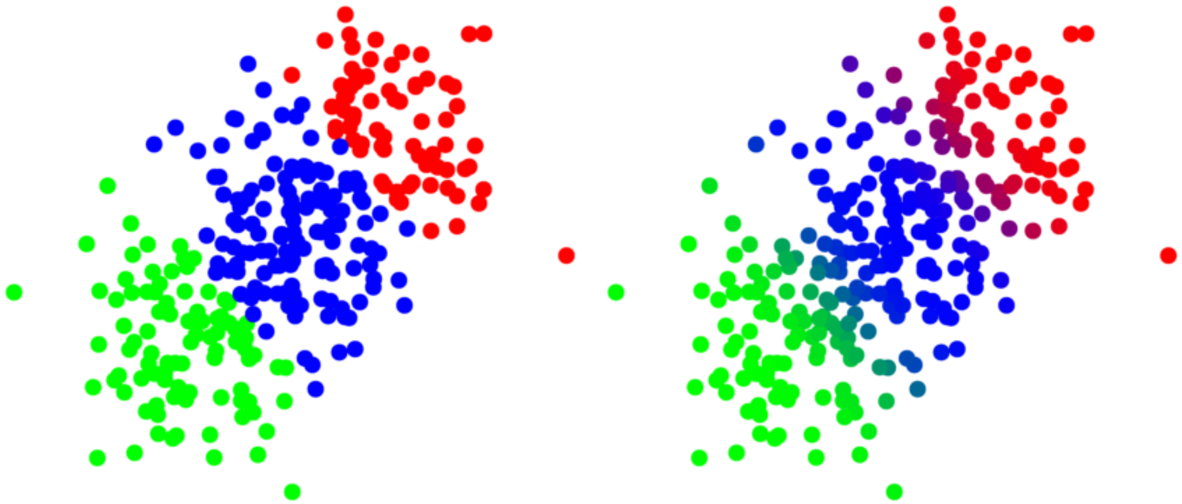
To review the generative model and latent variable, we use mixture distributions as an example (such as the Gaussian mixture), because the mixture model is a natural approach to model the data and an intuitive example to illustrate the theory concerning the function of the discrete latent variable.

The Gaussian mixture model (GMM) is defined as a linear superposition of Gaussian distributions that is motivated by having more insight into the underlining structure of the data compared to using only *single* Gaussian distributions when solving the statistical analysis problem. For clustering, this Gaussian mixture model requires an estimation of the covariance matrix, which allows the Gaussian mixture model to obtain the soft label based on the posterior probability of the latent variable as in Fig. 2.3. On the other hand, the k-means assumes only *spherical* clusters and obtains only a hard label such that each data point is associated with only a single cluster. Thus, the k-means is considered as a special case of mixture models [73].

Suppose we estimate the parameters of a standard GMM, consisting of K components, on a dataset $\mathcal{X}$ of N-samples in d-dimensional space $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^{N}$ and $\mathbf{x}_n \in \mathbb{R}^d$. The GMM is formulated as a joint distribution over the i.i.d. observed samples $\mathbf{x}_n$ and the mixture parameters $\boldsymbol{\Theta} = \{\pi_k, \Theta_k\}_{k=1}^{K}$. Thus, the definition of K components of the GMM over the dataset $\mathcal{X}$ is

$$p(\mathcal{X}|\boldsymbol{\Theta}) := \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\Theta_k) \tag{2.1}$$

where $\mathcal{N}(.)$ denotes a multivariate normal distribution, and $\pi_k \in [0,1]$ is the mixing parameter that indicates the proportion of a single cluster to the mixture model. The



(a) hard label

(b) soft label

Fig. 2.3. **Example of clustering using k-means and GMM**: (a) Result of clustering using k-means, i.e., hard label. (b) Distribution-based clustering (e.g., GMM) is able to obtain the soft label (membership), i.e., the probability of data belonging to the red, green, or blue clusters. The color of each data point has a different saturation to represent membership.

mixing parameter must be subjected to

$$\sum_{k=1}^{K} \pi_k = 1. \tag{2.2}$$

$\Theta_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ includes 1) the mean vector, $\boldsymbol{\mu}_k$, which contains the average of the observations for each variable, and 2) the covariance matrix, $\boldsymbol{\Sigma}_k$, which is the generalization of variance in multidimensional space. $\Theta_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ are associated with the k-th cluster. Each $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ is

$$\boldsymbol{\mu}_k \in \mathbb{R}^d \tag{2.3}$$

$$\boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}. \tag{2.4}$$

Let us introduce the *latent* label set (i.e., not *directly* observed) $\mathcal{Y} = \{\mathbf{y}_n\}_{n=1}^{K}$. Each $\mathbf{y}_n$ is defined as

$$\mathbf{y}_n = [y_n^1, ..., y_n^K]^T, \tag{2.5}$$

which is associated with the dataset $\mathcal{X}$. $\mathbf{y}_n$ is a K-dimensional binary random variable. Each latent variable is subject to

$$\mathbf{y}_n \in \{0, 1\}^K \tag{2.6}$$

and

$$\sum_{k=1}^{K} y_n^k = 1, \tag{2.7}$$

which means that each $\mathbf{y}_n$ has K possible states according to which element $y_n^k = 1$, i.e., $y_n^k = 1$ if and only if the corresponding data point $\mathbf{x}_n$ was generated from the k-th component of the GMM. Therefore, we can define

$$p(y_n^k = 1) := \pi_k \tag{2.8}$$

and the conditional probability of data given the latent variable is

$$p(\mathbf{x}_n | y_n^k = 1) := \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{2.9}$$

The probability that a data point $\mathbf{x}_n$ is generated from a GMM with parameters $\boldsymbol{\Theta}$ can be defined by the marginal distribution of $\mathbf{x}_n$, i.e., summing the joint distribution over all possible $\mathbf{y}_n$, such that

$$
\begin{aligned}
p(\mathbf{x}_n|\boldsymbol{\Theta}) &:= \sum_{\mathbf{y}_n} p(\mathbf{x}_n, \mathbf{y}_n) && (2.10)\\
&= \sum_{\mathbf{y}_n} p(\mathbf{y}_n) p(\mathbf{x}_n|\mathbf{y}_n) && (2.11)\\
&= \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). && (2.12)
\end{aligned}
$$

Therefore, the log of the likelihood of the observed dataset $\mathcal{X}$ governed by the mixture model parameters is

$$
\begin{aligned}
\log \mathcal{L}(\mathcal{X}, \boldsymbol{\Theta}) &:= \log p(\mathcal{X}|\boldsymbol{\Theta}) && (2.13)\\
&= \log \Big(\prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Big) && (2.14)\\
&= \sum_{n=1}^{N} \log \Big(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Big). && (2.15)
\end{aligned}
$$

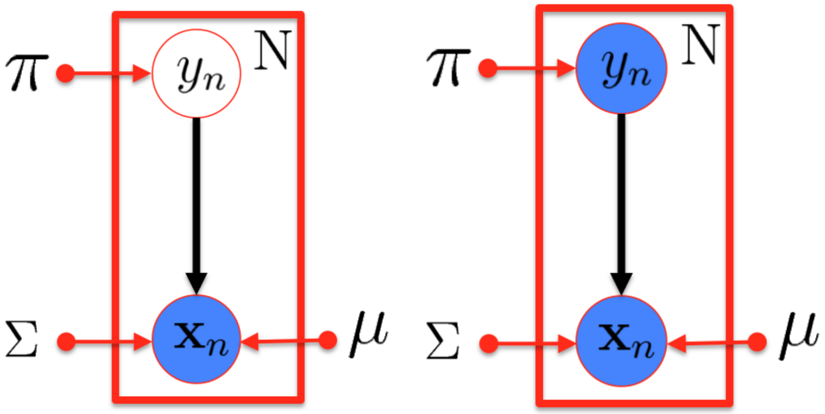
2.4.1.2 Graphical View of GMM

Recall that the GMM assumes that the observed data are drawn *independently from the identical distribution* (i.i.d.), so we can also express this i.i.d. using the graphical model. The graphical model is a probabilistic model that uses a graph to express the conditional dependence structure between random variables, as in Fig. 2.4-(a). The reason for introducing the graphical model here is that it is useful *not only* for visualizing the complex structure of the probability model *but also* for intuitively understanding the properties of the GMM. This dissertation defines the graphical model following [8].

As shown in Fig. 2.4-(a), directed graphs represent the joint probability distributions of the GMM. Each node is represented as a random variable. The blue shaded node $\mathbf{x}_n$ indicates the observed variable and the unshaded node is the unobserved variable, i.e., latent variable $\mathbf{y}_n$. The red box represents N nodes, of which only a single pair variable, $\mathbf{x}_n$ and $\mathbf{y}_n$, is shown. Any link going from node $\mathbf{y}_n$ to another node $\mathbf{x}_n$ means that node $\mathbf{y}_n$ is the parent of node $\mathbf{x}_n$. All parameters are shown by the smaller solid red nodes.

2.4.1.3 Latent Variable View of GMM

The goal is to find the maximum log-likelihood solution in equation (2.15); however, the difficulty in equation (2.15) is that the summation over k occurs inside the logarithm. The



(a) (b)

Fig. 2.4. **Graphical representation of GMM**: The data are sampled i.i.d. from a mixture of the Gaussian distributions given mixing parameters π, mean $\boldsymbol{\mu}$, and covariance matrix $\boldsymbol{\Sigma}$. (a) Observed data $\{\mathbf{x}_n\}_{n=1}^N$ and corresponding *unobserved* variable $\{\mathbf{y}_n\}_{n=1}^N$. (b) Observed data $\{\mathbf{x}_n\}_{n=1}^N$ and corresponding *observed* variable $\{\mathbf{y}_n\}_{n=1}^N$.

trick here is that we can combine the *observed* dataset $\mathcal{X}$ with the *unobserved* dataset (i.e., latent variable) $\mathcal{Y}$ to become a new dataset $\{\mathcal{X}, \mathcal{Y}\}$, which is called the *complete* dataset. The alternative view of the latent variable is that each $\mathbf{y}_n$ denotes a 1-of-K representation, i.e., we know *only* one element is nonzero. The $\mathcal{Y}$ can be viewed as the observed data as well as $\mathcal{X}$. From the graphical model point of view, we can change the node $\mathbf{y}_n$ in Fig. 2.4-(a) to be shaded, as in Fig. 2.4-(b). The node $\mathbf{y}_n$ provides information about which component of GMM generates certain data $\mathbf{x}_n$. Therefore, equations (2.8) and (2.9) can be revised as

$$p(\mathbf{y}_n) := \prod_{k=1}^{K} \pi_k^{y_n^k} \tag{2.16}$$

$$p(\mathbf{x}_n|\mathbf{y}_n) := \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{y_n^k}. \tag{2.17}$$

Now we consider that the latent variable $\mathcal{Y}$ is observed. Equation (2.15) can be revised as

$$\begin{aligned}
\log \mathcal{L}(\mathcal{X}, \mathcal{Y}, \boldsymbol{\Theta}) &:= \log p(\mathcal{X}, \mathcal{Y}|\boldsymbol{\Theta}) \\
&= \log\left(\prod_{k=1}^{K}\prod_{n=1}^{N} p(\mathbf{x}_n, y_n^k)\right) && (2.18)\\
&= \log\left(\prod_{k=1}^{K}\prod_{n=1}^{N} p(\mathbf{x}_n|y_n^k)p(y_n^k)\right) && (2.19)\\
&= \sum_{k=1}^{K}\sum_{n=1}^{N} \log\left([\pi_k\, \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]^{y_n^k}\right) && (2.20)\\
&= \sum_{k=1}^{K}\sum_{n=1}^{N} y_n^k \log\left(\pi_k\, \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) && (2.21)
\end{aligned}$$

where the y_n^k *restricted* the data point $\mathbf{x}_n$ to be generated from the k-th component of the Gaussian distribution, i.e., the *complete-data* log-likelihood function in equation (2.21) is a summation of the K-independent component of the Gaussian distribution. Compared to the original *incomplete* log-likelihood function in equation (2.15), it is clear that equation (2.21) has the simpler representation for the likelihood function.

2.4.1.4 Expectation-Maximization Algorithm

The maximum likelihood estimator is used for estimating the value of the parameters that maximize the likelihood based on the observed data. For example, if we flip a coin a number of times (i.e., observed data), the total number of flips should follow a model (e.g., binomial distribution and its parameters). Different values of parameters are able to fit the observed data more or less well. The goal of maximum likelihood is to find the parameters that can make the model better fit the data. Another intuitive thought is that the maximum likelihood of the Gaussian distribution is simply the minimum distance from all observed data according to the value of the estimated mean and covariance. In most unsupervised problems, such as clustering, the probability model, given the observed data, has a complex structure. Very often we need the latent variables that allow the statistical model to simplify the dependencies. The general technique for seeking the maximum likelihood estimators for the *latent variable model* is expectation-maximization (EM) [74]–[77]. In the following two subsections, 2.4.1.5 and 2.4.1.6, we review model fitting using expectation-maximization when assuming the data i.i.d. from the GMM.

2.4.1.5 E-Step

Although we treat the latent variable $\mathbf{y}_n$ as observed, in practice, we are still unable to directly access any latent variable. For the EM algorithm, however, we can consider another way to employ the latent variable, that is, using the expected value of the latent variable under the posterior distribution of the latent variable, also known as the E-step. In this way, the latent variable has been marginalized out, i.e., we can gain information about the latent variable via the posterior distribution $p(y_n^k|\mathbf{x}_n)$, so the E-step is used to create a distribution for the expectation of the likelihood evaluated using the *current* estimate for the parameters $\mathbf{\Theta}^{\text{old}}$.

$$\begin{aligned}\mathcal{Q}(\mathbf{\Theta}, \mathbf{\Theta}^{\text{old}}) &= \mathbb{E}_{\mathcal{Y}}[\log\ \mathcal{L}] \\ &= \sum_{\mathcal{Y}} p(\mathcal{Y}|\mathcal{X}, \mathbf{\Theta}^{\text{old}}) \log p(\mathcal{X}, \mathcal{Y}|\mathbf{\Theta}). \end{aligned} \tag{2.22}$$

First, we start with the initial values of the parameters and use them to evaluate the posterior distribution, which is taking the expectation of the log-likelihood in equation (2.21) with respect to the posterior distribution of y_n^k and bearing in mind that the latent variable $\mathbf{y}_n$ is a binary variable.

$$\begin{aligned}\mathbb{E}_{y_n^k|\mathbf{x}_n}[y_n^k] &= \sum_{k=1}^{K} y_n^k p(y_n^k|\mathbf{x}_n) && (2.23)\\ &= p(y_n^k|\mathbf{x}_n) && (2.24)\\ &= \frac{p(y_n^k)p(\mathbf{x}|y_n^k)}{\sum_{k'=1}^{K} p(y_n^{k'})p(\mathbf{x}|y_n^{k'})} && (2.25)\\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}, && (2.26)\end{aligned}$$

which can be viewed as the membership, i.e., soft-label, as in Fig. 2.3-(b).

2.4.1.6 M-Step

After the E-step, the EM algorithm keeps the posterior distribution fixed and uses it for maximizing the likelihood function by *revising* current parameters, i.e., updating parameters. Firstly, because the mixing parameter has to satisfy the constraint summation to be one, this determination can be achieved by the Lagrange multiplier.

$$\mathcal{Q} + \lambda\bigg(\sum_{k=1}^{K} \pi_k - 1\bigg). \tag{2.27}$$

λ is the Lagrange multiplier. Taking the derivative of equation (2.27) with respect to π_k,

$$\pi_k = -\sum_{n=1}^{N} \frac{p(y_n^k|\mathbf{x}_n)}{\lambda}. \tag{2.28}$$

By taking the derivative of equation (2.27) with respect to λ and equal to zero, we then can get $\sum_{k=1}^{K} \pi_k = 1$. We use the π_k in equation (2.28) to substitute $\sum_{k=1}^{K} \pi_k = 1$ and obtain $\lambda = -N$, which we then use to eliminate λ in equation (2.28). The mixing parameter is given by

$$\pi_k = \frac{N_k}{N}. \tag{2.29}$$

Let $N_k = \sum_{n=1}^{N} p(y_n^k|\mathbf{x}_n)$. Secondly, we take the derivative of $\mathcal{Q}$ with respect to $\boldsymbol{\mu}_k$ and equal to zero, which gives the closed-form solution for $\boldsymbol{\mu}_k$:

$$\sum_{n=1}^{N} p(y_n^k|\mathbf{x}_n)\big(\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)\big) = 0. \tag{2.30}$$

Then,

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} p(y_n^k|\mathbf{x}_n)\mathbf{x}_n}{N_k}. \tag{2.31}$$

Thirdly, taking the derivative of $\mathcal{Q}$ with respect to $\mathbf{\Sigma}_k^{-1}$ and equal to 0, we can get the closed-form solution for $\mathbf{\Sigma}_k$,

$$\sum_{n=1}^{N} p(y_n^k|\mathbf{x}_n)\Big(\Sigma_k - (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T\Big) = 0. \tag{2.32}$$

Hence:

$$\mathbf{\Sigma}_k = \frac{\sum_{n=1}^{N} p(y_n^k|\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T}{N_k}. \tag{2.33}$$

The algorithm iteration alternates between performing an E-step and M-step until the convergence criterion is satisfied, e.g., the log-likelihood converges.

2.4.1.7 Expectation-Maximization Algorithm in General

As described in the previous subsections, the EM is a two-stage (i.e., E-step and M-step) iterative *optimization* method and is a widely used maximum likelihood estimation technique for statistical models with latent variables. The advantage of EM is that it guarantees maximization of the likelihood function for every iteration, unless it has arrived at the local optima. Consider again a probabilistic model in which a set of observed data $\mathcal{X}$ and a set of latent variables $\mathcal{Y}$ are collected. The log-likelihood $\log \mathcal{L}$ is

$$\begin{aligned}
\log \mathcal{L}(\mathcal{X}, \mathbf{\Theta}) &:= \log\Big(\sum_{\mathcal{Y}} p(\mathcal{X}, \mathcal{Y}|\mathbf{\Theta})\Big) && (2.34)\\
&= \log\Big(\sum_{\mathcal{Y}} q(\mathcal{Y}|\mathcal{X})\frac{p(\mathcal{X}, \mathcal{Y}|\mathbf{\Theta})}{q(\mathcal{Y}|\mathcal{X})}\Big) && (2.35)\\
&\geq \sum_{\mathcal{Y}} q(\mathcal{Y}|\mathcal{X}) \log\Big(\frac{p(\mathcal{X}, \mathcal{Y}|\mathbf{\Theta})}{q(\mathcal{Y}|\mathcal{X})}\Big) && (2.36)\\
&= F(q, \mathbf{\Theta}) && (2.37)
\end{aligned}$$

where $q(\mathcal{Y}|\mathcal{X})$ is a nonnegative distribution and $\sum_{\mathcal{Y}} q(\mathcal{Y}|\mathcal{X}) = 1$. $F(q, \Theta)$ is a function of the distribution $q(\mathcal{Y}|\mathcal{X})$ and parameter $\mathbf{\Theta}$. Moreover, $F(q, \Theta)$ is auxiliary *lower* bound for log-likelihood, because of the Jensen's inequality for the log function in equation (2.36). $F(q, \Theta)$ also allows us to redefine the EM algorithm via decomposing the $F(q, \Theta)$, that is,

$$F(q, \boldsymbol{\Theta}) = \sum_{\mathcal{Y}} q(\mathcal{Y}|\mathcal{X}) \log \left(\frac{p(\mathcal{X}, \mathcal{Y}|\boldsymbol{\Theta})}{q(\mathcal{Y}|\mathcal{X})} \right) \tag{2.38}$$

$$= \sum_{\mathcal{Y}} q(\mathcal{Y}|\mathcal{X}) \log \left(\frac{p(\mathcal{X}, \mathcal{Y}|\boldsymbol{\Theta})}{p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\Theta})} \right) + \sum_{\mathcal{Y}} q(\mathcal{Y}|\mathcal{X}) \log \left(\frac{p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\Theta})}{q(\mathcal{Y}|\mathcal{X})} \right) \tag{2.39}$$

$$= \sum_{\mathcal{Y}} q(\mathcal{Y}|\mathcal{X}) \log p(\mathcal{X}|\boldsymbol{\Theta}) - \sum_{\mathcal{Y}} q(\mathcal{Y}|\mathcal{X}) \log \left(\frac{q(\mathcal{Y}|\mathcal{X})}{p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\Theta})} \right) \tag{2.40}$$

$$= \log \mathcal{L}(\mathcal{X}, \boldsymbol{\Theta}) - KL(q||p) \tag{2.41}$$

where $KL(q||p)$ is the Kullback-Leibler divergence between $q(\mathcal{Y}|\mathcal{X})$ and the posterior distribution of the latent variable $p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\Theta})$. Note $KL(q||p) \geq 0$ and $KL(q||p) = 0$ if and only if $q(\mathcal{Y}|\mathcal{X}) = p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\Theta})$. In equation (2.41), $F(q, \Theta)$ is decomposed into the log-likelihood function and KL-divergence. Consider again, as in Fig. 2.5, that the log-likelihood function is the sum of $F(q, \Theta)$ and KL-divergence. The EM algorithm can be redefined via optimizing equation (2.41). The original E-step is to compute the posterior distribution of the latent variable given the current value of parameters is $\boldsymbol{\Theta}^{\text{old}}$. Now, the **E-step** optimizes $q(\mathcal{Y}|\mathcal{X})$. In other words, the lower bound $F(q, \Theta)$ is maximized when the $KL(q||p) = 0$, i.e., $q(\mathcal{Y}|\mathcal{X}) = p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\Theta})$. Because $KL(q||p) \geq 0$ is always true, it means that $\log \mathcal{L}(\mathcal{X}, \boldsymbol{\Theta}^{\text{old}}) \geq F(q, \boldsymbol{\Theta}^{\text{old}})$ as well. In the **M-step**, $q(\mathcal{Y}|\mathcal{X})$ is fixed and is used to estimate new parameters $\boldsymbol{\Theta}^{\text{new}}$ for maximizing the lower bound $F(q, \Theta)$. Further, it does *indeed* increase the log-likelihood $\log \mathcal{L}$, (except that it is already at a local optimal) because the KL-divergence is not equal to zero, i.e., $q(\mathcal{Y}|\mathcal{X}) = p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\Theta}^{\text{old}}) \neq p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\Theta}^{\text{new}})$ (as the M-step in Fig. 2.5).

2.4.2 Kernel Method

In this subsection, we introduce the kernel method, reproducing kernel Hilbert space, and kernel principal component analysis.

The *kernel method* has been widely used in learning algorithms since the last decade [10], [78], [79], because this method has a more clear mathematical interpretation than the neural network in the pattern recognition community. Basically, the kernel method functions by mapping (e.g., feature map) the data from input space into high-dimensional space (e.g., kernel feature space), where a linear relation between the data exists. By allowing analysis of the data in a linear way via the kernel method, it is clear that the kernel method facilitates the following statistical learning analyses, such as clustering. Here we use an example for intuitively understanding the motivation of the kernel method. Fig. 2.6 illustrates a simple

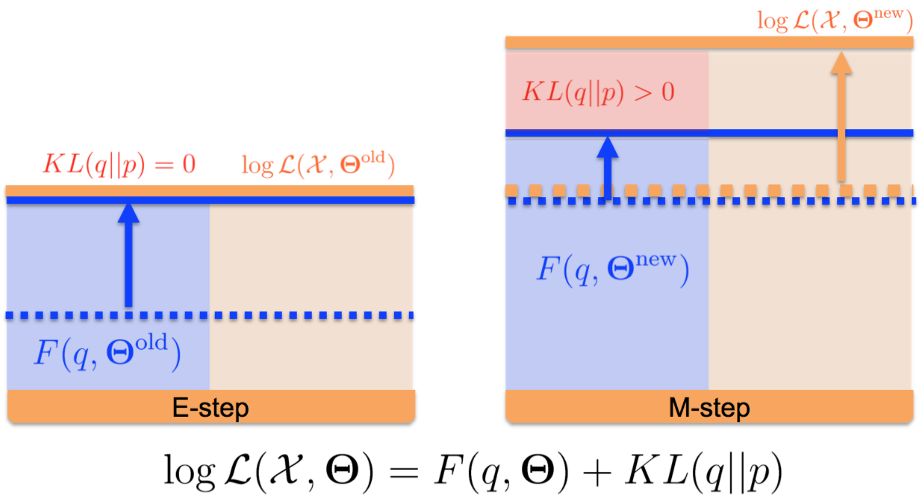

$KL(q||p) = 0$
$\log \mathcal{L}(\mathcal{X}, \mathbf{\Theta}^{\text{old}})$
$F(q, \mathbf{\Theta}^{\text{old}})$
E-step
$\log \mathcal{L}(\mathcal{X}, \mathbf{\Theta}^{\text{new}})$
$KL(q||p) > 0$
$F(q, \mathbf{\Theta}^{\text{new}})$
M-step
$\log \mathcal{L}(\mathcal{X}, \mathbf{\Theta}) = F(q, \mathbf{\Theta}) + KL(q||p)$

Fig. 2.5. **Illustration of expectation-maximization algorithm:** The log-likelihood $\log \mathcal{L}(\mathcal{X}, \mathbf{\Theta})$ can be decomposed into the lower bound $F(q, \mathbf{\Theta})$ and $KL(q||p)$. In the **E-step**, the current parameter $\mathbf{\Theta}^{\text{old}}$ is fixed. The q function is optimized to be equal to the posterior distribution $p(\mathcal{Y}|\mathcal{X}, \mathbf{\Theta}^{\text{old}})$, i.e., $KL(q||p) = 0$, which increases the lower bound to be equal to the log-likelihood function, i.e., $\log \mathcal{L}(\mathcal{X}, \mathbf{\Theta}^{\text{old}}) = F(q, \mathbf{\Theta}^{\text{old}})$. In the **M-step**, the q is fixed for updating the new parameters $\mathbf{\Theta}^{\text{new}}$, which updates the lower bound $F(q, \mathbf{\Theta}^{\text{new}})$ unless it is already at a local optimum. Also, because the q is estimated using the $\mathbf{\Theta}^{\text{old}}$, the p is computed by $\mathbf{\Theta}^{\text{new}}$. It does indeed cause the $KL(q||p) > 0$, i.e., increases the log-likelihood $\log \mathcal{L}(\mathcal{X}, \mathbf{\Theta}^{\text{new}})$.

example of data points originally in one-dimensional input space, but clearly we cannot find any straight line that can separate the data (orange circles and cyan triangles) into two clusters. However, if we map the data into two-dimensional space via a feature map utilizing a polynomial feature (e.g., each data point gets a nonlinear map $[x] \rightarrow [x, x^2]^T$), we can find a straight line (red dashed line in Fig. 2.6) to separate the data points. This example shows the power of the kernel method. However, in practice, the above method is problematic because the data get mapped into a very high-dimensional space. For example, if the data are a small size image (e.g., size is 28 by 28) and the parameter of the polynomial feature map is $d = 4$, the data will be mapped into the space where the total number of dimensions is $\geq 10^{10}$. De facto, there is a way to avoid this computation problem in such high-/infinite-dimensional space. If a model is based on the nonlinear feature map $\Phi(x)$ and there is a *kernel* function, or just said *kernel* satisfies Mercer's theorem [80], we can get

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}}. \tag{2.42}$$

The concept of equation (2.42) is that we can compute the inner products in infinite-dimensional space without knowing the feature map, i.e., implicitly map. In this way,

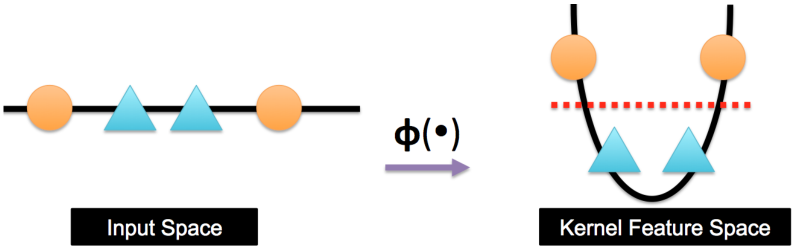


Fig. 2.6. **Illustration of the kernel method:** Each one-dimensional data point gets explicitly mapped ($[x] \rightarrow [x, x^2]^T$) into a two-dimensional space, where the linear hyperplane (red dashed line) exists for separating the datasets into two clusters (orange circles cluster and cyan triangles cluster).

the feature map $\Phi(\cdot)$ is no longer important, because the inner products $\langle \Phi(x), \Phi(x) \rangle_{\mathcal{F}}$ in infinite-dimensional space are equal to $k(x, x')$, which is also known as the *kernel trick* [10], [78], [79], [81]. In other words, if we have any equation that is formulated as inner products, then the term *inner products* can be replaced with some other selection of the Mercer kernel. The $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ indicates the inner products in kernel feature space, i.e., RKHS, which we will introduce in the next section.

2.4.2.1 Reproducing Kernel Hilbert Space (RKHS)

The kernel trick is widely used for dimensional reduction, clustering, and many other pattern recognition problems. In this subsection, we will introduce the reproducing kernel Hilbert space (RKHS). Starting with the kernel rather than the feature map, suppose we have a kernel. How can we construct the kernel feature space where the kernel trick is held, i.e., the inner product in the kernel feature space, or what is the Hilbert space representation of kernels? Before discussing the details of RKHS, we first introduce the *gram matrix* K. Let $\{x_n\}_{n=1}^N$ be a set of observations in input space. Then, the gram matrix is an N by N matrix with the element

$$K_{ij} := k(x_i, x_j). \tag{2.43}$$

In linear algebra, a *symmetric* N by N real matrix A is considered to be *positive definite* if $c^T Ac$ is positive for every nonzero column vector c with N real number elements. A *symmetric positive definite kernel* is the generalization of a symmetric positive definite matrix. If a kernel function k is able to give rise to a symmetric positive definite gram matrix K, we call k a symmetric positive definite kernel. Further, a symmetric positive definite kernel k is also a *reproducing* kernel [10], which provides a good example to explain the reproducing kernel on page 33 [10], [82]–[85]. The general concept of a reproducing kernel is that of a *representer of evaluation*, i.e., for any function f in Hilbert space, $f(x) = \langle k(x,\cdot), f(\cdot)\rangle$. The reproducing kernel satisfies *symmetry*, *bilinearity*, and *positive definite*. By using such a reproducing kernel, we can turn each data point into a function on the domain defined by a set of training data. In this way, each data point is represented by its similarity to all other data points, i.e., an inner product space is constructed for the feature space associated with a reproducing kernel.

$$\langle \Phi(x), \Phi(x')\rangle_{\mathcal{F}} = k(x, x'). \tag{2.44}$$

Hilbert space is a vector space endowed with an inner product that it is *complete*, i.e., the norm corresponding to the inner product. Therefore, an RKHS is a Hilbert space with a reproducing kernel defined on the data domain and expanded in terms of a symmetric positive definite kernel (i.e., reproducing kernel).

2.4.2.2 Kernel Principal Component Analysis

Kernel principal component analysis (kPCA) is a nonlinear version of principal component analysis (PCA) using the kernel method. In other words, kernel PCA performs linear PCA in RKHS. Fig. 2.7 is a good illustration of kernel PCA in which the relation of the data points is nonlinear in input space. Instead of directly working in that space, we can first map the data points into RKHS where the linear relation between the data points exists and then perform PCA in RKHS. Fig. 2.7 also shows that the first principal component (red arrow) can capture the largest variance in RKHS in a linear way, which means that the first principal component in RKHS can be viewed as a nonlinear principal component in input space. To understand the utility of kPCA, let us start with a set of observations $\{\Phi(x_n)\}_{n=1}^{N}$ that have a zero mean in RKHS. Thus, the sample covariance is defined as

$$C := \frac{1}{N}\sum_{n} \Phi(x_n)\Phi(x_n)^T. \tag{2.45}$$

For the kPCA, we want to find the eigenvector v and eigenvalue λ. If λ is a positive eignevalue and v is a corresponding eigenvector, then

$$v = \frac{Cv}{\lambda} = \sum_{n} \alpha_n \Phi(x_n) \tag{2.46}$$

where

$$\alpha_n = \frac{\langle \Phi(x_n), v\rangle_{\mathcal{F}}}{N\lambda}.$$

From equation (2.46), the problem is reduced from finding the eigenvector v to solving the α_n, which means that the α_n leads to a dual eigenvalue problem that is the eigenproblem of the gram matrix instead of the eigenproblem of covariance,

$$K\alpha = \lambda\alpha, \tag{2.47}$$

where $\alpha = \{\alpha_n\}_{n=1}^{N}$ is the eigenvector of the gram matrix K. Evaluating the projection from a new data point $\Phi(x')$ onto the k-th principal component can be done via a kernel trick,

$$\langle \Phi(x'), v_k \rangle_{\mathcal{F}} = \sum_n \alpha_n \langle \Phi(x'), \Phi(x_n) \rangle_{\mathcal{F}} = \sum_n \alpha_n k(x', x_n). \tag{2.48}$$

In general, equation (2.48) is computationally far less expensive than explicitly doing the inner product in the feature space.

However, as in Fig. 2.8, for many of the most widely used kernels, such as one of the most popular kernels, the Gaussian kernel, all data points have a unitary norm in RKHS. To address this issue, this dissertation proposes a new nonlinear statistical tool to analyze this particular geometry in RKHS. Fig. 2.8 illustrates that the concept is to employ a geodesic distance on the hypersphere instead of using the Euclidean-based tool, e.g., kernel PCA.

2.5 Geometry of the Hilbert Sphere in RKHS

Many popular kernels are associated with a RKHS that is infinite-dimensional. Thus, the analysis in this chapter focuses on such spaces. Nevertheless, the analogous theory holds for other important kernels (e.g., normalized polynomial) where the RKHS is finite-dimensional.

Let X be a random variable taking values x in *input space* $\mathcal{X}$. Let $\{x_n\}_{n=1}^{N}$ be a set of observations in input space. Let $k(\cdot, \cdot)$ be a real-valued Mercer kernel with an associated map $\Phi(\cdot)$ that maps x to $\Phi(x) := k(\cdot, x)$ in an RKHS $\mathcal{F}$ [10], [86]. Consider two points in RKHS: $f := \sum_{i=1}^{I} \alpha_i \Phi(x_i)$ and $f' := \sum_{j=1}^{J} \beta_j \Phi(x_j)$. The inner product $\langle f, f' \rangle_{\mathcal{F}} := \sum_{i=1}^{I} \sum_{j=1}^{J} \alpha_i \beta_j k(x_i, x_j)$. The norm $\|f\|_{\mathcal{F}} := \sqrt{\langle f, f \rangle_{\mathcal{F}}}$. When $f, f' \in \mathcal{F} \backslash \{0\}$, let $f \otimes f'$ be the rank-one operator defined as $f \otimes f'(h) := \langle f', h \rangle_{\mathcal{F}} f$. Let $Y := \Phi(X)$ be the random variable taking values y in RKHS.

Assuming Y is bounded and assuming the expectation and covariance operators of Y exist and are well defined, kPCA uses observations $\{y_n := \Phi(x_n)\}_{n=1}^{N}$ to estimate the eigenvalues, and associated eigenfunctions, of the covariance operator of Y [40], [87]. The analysis in this chapter applies to kernels that map points in input space to a Hilbert sphere in RKHS, i.e., $\forall x : k(x, x) = \kappa$, a constant (without loss of generality, we assume $\kappa = 1$).

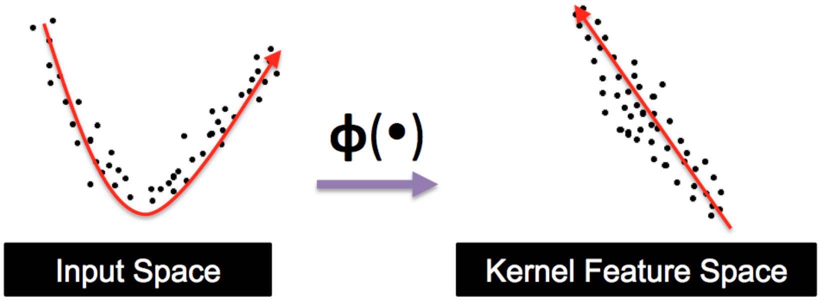


Fig. 2.7. **Illustration of kernel principal component analysis:** Kernel principal component analysis is the nonlinear version of principal component analysis. In the image on the right, the linear representation (red arrow) is the first principal component that has the largest possible variance in kernel feature space, which corresponds to the nonlinear one (left) in input space.

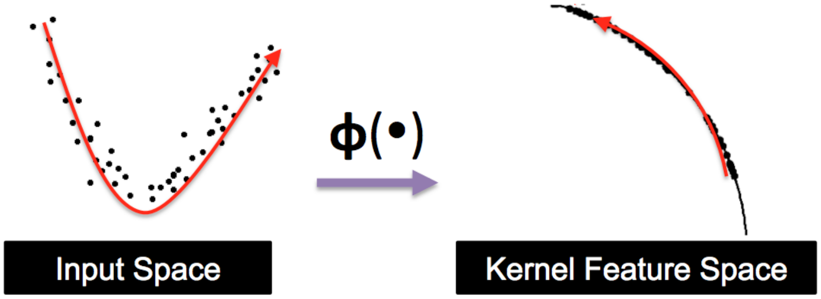


Fig. 2.8. **The motivation of kernel principal geodesic analysis:** As in the image on the right, for many of the most useful kernels (e.g., Gaussian, exponential, Matern, spherical, circular, wave, power, log, rational quadratic), the input data x gets mapped to a *hypersphere* in the kernel feature space, i.e., the self-similarity of any data point x equals unity $k(x, x) = 1$. This dissertation proposes to use principal geodesic analysis [55] in kernel feature space instead of principal component analysis as in Fig. 2.7 (right).

For such kernels, the proposed kPGA modifies kPCA using statistical modeling on the Riemannian manifold of the unit Hilbert sphere [88], [89] in RKHS.

2.5.1 Logarithmic Map in RKHS

Consider a and b on the unit Hilbert sphere in RKHS represented, in general, as $a := \sum_n \gamma_n \Phi(x_n)$ and $b := \sum_n \delta_n \Phi(x_n)$. In Fig. 2.9, the *logarithmic map*, or Log map, of a with respect to b is the vector

$$\mathrm{Log}_b(a) = \frac{a - \langle a, b \rangle_{\mathcal{F}} b}{\| a - \langle a, b \rangle_{\mathcal{F}} b \|_{\mathcal{F}}} \arccos(\langle a, b \rangle_{\mathcal{F}}) = \sum_n \zeta_n \Phi(x_n), \tag{2.49}$$

where

$$\forall n : \zeta_n \in \mathbb{R}.$$

Clearly, $\mathrm{Log}_b(a)$ can always be written as a weighted sum of the vectors $\{\Phi(x_n)\}_{n=1}^{N}$. The *tangent vector* $\mathrm{Log}_b(a)$ lies in the *tangent space*, at b, of the unit Hilbert sphere. The tangent space to the Hilbert sphere in RKHS inherits the same structure (inner product) as the ambient space and, thus, is also an RKHS. The geodesic distance between a and b is $d_g(a, b) = \|\mathrm{Log}_b(a)\|_{\mathcal{F}} = \|\mathrm{Log}_a(b)\|_{\mathcal{F}}$.

2.5.2 Exponential Map in RKHS

Now, consider a tangent vector $t := \sum_n \beta_n \Phi(x_n)$ lying in the tangent space at b. Fig. 2.10 visualizes the *exponential map*, or Exp map, of t with respect to b is

$$\mathrm{Exp}_b(t) = \cos(\|t\|_{\mathcal{F}}) b + \sin(\|t\|_{\mathcal{F}}) \frac{t}{\|t\|_{\mathcal{F}}} = \sum_n \omega_n \Phi(x_n), \tag{2.50}$$

where

$$\forall n : \omega_n \in \mathbb{R}.$$

Clearly, $\mathrm{Exp}_b(t)$ can always be written as a weighted sum of the vectors $\{\Phi(x_n)\}_{n=1}^{N}$. $\mathrm{Exp}_b(t)$ maps a tangent vector t to the unit Hilbert sphere, i.e., $\|\mathrm{Exp}_b(t)\|_{\mathcal{F}} = 1$.

2.6 PCA on the Hilbert Sphere in RKHS

This section proposes the kPGA algorithm for PCA on the unit Hilbert sphere in RKHS.

2.6.1 Sample Karcher Mean

The sample Karcher mean on Riemannian manifolds is a consistent estimator of the theoretical Karcher mean of the underlying random variable [90], [91]. The sample-weighted

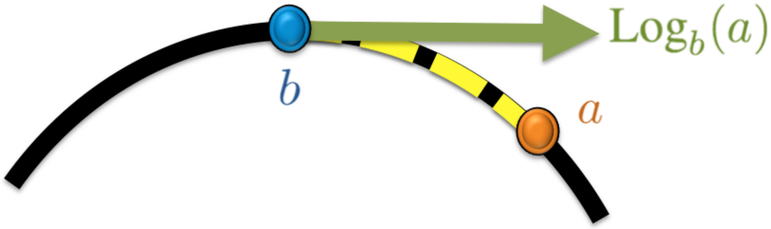


Fig. 2.9. **Illustration of the logarithmic map:** The *logarithmic map* of a with respect to b.

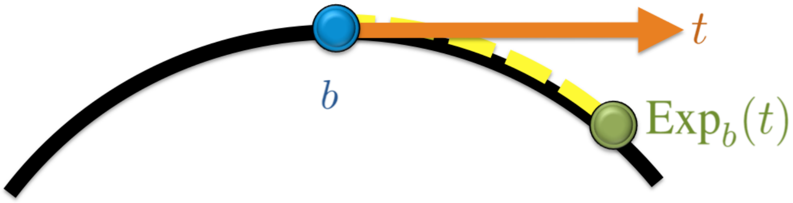


Fig. 2.10. **Illustration of the exponential map:** The *exponential map* of a tangent point t with respect to b.

Karcher mean of a set of observations $\{y_m\}_{m=1}^M$, on the unit Hilbert sphere in RKHS, with associated weights $\{p_m \in \mathbb{R}^+\}_{m=1}^M$ is defined as

$$\mu := \arg\min_{\nu} \sum_m p_m d_g^2(\nu, y_m). \tag{2.51}$$

The existence and uniqueness properties of the Karcher mean on the Riemannian manifold of the unit Hilbert sphere are well studied [92]–[94]; a study on finite-dimensional Hilbert spheres appears in [53]. The sample Karcher mean on a Hilbert sphere exists and is unique if the pointset is contained within 1) an open convex Riemannian ball of radius $\pi/2$ [94], i.e., an open hemisphere, or 2) a similar closed ball if one of the points lies in its interior [53]. Thus, the sample Karcher mean exists and is unique for all kernels that map points within a single orthant of the Hilbert sphere in RKHS; this is true for all positive-valued kernels, e.g., the Gaussian kernel.

Clearly, a Karcher mean μ must lie within the space spanned by $\{y_m\}_{m=1}^M$; if not, we could project the assumed "mean" ν' onto the span of $\{y_m\}_{m=1}^M$ and reduce all distances $d_g(y_m, \nu')$ on the Hilbert sphere because of the spherical Pythagoras theorem, thereby resulting in a more optimal mean ν'' with $d_g(y_m, \nu'') < d_g(y_m, \nu'), \forall m$ and a contradiction to the initial assumption. Therefore, if the points y_m are represented using another set of points $\{\Phi(x_n)\}_{n=1}^N$, i.e., $\forall m, y_m := \sum_n w_{mn}\Phi(x_n)$, then the mean μ can be represented as $\mu = \sum_n \xi_n \Phi(x_n)$, where $\forall n : \xi_n \in \mathbb{R}$.

We propose the following gradient-descent algorithm to compute the mean μ:

1) **Input:** A set of points $\{y_m\}_{m=1}^M$ on the unit Hilbert sphere in RKHS. Weights $\{p_m\}_{m=1}^M$. As described previously, we assume that, in general, each y_m is represented using another set of points $\{\Phi(x_n)\}_{n=1}^N$ and weights w_{mn} on the unit Hilbert sphere in RKHS, i.e., $y_m := \sum_n w_{mn}\Phi(x_n)$.

2) Initialize iteration count: $i = 0$. Initialize the mean estimate to

$$\mu^0 = \frac{\sum_m p_m y_m}{\|\sum_m p_m y_m\|_{\mathcal{F}}} = \sum_n \xi_n \Phi(x_n), \tag{2.52}$$

where

$$\xi_n = \frac{\sum_m p_m w_{mn}}{\|\sum_m p_m y_m\|_{\mathcal{F}}}. \tag{2.53}$$

3) Iteratively update the mean estimate until convergence by a) taking the Log maps of all points with respect to the current mean estimate, b) performing a weighted

average of the resulting tangent vectors, and c) taking the Exp map of the weighted average scaled by a step size τ^i, i.e.,

$$\mu^{i+1} = \mathrm{Exp}_{\mu^i}\left(\frac{\tau^i}{M}\sum_m p_m \mathrm{Log}_{\mu^i}(y_m)\right), \tag{2.54}$$

where

$$\tau^i \in (0,1). \tag{2.55}$$

4) **Output:** Mean μ lying on the unit Hilbert sphere in RKHS.

In practice, we use a gradient-descent algorithm with an adaptive step size τ^i such that the algorithm 1) guarantees that the objective-function value is nonincreasing in every iteration and 2) increases/decreases the step size of each iteration to aid faster convergence. We detect convergence as the point when the objective function cannot be reduced using any nonzero step size. Typically, in practice, a few iterations suffice for convergence.

The convergence of gradient descent for finding Karcher means has been studied [53],[95]. In certain conditions, such as those described earlier when the sample Karcher mean on a Hilbert sphere is unique, the objective function becomes convex [96], which leads the gradient descent to the global minimum.

2.6.2 Sample Karcher Covariance and Eigenanalysis

Given the sample-weighted Karcher mean μ, consider a random variable $Z := \mathrm{Log}_\mu(Y)$ taking values in the tangent space at μ. Assuming that both the expectation and covariance operators of Z exist and are well defined (this follows from the similar assumption on Y), the sample-weighted Karcher covariance operator, in the tangent space at μ, is

$$C := (1/M)\sum_m p_m z_m \otimes z_m, \tag{2.56}$$

where

$$z_m := \mathrm{Log}_\mu(y_m).$$

Because the tangent space is an RKHS, the theoretical analysis of covariance in RKHS in standard kPCA [87], [97] applies to C as well (note that the set $\{z_m\}_{m=1}^M$ is empirically centered by construction; i.e., $\sum_m z_m = 0$). Thus, as the sample size $M \to \infty$, the partial sums of the empirically computed eigenvalues converge to the partial sums of the eigenvalues of the theoretical covariance operator of Z.

Using the Log map representation in Section 2.5, $z_m = \sum_{n'} \beta_{n'm}\Phi(x_{n'})$ leading to

$$C = \sum_{n'}\sum_{n''} E_{n'n''}\Phi(x_{n'}) \otimes \Phi(x_{n''}), \tag{2.57}$$

where

$$E_{n'n''} = \frac{1}{M}\sum_{m} p_m \beta_{n'm}\beta_{n''m}.$$

If λ is a positive eigenvalue of C and v is the corresponding eigenfunction, then

$$v = \frac{Cv}{\lambda} = \frac{1}{\lambda}\sum_{n'}\sum_{n''} E_{n'n''}\Phi(x_{n'}) \otimes \Phi(x_{n''})v = \sum_{n'} \alpha_{n'}\Phi(x_{n'}), \tag{2.58}$$

where

$$\alpha_{n'} = \sum_{n''} \frac{E_{n'n''}}{\lambda}\langle \Phi(x_{n''}), v\rangle_{\mathcal{F}}.$$

Thus, any eigenfunction v of C lies within the span of the set of points $\{\Phi(x_n)\}_{n=1}^{N}$ used to represent $\{y_m\}_{m=1}^{M}$. For any $\Phi(x_\eta) \in \{\Phi(x_n)\}_{n=1}^{N}$ and the eigenfunction v,

$$\langle \Phi(x_\eta), Cv\rangle_{\mathcal{F}} = \lambda\langle \Phi(x_\eta), v\rangle_{\mathcal{F}}. \tag{2.59}$$

Hence,

$$\langle \Phi(x_\eta), \sum_{n'}\sum_{n''} E_{n'n''}\Phi(x_{n'})\otimes\Phi(x_{n''})\sum_{n'''}\alpha_{n'''}\Phi(x_{n'''})\rangle_{\mathcal{F}} = \lambda\langle \quad \Phi(x_\eta), \sum_{n'''}\alpha_{n'''}\Phi(x_{n'''})\rangle_{\mathcal{F}}. \tag{2.60}$$

Thus,

$$\sum_{n'''}\left(\sum_{n'} K_{\eta n'}\sum_{n''} E_{n'n''}K_{n''n'''}\right)\alpha_{n'''} = \lambda\sum_{n'''} K_{\eta n'''}\alpha_{n'''}, \tag{2.61}$$

where

$$K_{ij} := \langle \Phi(x_i), \Phi(x_j)\rangle_{\mathcal{F}}$$

is the element in row i and column j of the gram matrix K. Considering E and K as $N \times N$ real matrices and defining $F := EK$ and $G := KF$ leads to

$$\sum_{n''} E_{n'n''}K_{n''n'''} = F_{n'n'''} \text{ and } \sum_{n'} K_{\eta n'}\sum_{n''} E_{n'n''}K_{n''n'''} = G_{\eta n'''}. \tag{2.62}$$

Therefore, the left-hand side of equation (2.59) equals $G_{\eta\bullet}\alpha$, where 1) $G_{\eta\bullet}$ is the η^{th} row of the $N \times N$ matrix G and 2) α is the $N \times 1$ column vector with the n^{th} component as

α_n. Similarly, the right-hand side of equation (2.59) equals $K_{\eta\bullet}\alpha$, where $K_{\eta\bullet}$ is the η^{th} row of the $N \times N$ matrix K. Using equation (2.59) to form one equation for all $\eta = 1, \cdots, N$, gives the following generalized eigenanalysis problem:

$$G\alpha = \lambda K \alpha. \tag{2.63}$$

If $k(\cdot,\cdot)$ is a symmetric positive-*definite* (SPD) Mercer kernel and the points $\{\Phi(x_n)\}_{n=1}^{N}$ are *distinct*, then K is SPD (hence, invertible) and the generalized eigenanalysis problem reduces to the standard eigenanalysis problem

$$EK\alpha = \lambda\alpha. \tag{2.64}$$

Thus, 1) the eigenvalues $\{\lambda_n\}_{n=1}^{N}$ are the same as the eigenvalues of the sample covariance operator C, and 2) each eigenvector α gives one eigenfunction of C through equation (2.58). Note that standard kPCA requires eigen decomposition of the (centralized) matrix K.

The definition of the sample covariance operator C implies that the rank of C is upper bounded by the sample size M. Because the eigenvalues of C are the same as those for EK or for the pair (G, K), if $M < N$, then the rank of the $N \times N$ matrices EK and G is also upper bounded by M. While K is an $N \times N$ symmetric positive (semi) definite matrix of rank at-most N, E is an $N \times N$ symmetric positive (semi) definite matrix of rank at-most M because $E = BPB^T$, where 1) B is a $N \times M$ matrix where $B_{nm} = \beta_{nm}$ and 2) P is an $M \times M$ diagonal matrix where $P_{mm} = p_m/M$.

2.6.3 Kernel Principal Geodesic Analysis (kPGA) Algorithm

We summarize the proposed **kPGA** algorithm below.

1) **Input:** a) A set of points $\{y_m\}_{m=1}^{M}$ on the unit Hilbert sphere in RKHS. b) Weights $\{p_m\}_{m=1}^{M}$. As described previously, we assume that, in general, each y_m is represented using another set of points $\{\Phi(x_n)\}_{n=1}^{N}$ and weights w_{mn} on the unit Hilbert sphere in RKHS, i.e., $y_m := \sum_n w_{mn}\Phi(x_n)$.

2) Compute the gram matrix K.

3) Compute the Karcher mean μ using the algorithm in Section 2.6.1.

4) Compute the matrix E or $G = KEK$ as described in Section 2.6.2.

5) To analyze the Karcher covariance, perform eigenanalysis for the linear system $G\alpha = \lambda K\alpha$ or $EK\alpha = \lambda\alpha$ to give eigenvalues $\{\lambda_\eta\}_{\eta=1}^{N}$ (sorted in nonincreasing order) and eigenvectors $\{\alpha_\eta\}_{\eta=1}^{N}$.

6) **Output:** a) Mean μ lying on the unit Hilbert sphere in RKHS. b) Principal components or eigenfunctions $\{v_n = \sum_{n'} \alpha_{\eta n'} \Phi(x_{n'})\}_{n=1}^{N}$ in the tangent space at μ. c) Eigenvalues $\{\lambda_n = \lambda_\eta\}_{n=1}^{N}$ capturing variance along principal components.

2.7 Nonlinear Dimensionality Reduction

This section proposes kPGA-based algorithms for nonlinear dimensionality reduction. First, we propose the following algorithm for dimensionality reduction using kPGA:

1) **Input:** A set of points $\{x_n\}_{n=1}^{N}$ along with their maps $\{\Phi(x_n)\}_{n=1}^{N}$ on the unit Hilbert sphere in RKHS. Weights $\{p_n = 1\}_{n=1}^{N}$.

2) Apply the kPGA algorithm in Section 2.6.2 to the observed sample $\{\Phi(x_n)\}_{n=1}^{N}$ to compute mean μ, eigenvalues $\{\lambda_n\}_{n=1}^{N}$ (sorted in nonincreasing order), and corresponding eigenfunctions $\{v_n\}_{n=1}^{N}$.

3) Select the largest $Q < N$ eigenvalues $\{\lambda_q\}_{q=1}^{Q}$ that capture a certain fraction of energy in the eigenspectrum. Select the corresponding subspace $\mathbb{G}_Q =< v_1, \cdots, v_Q >$.

4) Project the Log map of each point $\Phi(x_n)$ on the subspace $\mathbb{G}_Q$ to give the embedding coordinates $e_{nq} := \langle \mathrm{Log}_\mu \Phi(x_n), v_q \rangle_{\mathcal{F}}$ and projected tangent vectors $t_n = \sum_q e_{nq} v_q$ in the tangent space at the mean μ.

5) Take the Exp map of projections $\{t_n\}_{n=1}^{N}$ to produce $\{y_n = \mathrm{Exp}_\mu(t_n)\}_{n=1}^{N}$ lying within a Q-dimensional subsphere on the unit Hilbert sphere in RKHS.

6) **Output**: Embedding subspace (lower dimensional) $\mathbb{G}_Q$, embedding coordinates $\{(e_{n1}, \cdots, e_{nQ})\}_{n=1}^{N}$, and (re)mapped points on the Hilbert subsphere $\{y_n\}_{n=1}^{N}$.

2.8 Clustering Using Mixture Modeling and Expectation-Maximization

This section proposes kPGA-based algorithms for clustering using a mixture model fitted using expectation-maximization.

2.8.1 Mixture Model on Hilbert Sphere in RKHS

We now propose an algorithm for clustering a set of points $\{x_n\}_{n=1}^{N}$, into a fixed number of clusters by fitting a mixture model on the unit Hilbert sphere in RKHS.

The proposed approach entails mixture modeling in a finite-dimensional subsphere of the unit Hilbert sphere in RKHS, after the dimensionality reduction of the points $\{\Phi(x_n)\}$ to a new set of points $\{y_n\}$ (as in Section 2.7). Modeling PDFs on Hilbert spheres entails fundamental trade-offs between model generality and the viability of the underlying parameter estimation. For instance, although Fisher-Bingham probability density functions (PDF) on $\mathbb{S}^d$ are able to model generic anisotropic distributions (anisotropy around the mean) using $O(d^2)$ parameters, their parameter estimation may be intractable [98]–[100]. On the other hand, parameter estimation for the $O(d)$-parameter von Mises-Fisher PDF is tractable [98], but this PDF can model only isotropic distributions. We take another approach that uses a tractable approximation of a normal law on a Riemannian manifold [101], allowing modeling of anisotropic distributions through its covariance parameter in the tangent space at the mean. Thus, the proposed PDF evaluated at $\Phi(x)$ is

$$P(\Phi(x)|\mu, C) \doteq \frac{\exp\left(-0.5 d_g^2(\mu, \Phi(x); C)\right)}{((2\pi)^{Q/2}|C|^{1/2})} \tag{2.65}$$

where

$$|C| = \Pi_{q=1}^{Q} \lambda_q \tag{2.66}$$

and $d_g(\mu, \nu; C)$ is the *geodesic Mahalanobis distance* between the point $\Phi(x)$ and mean μ, given covariance C.

The geodesic Mahalanobis distance relies on a regularized sample inverse-covariance operator [102]

$$C^{-1} := \sum_{q=1}^{Q} (1/\lambda_q) v_q \otimes v_q, \tag{2.67}$$

where λ_q is the q^{th} sorted eigenvalue of C, v_q is the corresponding eigenfunction, and $Q \leq \min(M, N)$ is a regularization parameter. Then, the corresponding square-root inverse-covariance operator is

$$C^{-1/2} := \sum_{q} (1/\sqrt{\lambda_q}) v_q \otimes v_q \tag{2.68}$$

and the geodesic Mahalanobis distance of the point ν from mean μ is

$$d_g(\nu, \mu; C) := (\langle C^{-1/2} t, C^{-1/2} t \rangle_{\mathcal{F}})^{0.5} \tag{2.69}$$

where

$$t := \mathrm{Log}_\mu(\nu).$$

Let Y be a random variable that generates the N independent and identically distributed data points $\{y_n\}_{n=1}^{N}$ as follows. For each n, we first draw a cluster number $l \in \{1, 2, \cdots, L\}$

with probability w_l (where $\forall l : w_l > 0$ and $\sum_l w_l = 1$) and then draw y_n from $P(Y|\mu_l, C_l)$. Thus, the probability of observing y_n is $P(y_n) = \sum_l w_l P(y_n|\mu_l, C_l)$.

The *parameters* for $P(Y)$ are $\theta = \{w_l, \mu_l, C_l\}_{l=1}^L$. We solve for the maximum-likelihood estimate of θ via EM. Let $\{S_n\}_{n=1}^N$ be *hidden* random variables that give, for each n, the cluster number $s_n \in \{1, \cdots, L\}$ that generated data point y_n.

2.8.2 Expectation-Maximization

EM performs iterative optimization. Each EM iteration involves an E-step and an M-step. At iteration i, given parameter estimates θ^i, the E-step defines a function $\mathcal{Q}(\theta|\theta^i) := E_{P(\{S_n\}_{n=1}^N|\{y_n\}_{n=1}^N, \theta^i)}[\log P(\{S_n, y_n\}_{n=1}^N|\theta)]$. For our mixture model,

$$\mathcal{Q}(\theta|\theta^i) = \sum_n \sum_l P(s_n = l|y_n, \theta^i) \left(\log w_l - 0.5 \log |C_l| - 0.5 d_g^2(\mu_l, y_n; C_l)\right) + \text{ constant}, \tag{2.70}$$

where

$$P(s_n = l|y_n, \theta^i) = \frac{P(s_n = l|\theta^i) P(y_n|s_n = l, \theta^i)}{P(y_n|\theta^i)} = \frac{w_l^i P(y_n|\mu_l^i, C_l^i)}{\sum_l w_l^i P(y_n|\mu_l^i, C_l^i)}. \tag{2.71}$$

We denote $P(s_n = l|y_n, \theta^i)$ in shorthand by the class membership P_{nl}^i. We denote $\sum_n P_{nl}^i$ in shorthand by P_l^i. Simplifying gives

$$\mathcal{Q}(\theta|\theta^i) = \sum_l P_l^i \left(\log w_l - 0.5 \log |C_l|\right) - 0.5 \sum_n \sum_l P_{nl}^i d_g^2(\mu_l, y_n; C_l) + \text{ constant}. \tag{2.72}$$

The M-step maximizes $\mathcal{Q}(\theta)$, under the constraints on w_l, using the method of Lagrange multipliers, to give the optimal values and, hence, the updates, for parameters θ.

Thus, the proposed clustering algorithm is as follows:

1) **Input:** A set of points $\{\Phi(x_n)\}_{n=1}^N$ on the unit Hilbert sphere in RKHS with all associated weights p_n set to unity.

2) Reduce the dimensionality of the input using the algorithm in Section 2.7 to give points $\{y_n\}_{n=1}^N$ on a lower dimensional subsphere of the Hilbert sphere in RKHS.

3) Initialize iteration count $i := 0$. Initialize parameters $\theta^0 = \{w_l^0, \mu_l^0, C_l^0\}_{l=1}^L$ as follows: run farthest-point clustering [103] (with kernel-based distances; with randomly selected first point) to initialize kernel k means [40] that, in turn, initializes μ_l^0 and C_l^0 to be the mean and covariances of cluster l, respectively, and w_l^0 equal to the number of points in cluster l divided by N.

4) Iteratively update the parameter estimates until convergence, as follows:

5) Evaluate probabilities $\{P^i_{nl}\}$ using current parameter estimates θ^i.

6) Update means $\mu_l^{i+1} = \arg\min_\mu \sum_n P^i_{nl} d_g^2(\mu_l, y_n; C_l)$ using a gradient-descent algorithm similar to that used in Section 2.6.1 for the sample-weighted Karcher mean.

7) Update covariances $C_l^{i+1} = \sum_n (P^i_{nl}/P^i_l) \mathrm{Log}_{\mu_l^{i+1}}(y_n) \otimes \mathrm{Log}_{\mu_l^{i+1}}(y_n)$.

8) Update probabilities $w_l^{i+1} = P^i_l / (\sum_l P^i_l)$.

9) **Output:** Parameters: $\theta = \{w_l, \mu_l, C_l\}_{l=1}^L$. Labeling: Assign $\Phi(x_n)$ to the cluster l that maximizes $P(y_n|\mu_l, C_l)$.

2.9 Experiments

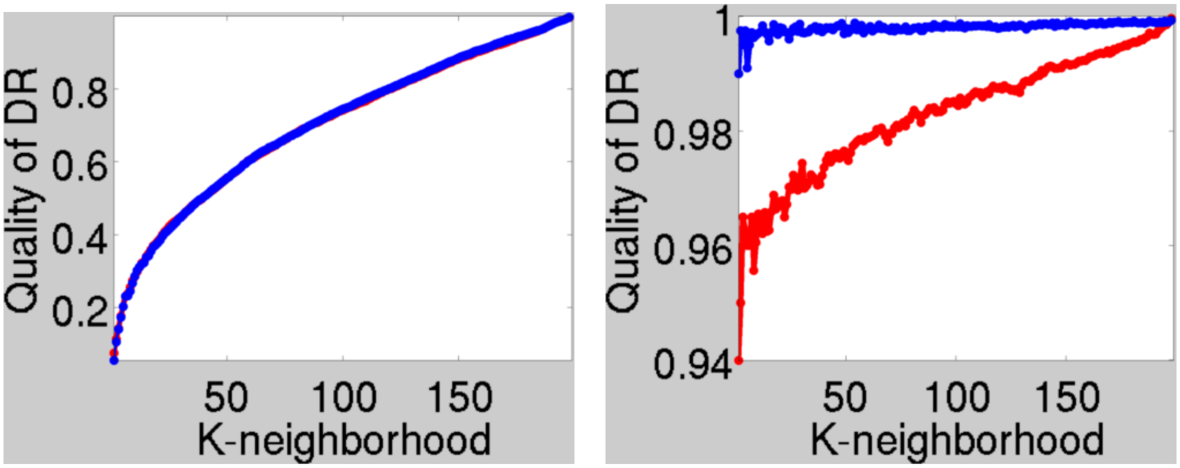
This section shows results on simulated data, real-world face images from the Olivetti Research Laboratory (ORL) [104], and real-world data from the University of California Irvine (UCI) machine learning repository [105].

2.9.1 Nonlinear Dimensionality Reduction

We employ kPCA and the proposed kPGA for nonlinear dimensionality reduction on simulated and real-world databases. To evaluate the quality of dimensionality reduction, we use the co-ranking matrix [106] to compare rankings of pairwise distances between 1) data points in the original high-dimensional space (i.e., without any dimensionality reduction) and 2) the projected data points in the lower dimensional embedding found by the algorithm. Based on this motivation, a standard measure to evaluate the quality of dimensionality-reduction algorithms is to average, over all data points, the fraction of other data points that remain inside a κ neighborhood defined based on the original distances [106]. For a fixed number of reduced dimensions, an ideal dimensionality-reduction algorithm would lead to this quality measure being 1 for every value of $\kappa \in \{1, 2, \cdots, N-1\}$, where N is the total number of points in the dataset.

2.9.1.1 Simulated Data – Points on a High-Dimensional Unit Hilbert Sphere

We generate $N = 200$ data points lying on the unit Hilbert sphere in $\mathbb{R}^{100}$. We ensure the intrinsic dimensionality of the dataset to be 2 by considering a subsphere $\mathbb{S}^2$ of dimension 2 and sampling points from a von Mises-Fisher distribution on $\mathbb{S}^2$ [99]. We set the kernel as $k(x, y) := \langle x, y \rangle$ that reduces the map $\Phi(\cdot)$ to identity (i.e., $\Phi(x) := x$) and, thereby, performs the analysis on the original data that lie on a Hilbert sphere in input space. Fig. 2.11 shows the results of the dimensionality reduction using kPCA and



Reduced Dimension $Q = 1$

Reduced Dimension $Q = 2$

Fig. 2.11. Nonlinear dimensionality reduction on simulated data: The performance for the proposed kPGA is in blue and that for the standard kPCA is in red. The horizontal axis shows values of κ in the κ neighborhood [106]. The quality measure on the vertical axis indicates the preservation of κ-sized neighborhoods based on distances in the original space (see text). For a fixed number of reduced dimensions Q, the ideal performance is a quality measure of 1 for all κ.

kPGA. When the reduced dimensionality is forced to be 1, which we know is suboptimal, both kPCA and kPGA perform comparably. However, when the reduced dimensionality is forced to be 2 (which equals the intrinsic dimension of the data), then kPGA clearly outperforms kPCA; kPGA preserves the distance-based κ neighborhoods for almost every value of $\kappa \in \{1, \cdots, 199\}$. The result in Fig. 2.11 is also consistent with the covariance eigenspectra produced by kPCA and kPGA. Standard kPCA, undesirably, gives three nonzero eigenvalues $(0.106, 0.0961, 0.0113)$ that reflect the dimensionality of the data representation for points on $\mathbb{S}^2$. On the other hand, the proposed kPGA gives only two nonzero eigenvalues $(0.1246, 0.1211)$ that reflect the intrinsic dimension of the data. Thus, kPGA needs fewer components/dimensions to represent the data.

2.9.1.2 Real-World Data – ORL Face Image Database

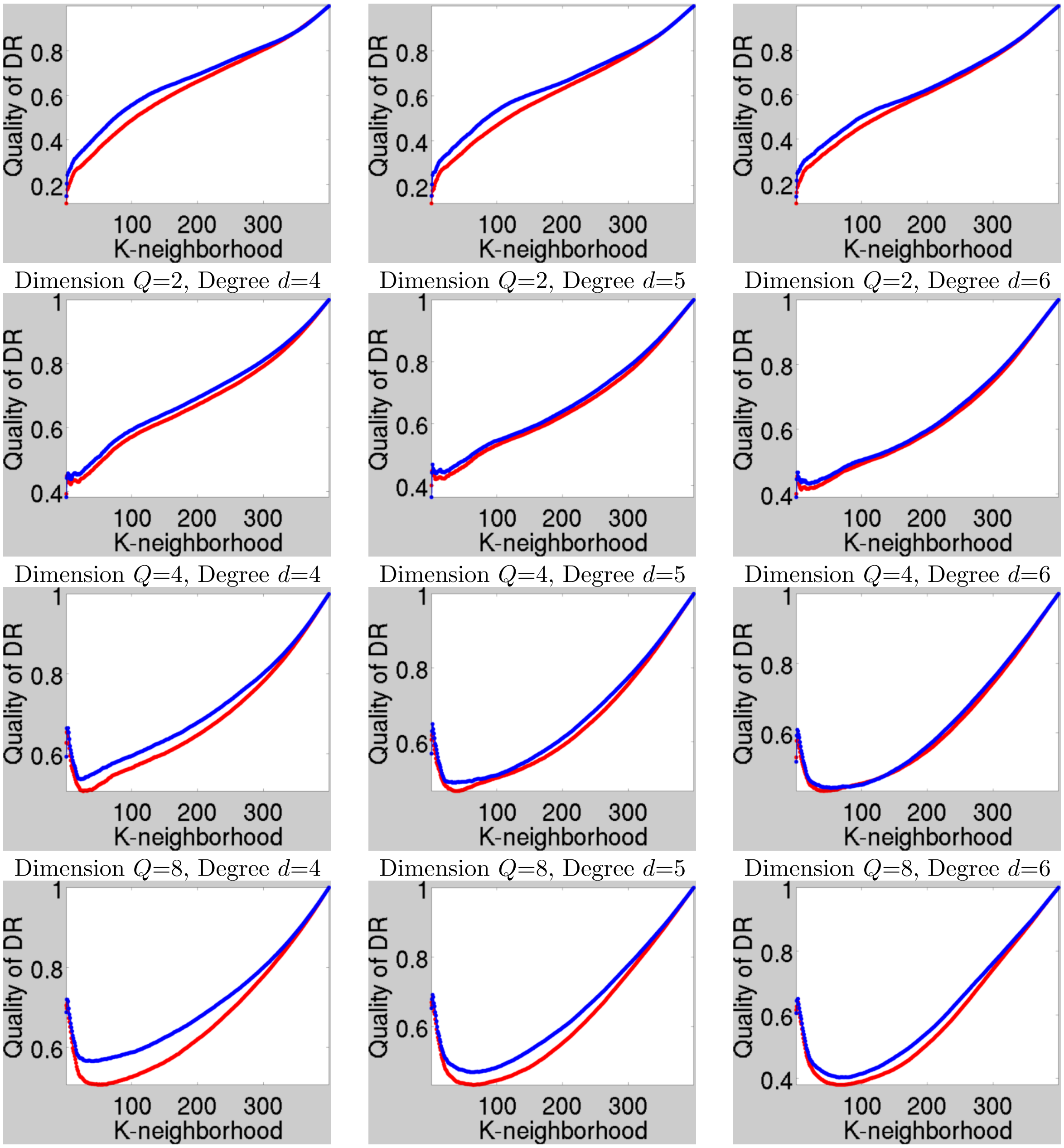
The ORL database [104] comprises $N = 400$ face images of size 112×92 pixels. To measure image similarity, a justifiable kernel is the polynomial kernel $k(x, y) := (\langle x, y \rangle)^d$ after normalizing the intensities in each image x (i.e., subtract mean and divide by standard deviation) so that $\langle x, x \rangle = 1 = k(x, x)$ [10]. Fig. 2.12 shows the results of nonlinear dimensionality reduction using standard kPCA and the proposed kPGA. For a range of values of the reduced dimension (i.e., $2, 4, 8, 16, 32, 64, 128, 256$) and a range of values of the polynomial kernel degree d (i.e., $d = 4, 5, 6$), the proposed kPGA outperforms standard kPCA with respect to the κ-neighborhood-based quality measure.

2.9.2 Clustering

We use the UCI repository to evaluate clustering in RKHS. Interestingly, for all but two of the UCI datasets used in this paper, the number of modes in kPCA (using the Gaussian kernel) capturing 90% of the spectrum energy ranges from 3–15 (mean 8.5, standard deviation 4.5). For only two datasets is the corresponding number of modes more than 20. This number is usually close to the intrinsic dimension of the data.

2.9.2.1 Real-World Data – UCI Machine Learning Repository

We evaluate clustering algorithms by measuring the error rate in the assignments of data points to clusters; we define error rate as the fraction of the total number of points in the dataset assigned to the incorrect cluster. We evaluate clustering error rates on a wide range of subspace dimensions $Q \in \{1, \cdots, 30\}$. For each Q, we repeat the following process 50 times: we randomly select 70% points from each cluster, run the clustering algorithm,



Dimension Q=2, Degree d=4

Dimension Q=2, Degree d=5

Dimension Q=2, Degree d=6

Dimension Q=4, Degree d=4

Dimension Q=4, Degree d=5

Dimension Q=4, Degree d=6

Dimension Q=8, Degree d=4

Dimension Q=8, Degree d=5

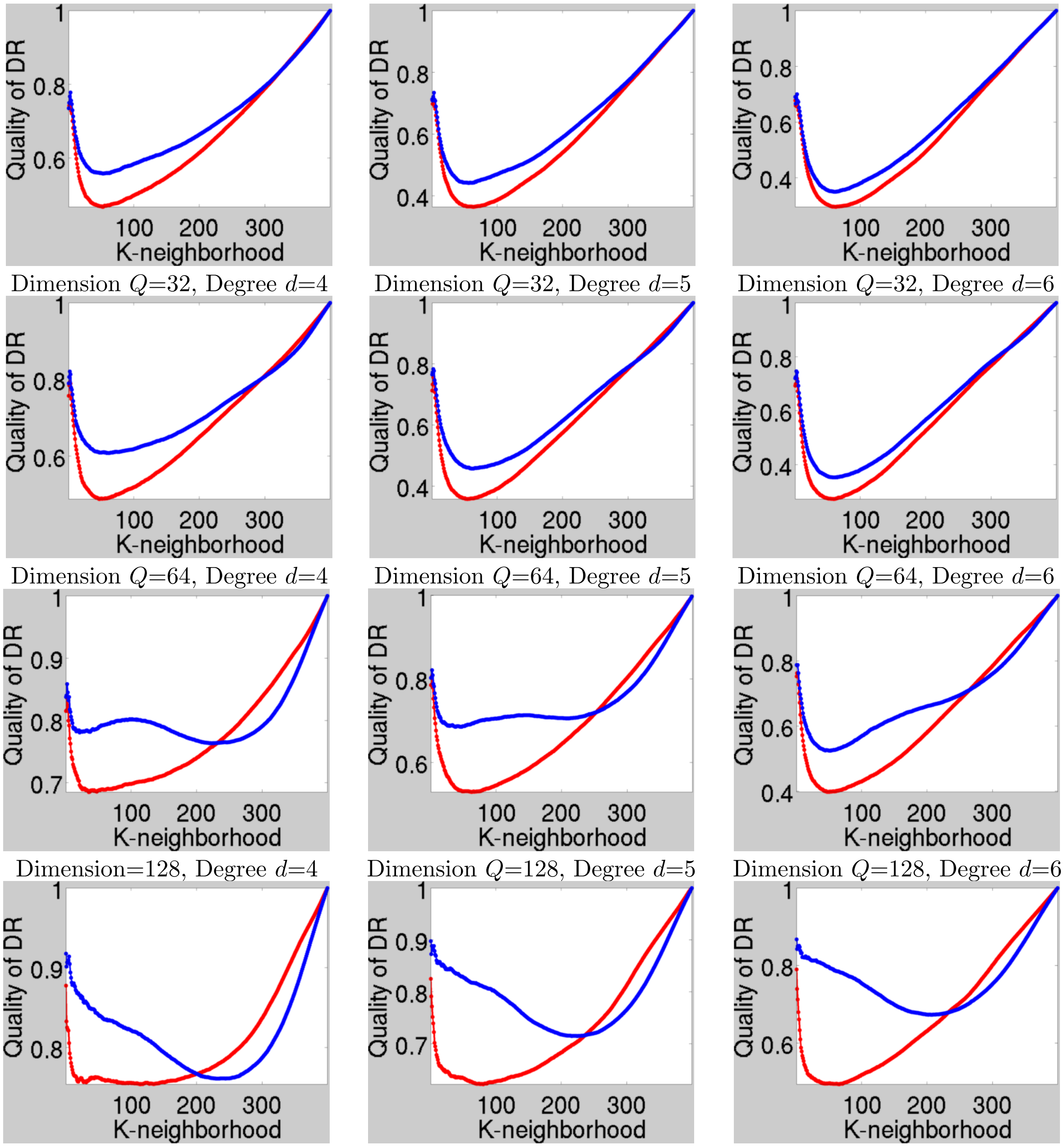
Dimension Q=8, Degree d=6

Dimension Q=16, Degree d=4

Dimension Q=16, Degree d=5

Dimension Q=16, Degree d=6

Fig. 2.12. Nonlinear dimensionality reduction on ORL face images: The **blue** curves represent the **proposed kPGA** and the **red** curves represent **standard kPCA**. Each subfigure plots quality measures (on vertical axis) for reduced-dimension values $Q = 2, 4, 8, 16$ and polynomial-kernel-parameter values $d = 4, 5, 6$. Within each subfigure (on horizontal axis), $\kappa = 1, \cdots, 399$. See next page for additional results with reduced-dimension values $Q = 32, 64, 128, 256$.



Dimension Q=32, Degree d=4

Dimension Q=32, Degree d=5

Dimension Q=32, Degree d=6

Dimension Q=64, Degree d=4

Dimension Q=64, Degree d=5

Dimension Q=64, Degree d=6

Dimension=128, Degree d=4

Dimension Q=128, Degree d=5

Dimension Q=128, Degree d=6

Dimension=256, Degree d=4

Dimension Q=256, Degree d=5

Dimension Q=256, Degree d=6

Fig. 2.12. Continued.

and compute the error rate. We use the Gaussian kernel $k(x_i, x_j) = \exp(-0.5\|x_i - x_j\|_2^2/\sigma^2)$ and set σ^2, as per convention, to the average squared distance between all pairs (x_i, x_j).

From Fig. 2.13 to Fig. 2.20, we compare the performance of spectral clustering [107], standard kPCA, and the proposed kPGA. Fig. 2.13 shows the result of wine, where kPGA shows the lowest error rates (over all Q) and outperforms spectral clustering. Fig. 2.14 shows the result of Harberman, where kPGA has lower error rates than both kPCA and spectral clustering in most Q. From Fig. 2.15 to Fig. 2.18 (iris, vote, heart, and ecoli in order), the kPGA performs better or as well as for almost all choices of Q, but the kPGA

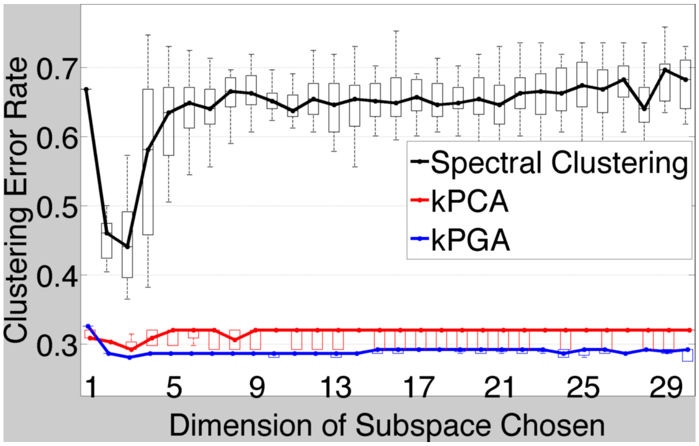


Fig. 2.13. **Clustering result of wine:** The blue line indicates the kPGA, the red line indicates kPCA, and the black line is spectral clustering. The x-axis is subspace dimensions $Q \in \{1, \cdots, 30\}$. For each Q, we repeat the following process 50 times: we randomly select 70% points from each cluster, run the clustering algorithm, and compute the error rate (y-axis). The kPGA has the lowest error rates (over all Q) of all methods.

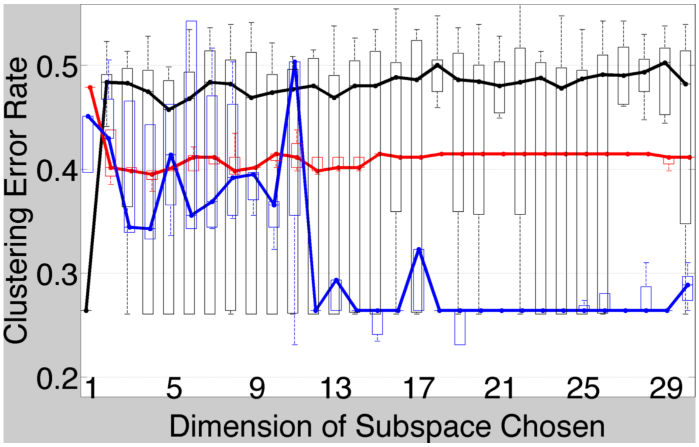


Fig. 2.14. **Clustering result of Harberman:** The kPGA has lower error rates than kPCA and spectra clustering in most of Q except for $Q = 1, 2, 5$, and 11.

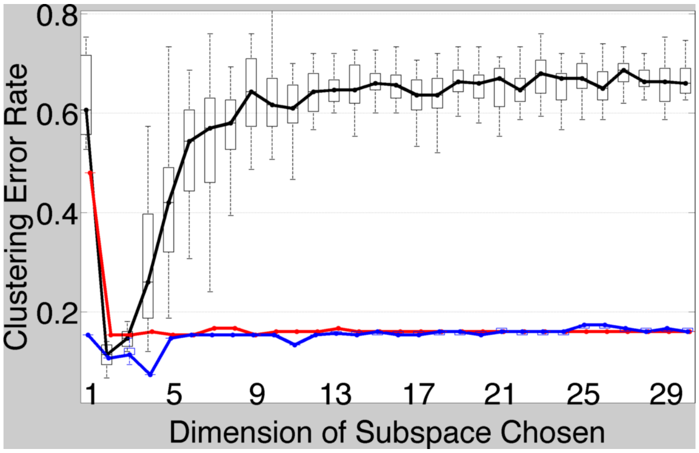


Fig. 2.15. **Clustering result of iris:** The performance of kPGA is less than or equal to kPCA for all Q, and both kPGA and kPCA outperform the spectral clustering for all Q. The kPGA has the best error rate when $Q = 4$.

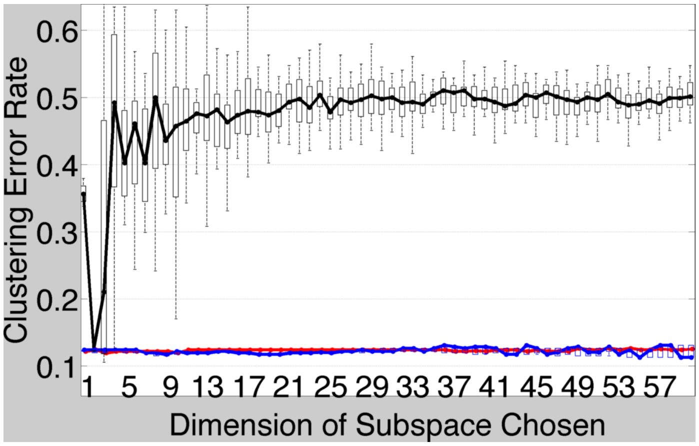


Fig. 2.16. **Clustering result of vote:** The performance of kPGA is less than or equal to kPCA for all Q, and both kPGA and kPCA outperform the spectral clustering for all Q. The kPGA has the best error rate when $Q = 55$.

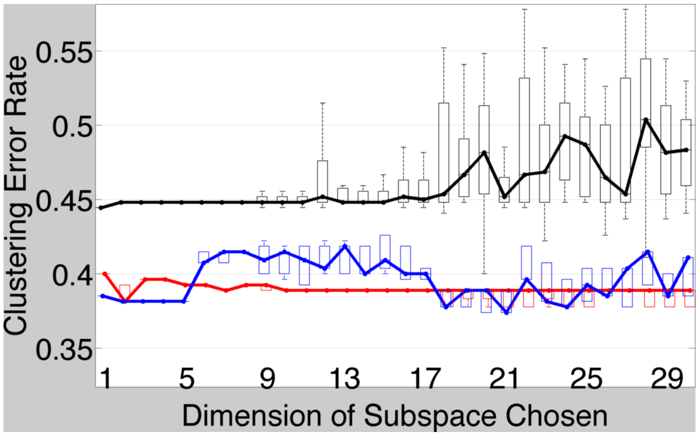


Fig. 2.17. **Clustering result of heart:** The performance of kPGA is less than or equal to kPCA over all Q, and both kPGA and kPCA outperform the spectral clustering for all Q. The kPGA has the best error rate when $Q = 21$.

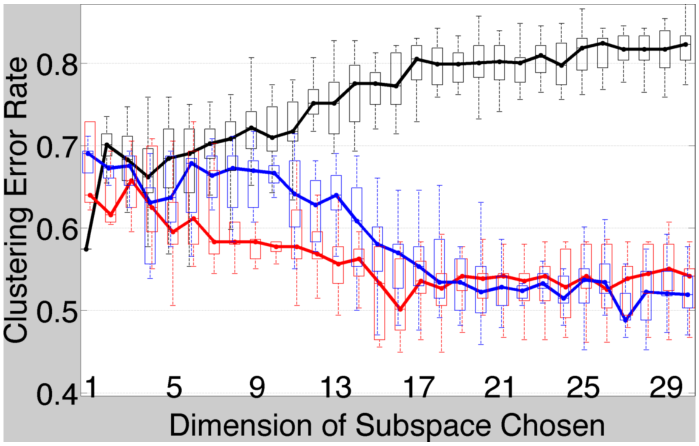


Fig. 2.18. **Clustering result of ecoli:** The performance of kPGA is less than or equal to kPCA over all Q, and both kPGA and kPCA outperform the spectral clustering for all Q. The kPGA has the best error rate when $Q = 27$.

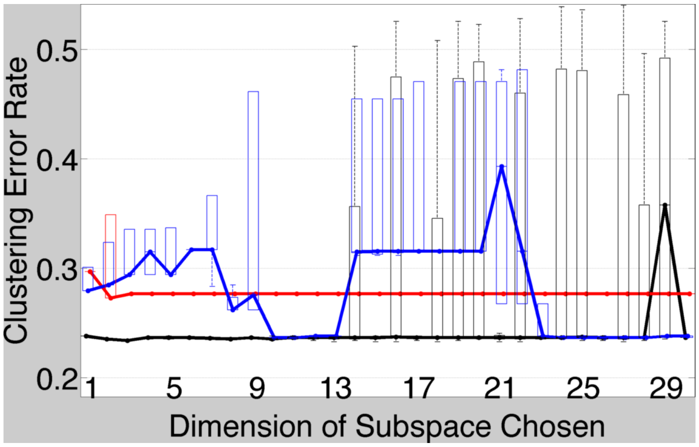


Fig. 2.19. **Clustering result of blood:** The kPGA performs as well as spectral clustering and has better performance than the kPCA when $8 < Q \leq 13$ and $Q \geq 23$.

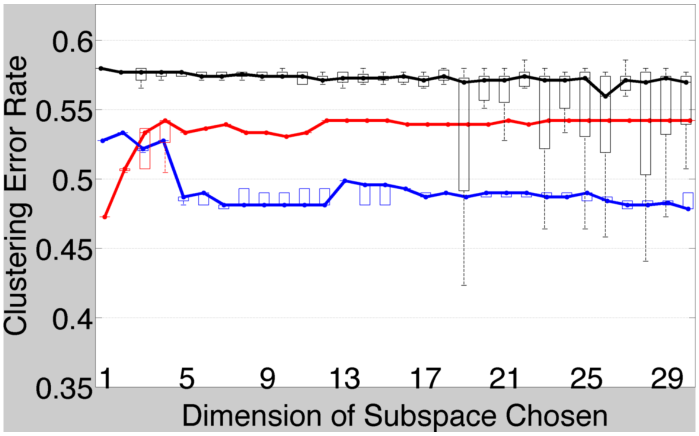


Fig. 2.20. **Clustering result of liver:** The performance of kPGA is better than both the kPCA and spectral clustering when $Q \geq 2$.

gives the lowest error rates (over all Q) and outperforms spectral clustering. In Fig. 2.19 is the blood dataset for which the kPGA performs as well as spectral clustering (over all Q). In Fig. 2.20 is the liver dataset in which kPCA has the best performance when $Q = 1$; however, kPGA performs the best whenever $Q > 2$.

2.10 Conclusion

This dissertation addresses the hyperspherical geometry of points in kernel feature space, which naturally arises from many popular kernels and kernel normalization. This dissertation proposes kPGA to perform PGA on the Hilbert sphere manifold in RKHS, through algorithms for computing the sample-weighted Karcher mean and the eigenvalues and eigenfunctions of the sample-weighted Karcher covariance. It leverages kPGA to propose methods for 1) nonlinear dimensionality reduction and 2) clustering using mixture-model fitting on the Hilbert sphere in RKHS.

CHAPTER 3

HIERARCHICAL BAYESIAN MODELING, ESTIMATION, AND SAMPLING FOR MULTIGROUP SHAPE ANALYSIS

3.1 Abstract

This dissertation proposes a novel method for the analysis of anatomical shapes present in biomedical image data. Motivated by the natural organization of population data into multiple groups, this dissertation presents a novel *hierarchical generative* statistical model on shapes. The proposed method represents shapes using pointsets and defines a joint distribution on the population's 1) shape variables and 2) object-boundary data. The proposed method solves for optimal 1) point locations, 2) correspondences, and 3) model-parameter values as a *single* optimization problem. The optimization uses expectation-maximization relying on a novel Markov-chain Monte-Carlo algorithm for *sampling* in Kendall shape space. Results on clinical brain images demonstrate advantages over the state-of-the-art.

3.2 Introduction and Related Work

Shape analysis [32], [33] entails the inference of shape models from population data and associated statistical analyses, e.g., hypothesis testing for comparing groups. The natural organization of biomedical data into groups, and possibly subgroups, calls for a *hierarchical* modeling strategy. Previous works on hierarchical shape modeling typically concern 1) multiresolution models [108], e.g., a face model at fine-to-coarse resolutions, or 2) multipart models [109], e.g., a car decomposed into body, tires, and trunk. In contrast,

This chapter is a revised and adapted with permission (copyright by authors) from

Hierarchical Bayesian Modeling, Estimation, and Sampling for Multigroup Shape Analysis. *Yen-Yun Yu, P. Thomas Fletcher, and Suyash P. Awate.* Medical Image Computing and Computer-Assisted Intervention - MICCAI 2014 Volume 8675 of the series Lecture Notes in Computer Science pages 9-16 (17th International Conference on the Medical Image Computing and Computer-Assisted Intervention, Boston, MA, USA, September 14-18, 2014) the proposed framework deals with population data comprising multiple groups, e.g., the Alzheimer's disease (AD) population comprising people with 1) dementia due to AD, 2) mild cognitive impairment due to AD, and 3) preclinical AD.

Fig. 3.1 outlines the proposed *generative* model, where 1) top-level variables capture the shape properties across the population (e.g., all individuals with and without medical conditions), 2) variables at a level below capture the shape distribution in different groups within the population (e.g., clinical cohorts based on gender or type of disease within a spectrum disorder), and 3) variables at the next lower level capture individual shapes, which finally relate to 4) individual image data at the lowest level. Moreover, the top-level population variables provide a common reference frame for the group shape models, which is necessary to enable comparison between the groups.

This dissertation makes several contributions. 1) It proposes a novel hierarchical generative model for population shape data. It represents a shape as an equivalence class of pointsets modulo translation, rotation, and isotropic scaling [32]. This model tightly couples each individual's shape (unknown) to the observed image data by designing its joint probability density function (PDF) using current distance or kernel distance [110], [111]. The current distance makes the logarithm of the joint PDF a nonlinear function of the point locations. Subsequently, the proposed method solves a *single unified model-fitting optimization problem to estimate optimal point locations, correspondences, and parameter values.* 2) The proposed model fitting relies on expectation-maximization (EM), treating the individual-shape and group-shape variables as hidden random variables, thereby integrating them out while estimating parameters (e.g., the population shape mean and covariance). In this way, the proposed EM algorithm improves over typical methods that use mode approximation for shape variables. 3) The EM algorithm entails evaluating an expectation over the posterior PDF of the shape variables. For instance, the posterior PDF for individual-shape variables involves the a) likelihood PDF designed using the current distance and b) prior PDF conditioned on the group shape model. To compute the expectation, the proposed EM algorithm relies on a novel adaptation of Hamiltonian Monte Carlo (HMC) [112] *sampling in Kendall shape space.* 4) The results show that the hierarchical model leads to more compact model fits and improved detection of subtle shape variations between groups.

Early approaches [32], [113] to statistical shape modeling rely on manually defined homologous landmarks. Later approaches optimize point positions or correspondences using statistical compactness criteria such as the 1) logarithm of the determinant of the model covariance matrix [114], 2) minimum description length [115], [116], or 3) minimum

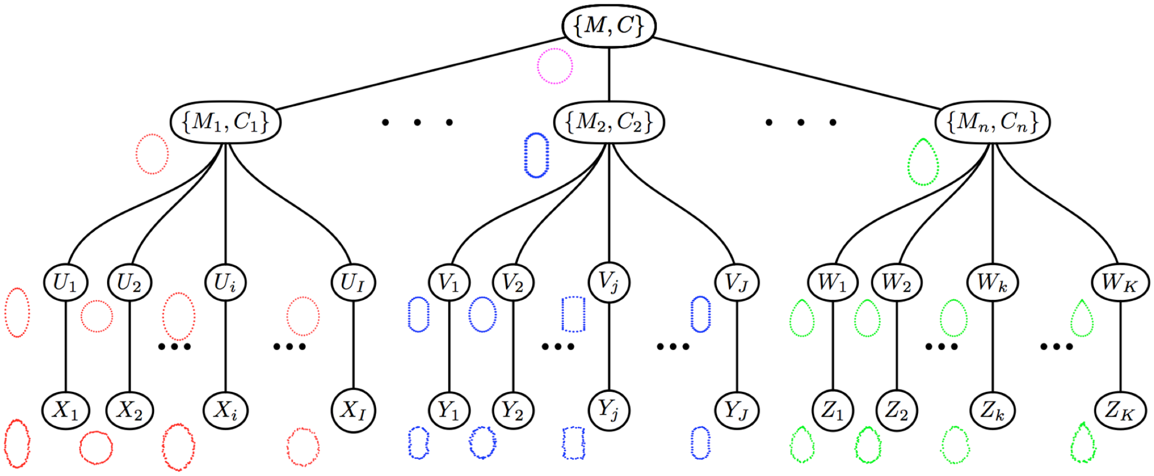


Fig. 3.1. **Proposed hierarchical generative statistical model for multigroup shape data:** The model variables at the top level capture statistical properties of the population, the variables at a lower level capture statistical properties of different groups within a population, and the variables at the lowest level capture individual properties.

entropy [117]. However, these approaches 1) do *not* incorporate a generative statistical model, 2) introduce ad hoc terms in the objective function to obtain correspondences, and 3) do *not* estimate shape-model parameters within the aforementioned optimization. Some generative models for shape analysis do exist [35]–[38], but these models rely on a predetermined template shape with manually placed landmarks.

Experiments in this chapter use the fitted hierarchical model to perform 1) hypothesis testing for comparison between pairs of groups using permutation testing and 2) classification for image retrieval.

3.3 Background

Shape is emerging as an important feature to describe the geometry of objects. The definition of shape is *all the geometrical information that remains when location, scale, and rotational effects are filtered out from an object* [33]. In other words, two objects have the same shape if they are invariant in Euclidean similarity transformations (Fig. 3.2). *Landmarks* are homologous points that lie on the surface of the anatomy. One of the standard ways to define shape is *landmark-based shape*, in which a finite number of points (i.e., landmarks) are located on the surface of the object. Each point of correspondence on each object is a match between and within a population of shapes. Landmarks can be generated by different approaches, such as selection by a domain expert manually or by following the geometrical property; thus, it is known as the *point location* problem [118], [119]. Next, finding a meaningful correspondence within a shape population is a prerequisite for shape analysis, which is known as the *shape correspondence* problem. Suppose a shape X has N landmarks where each landmark is denoted as $X(n) \in \mathbb{R}^D$, then the shape is represented as $X := (X(1), X(2), \ldots, X(N))^T$ in $\mathbb{R}^{ND}$.

3.4 Hierarchical Generative Shape Model

This section introduces the proposed hierarchical Bayesian shape model for statistical shape analysis. We first describe the proposed hierarchical model for multigroup shape data.

3.4.1 Observed Data

Consider a group of I vector random variables $X := \{X_i\}_{i=1}^I$, where X_i is a vector random variable denoting a given set of points on the boundary of an anatomical structure in the i-th individual's image data. That is, $X_i := \{X_i(n)\}_{n=1}^{N_i}$ where $X_i(n) \in \mathbb{R}^D$ is the D-dimensional spatial coordinate of the n-th point in the pointset. Such points can be

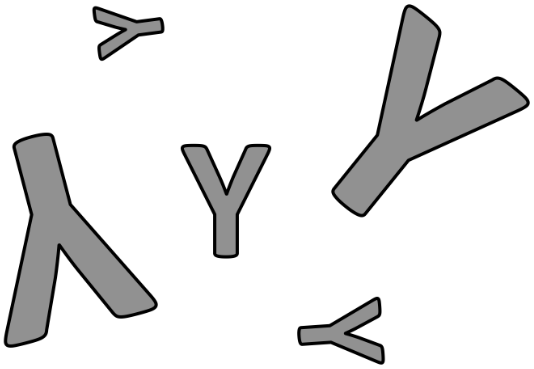


Fig. 3.2. **Example of the shapes:** Five objects have the same shape, but under different location, scale, and rotational effects.

obtained from a given segmentation or delineation of the anatomical structure. In this chapter, $D = 3$. In any individual's image data, the number of boundary points N_i can be arbitrary. Similarly, consider other groups of data, e.g., data $Y := \{Y_j\}_{j=1}^{J}$ derived from a group of J individuals, data $\{Z_k\}_{k=1}^{K}$, etc.

3.4.2 Individual Shape Variable

For the first group (corresponding to data X), consider a group of I *latent/hidden* random variables $U := \{U_i\}_{i=1}^{I}$, where U_i is a vector random variable representing the shape of the anatomical structure of the i-th individual. That is, $U_i := \{U_i(t)\}_{t=1}^{T}$, where $U_i(t) \in \mathbb{R}^D$ is the D-dimensional coordinate of the t-th point in the shape representation of the i-th individual's structure. We assume the observations X_i to be derived from the individual shape U_i. Similarly, we consider latent random variables, i.e., V, W, etc., representing shapes for the other groups. To enable intragroup and intergroup statistical analysis, we ensure that all shape models lie in the same space by enforcing the same number of points T in all shape models.

3.4.3 Group Shape Variable

Consider the first group of shapes U to be derived from a shape probability density function having a mean shape M_1 and a shape covariance C_1. Consider other groups of shapes modeled analogously, i.e., V derived from a group with shape mean and covariance (M_2, C_2), W derived from a group with shape mean and covariance (M_n, C_n), etc. This chapter treats the group means, i.e., M_1, M_2,$\cdots$,M_n, as latent random variables and the group covariances, i.e., C_1, C_2,$\cdots$,C_n, as parameters. The proposed method can be generalized to treat the group covariances as random variables.

3.4.4 Population Shape Variable

Consider all group shape means, i.e., M_1, M_2,$\cdots$,M_n, to be derived from a single population of shapes with mean M and covariance C. In this chapter, without loss of generality, we consider only two groups ($n = 2$) for simplicity.

3.5 Joint Probability Density Function (PDF)

We model the joint PDF with 1) parameters M, C, C_1, C_2, 2) group shape variables M_1, M_2, 3) individual shape variables U, V, and 4) data X, Y as

$$P(M_1, M_2, U, V, X, Y | M, C, C_1, C_2) := \tag{3.1}$$

$$P(M_1|M, C)P(M_2|M, C)\Pi_{i=1}^{I} P(U_i|M_1, C_1)P(X_i|U_i)\Pi_{j=1}^{J} P(V_j|M_2, C_2)P(Y_j|V_j). \tag{3.2}$$

3.5.1 PDF of Observed Data Given Individual Shape Variable

We model $P(X_i|U_i)$, $P(Y_j|V_j)$ using current distance. As in Fig. 3.3, between pointsets $A := \{a_i\}_{i=1}^{I}$ and $B := \{b_j\}_{j=1}^{J}$, the squared current distance is

$$d_K^2(A, B) := \sum_{i=1}^{I}\sum_{i'=1}^{I} K(a_i, a_{i'}) + \sum_{j=1}^{J}\sum_{j'=1}^{J} K(b_j, b_{j'}) - 2\sum_{i=1}^{I}\sum_{j=1}^{J} K(a_i, b_j) \tag{3.3}$$

where $K(\cdot, \cdot)$ is a Mercer kernel. In this chapter, $K(\cdot, \cdot)$ is the Gaussian kernel with isotropic covariance $\sigma^2\mathbf{I_D}$. We use the current distance to define

$$P(X_i|U_i) := (1/\gamma) \exp\left(-d_K^2(X_i, U_i)\right), \tag{3.4}$$

over finite support, where γ is the normalization constant. The current-distance model allows the number of points in the shape models U_i to be different from the number of boundary points in the data X_i.

3.5.2 PDF of Group Shape Variable

We model $P(U_i|M_1, C_1)$ as Gaussian with mean M_1 and covariance C_1 and $P(V_j|M_2, C_2)$ as Gaussian with mean M_2 and covariance C_2.

3.5.3 PDF of Group Shape Variables Given Population Parameters

We model $P(M_1|M, C)$ and $P(M_2|M, C)$ as Gaussian with mean M and covariance C; we choose the Gaussian 1) to be maximally noncommittal during model design and 2) as the conjugate prior for the Gaussian means M_1, M_2. Under the Gaussian model, strange-looking shapes can be avoided by preventing overregularization of the covariance estimate and preventing very large deviations from the mean (which are rare events under

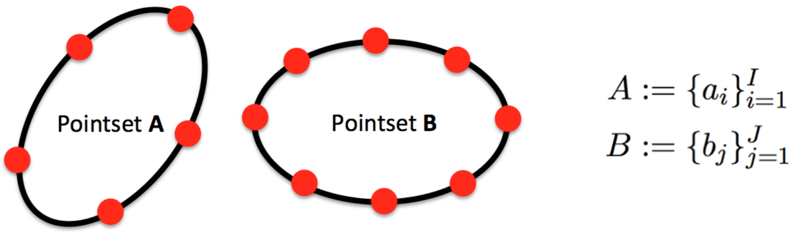


Fig. 3.3. **Example of current distance**: The current distance model allows the number of points in pointset A to be different from the number of points in pointset B.

the Gaussian). More importantly, the hierarchical model alleviates this issue by producing covariance estimates that are more compact and restrict variation over fewer modes.

3.6 Monte-Carlo Expectation-Maximization

This section presents the EM algorithm for the model-fitting optimization problem. The proposed model uses Monte-Carlo EM to fit the shape model to data. The parameters in our model are 1) the population mean M and covariance C and 2) the group covariances C_1, C_2. Denoting $\theta := \{M, C, C_1, C_2\}$, the optimal model fit is

$$\arg\max_{\theta} P(x, y|\theta) = \arg\max_{\theta} \int P(u, v, m_1, m_2, x, y|\theta) du dv dm_1 dm_2. \tag{3.5}$$

3.6.1 E-Step: Hamiltonian Monte Carlo (HMC)

In the i-th iteration, with parameter estimate $\widehat{\theta^i}$, the E-step constructs the Q function as

$$Q(\theta|\widehat{\theta^i}) := E_{P(U,V,M_1,M_2|x,y,\widehat{\theta^i})} \log P(U, V, M_1, M_2, x, y|\theta). \tag{3.6}$$

Because of the analytical intractability of this expectation, we approximate

$$Q(\theta|\widehat{\theta^i}) \doteq \widehat{Q}(\theta|\widehat{\theta^i}) := \sum_{s=1}^{S} (1/S) \log P(u^s, v^s, m_1^s, m_2^s, x, y|\theta) \tag{3.7}$$

using Monte-Carlo simulation. To sample the set of individual shapes u^s, v^s and the group-mean shapes m_1^s, m_2^s from $P(U, V, M_1, M_2|x, y, \widehat{\theta^i})$, we propose Gibbs sampling coupled with a novel adaptation of the HMC sampler [112]. Before describing the adapted HMC sampler, we outline the proposed shape-sampling algorithm for generating a sample of size S:

1) Set the sample index variable s to 0. Initialize the sampling algorithm with the sample point $s = 0$ denoted by $u^0 := \{u_i^0\}_{i=1}^{I}, v^0 := \{v_j^0\}_{j=1}^{J}, m_1^0, m_2^0$.

   Given sample point s, sample the $(s+1)$-th sample point as follows:

2) Initialized with u_i^s, $\forall i$ sample $u_i^{s+1} \sim P(U_i | v^s, m_1^s, m_2^s, x, y, \widehat{\theta^i})$.

3) Initialized with v_j^s, $\forall j$ sample $v_j^{s+1} \sim P(V_j | u^{s+1}, m_1^s, m_2^s, x, y, \widehat{\theta^i})$.

4) Initialized with m_1^s, sample $m_1^{s+1} \sim P(M_1 | u^{s+1}, v^{s+1}, m_2^s, x, y, \widehat{\theta^i})$.

5) Initialized with m_2^s, sample $m_2^{s+1} \sim P(M_2 | u^{s+1}, v^{s+1}, m_1^{s+1}, x, y, \widehat{\theta^i})$.

6) If $s+1 = S$, then stop; otherwise increment s by 1 and repeat the previous four steps.

We ensure the independence of samples between Gibbs iteration s and the next $s+1$ by running the HMC algorithm sufficiently long and discarding the first few samples s.

HMC is a Markov-chain Monte-Carlo sampling algorithm. HMC exploits the gradient of the log PDF for fast exploration of the space of the random variables. The HMC approach first augments the original random variables with auxiliary momentum variables, then defines a Hamiltonian function combining the original and auxiliary variables, and, subsequently, alternates between simple updates for the auxiliary variables and Metropolis updates for the original variables. HMC proposes new states by computing a trajectory according to the Hamiltonian dynamics implemented with a leapfrog method and guarantees the new proposal states to be accepted with high probability. In our case, HMC requires gradients of $\log P(U, V, M_1, M_2 | x, y, \widehat{\theta^i})$ with respect to the latent variables $\{U_i\}_{i=1}^I, \{V_j\}_{j=1}^J, M_1, M_2$.

3.6.2 E-Step: Sampling in Shape Space

Using HMC naively leads to pointset updates that can change the location, scale, and pose of the pointset, thereby making the sampler very inefficient. For this problem, we propose to modify HMC by replacing the gradient of the log posterior by a *projected gradient* that restricts the updated shape to Kendall shape space. As shown in Fig. 3.4, starting with pointset u_i, the log-posterior gradient r^1 is first projected onto the preshape space to produce r^4, which has the same centroid and scale as u_i. Then, to remove rotation effects, the resulting preshape r^4 is rotationally aligned with the u_i, yielding r^5 (not shown in the figure). These steps project the log-posterior gradient at u_i, within HMC, to generate an updated shape r^5 as part of the trajectory within HMC.

3.6.3 M-Step: Parameters Estimation

In iteration i of the EM optimization, the M step maximizes $\widehat{Q}(\theta | \widehat{\theta^i})$ over θ and sets $\widehat{\theta}^{i+1} \leftarrow \arg\max_\theta \sum_{s=1}^S \log P(u^s, v^s, m_1^s, m_2^s, x, y | \theta)$. Subsequently, we get optimal values in closed form for the parameters $\hat{C}_1^{i+1}, \hat{C}_2^{i+1}, \hat{M}^{i+1}, \hat{C}^{i+1}$:

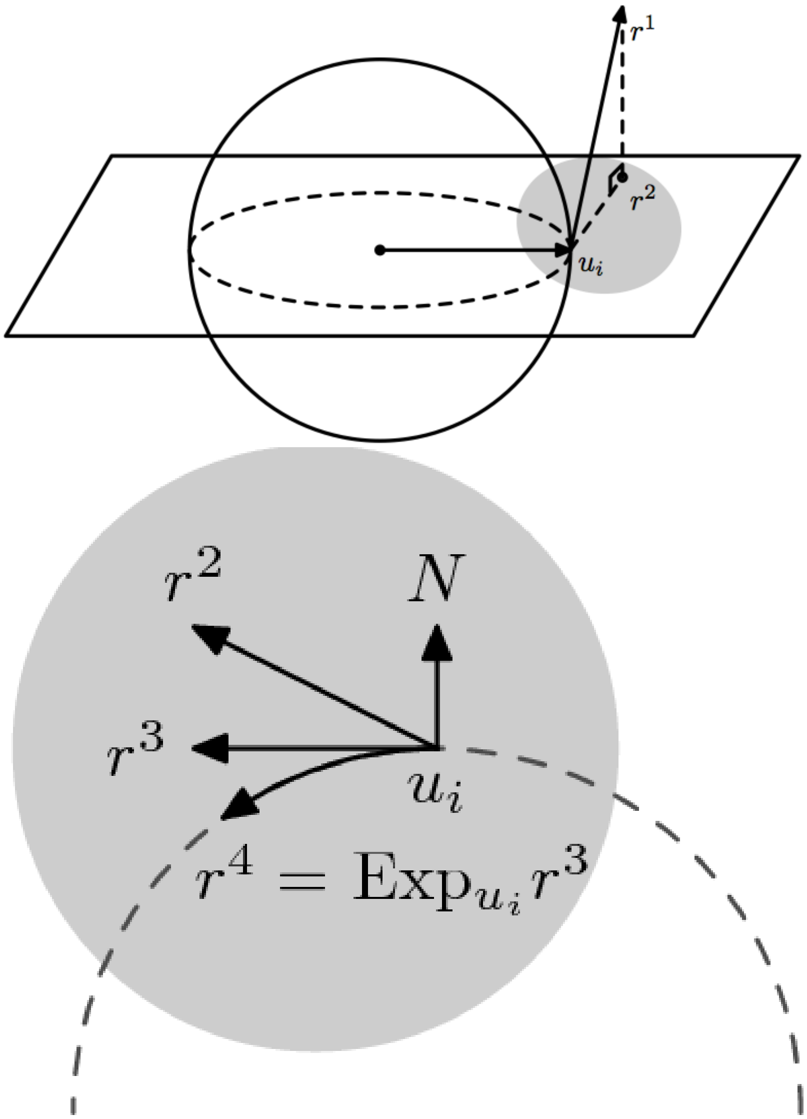


Fig. 3.4. **Illustration of projected gradient that restricts the updated shape to Kendall shape space: Top:** Kendall preshape space [33] (dotted hypersphere) that is the intersection of the (bold) hypersphere of fixed radius ρ (i.e., $\sum_t \|u_i(t)\|^2_{\mathcal{F}} = \rho^2$; fixes scale) and the hyperplane through the origin (i.e., $\sum_t u_i(t) = \mathbf{0}$; fixes translation). For a pointset u_i, log-posterior gradients r^1 are projected onto the hyperplane to produce r^2, which eliminates translation. **Bottom:** To remove changes in scale, the resulting projection r^2 is then projected onto the tangent space at u_i, tangent to the preshape space, and the resulting tangent-space projection r^3 is mapped to the preshape space via the manifold exponential map to give r^4. The text describes the last part of the projection.

$$\hat{C}_1^{i+1} = \frac{1}{SI}\sum_{s=1}^{S}\sum_{i=1}^{I}\big((u_i^s - m_1^s)(u_i^s - m_1^s)^T\big) \tag{3.8}$$

$$\hat{C}_2^{i+1} = \frac{1}{SJ}\sum_{s=1}^{S}\sum_{j=1}^{J}\big((v_j^s - m_2^s)(v_j^s - m_2^s)^T\big) \tag{3.9}$$

$$\hat{M}^{i+1} = \frac{1}{2S}\sum_{s=1}^{S}(m_1^s + m_2^s) \tag{3.10}$$

$$\hat{C}^{i+1} = \frac{1}{2S}\sum_{s=1}^{S}\big((m_1^s - M)(m_1^s - M)^T + (m_2^s - M)(m_2^s - M)^T\big). \tag{3.11}$$

3.7 Experiments

This section shows results for simulated and real data, including two dimensions (2D) and three dimensions (3D). We demonstrate the performance of the proposed model on the correspondence problem, classification, and hypothesis testing.

3.7.1 Correspondence Problem

First, we validate the hierarchical shape model for the correspondence problem on simulated bump shapes. This subsection provides a 2D example of a standard simulated test dataset [114] and shows that the proposed framework is able to correctly learn the true group and population models. Fig. 3.5 (top row) shows simulated (ground-truth) pointsets, which are very similar to those used in [114], in which the population is a collection of *box-bump* [114] shapes where the location of the bump (and each point) exhibits linear variation, across the group, over the top of the box. Whereas the desired population mean M corresponds to a shape with the bump exactly in the middle, the bumps in the true group means M_1, M_2 are located symmetrically on either side of the middle. The true covariance matrices for the groups (C_1, C_2) and the population (C) have a single nonzero eigenvalue. Fig. 3.5 (bottom row) shows the observed corrupted data, i.e., $\{x_i\}_{i=1}^4, \{y_j\}_{j=1}^4$, where we induce poor correspondences and noise in the point locations.

Fig. 3.6 shows the means for the groups and population after EM optimization. We see that the estimated population mean M and the expected values of the group means M_1, M_2 after EM optimization are close to their true values. Fig. 3.6 also shows that the correspondence of corrupted data points $x_i(t)$, across the group $(i = 1, \ldots, 4)$, is poor, indicating a large variance. On the other hand, after EM optimization, the correspondence of the expected values of the shape variables $U_i(t)$ indicates a significantly lower variance and correctly shows the linear variation in the point location across the group.

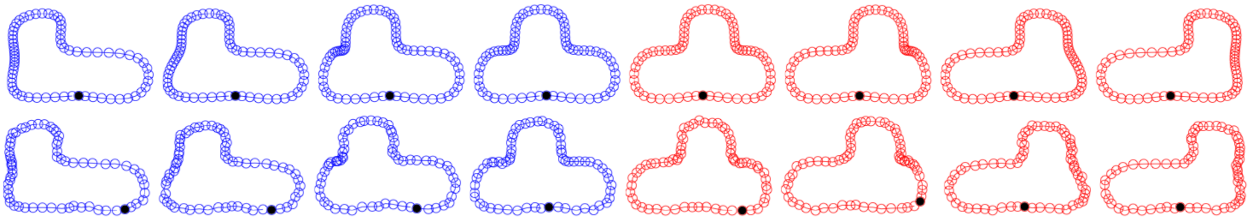


Fig. 3.5. **Simulated box-bump shapes** [114]. **Top Row:** Simulated ground-truth shape models for the two groups, each with four shapes. The bump for the first group (blue) is on the left and the bump for the second group (red) is on the right. Each shape has 64 points, shown by circles. The black filled circle indicates the first point in the list; other points are numbered counterclockwise. **Bottom Row:** Corrupted data where the point ordering in each shape is randomly circularly shifted (to induce poor correspondences), and independent Gaussian noise is added to each point position (to mimic errors in boundary detection).

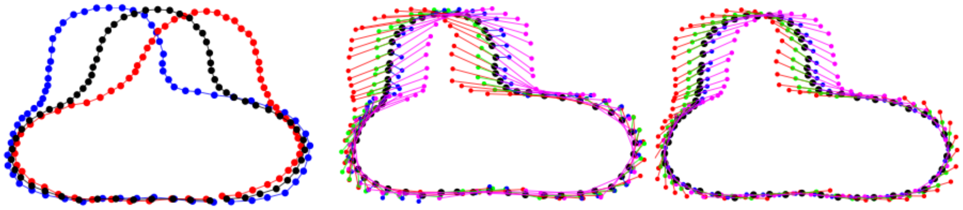


Fig. 3.6. **Shape correspondence on simulated box-bump shapes: Left:** The optimal population mean M (black dots) along with the expected values for the group means M_1 (blue dots) and M_2 (red dots) after EM inference. **Middle:** The **correspondence** of points for the corrupted data x_i across a selected group (blue shapes). **Right:** The correspondences for the expected values of shape models U_i after EM inference.

3.7.2 Classification

This section shows results of the proposed hierarchical multigroup shape modeling used for classification on a real dataset, namely the Tree Leaf Database [120] comprising images of leaves of 90 wood species growing in the Czech Republic. We used two of the largest available groups, comprising the species 1) *Carpinus betulus*, with 23 leaf images (Fig. 3.7; blue group) and 2) *Fagus sylvatica*, also with 23 leaf images (Fig. 3.7; red group). The leaf stem was removed from the images manually. Interestingly, the blue group has oblong leaves that have high curvature at one end and low at the other, whereas the red group has oblong leaves that are more symmetric with similar curvatures at both ends.

After the multigroup model is fit to *training* data, we can classify unseen shapes as follows. We evaluate the probability of the *test* pointset z being drawn from each group model, i.e., $P(z|M_1, C_1)$ and $P(z|M_2, C_2)$, and classify z to the class that yields a higher probability. We can evaluate the aforementioned probabilities as follows:

$$\begin{aligned} P(z|M_1, C_1) &= \int_w P(z, w|M_1, C_1) dw \\ &= \int_w P(z|w, M_1, C_1) P(w|M_1, C_1) dw \\ &\approx \sum_{s=1}^{S} \frac{1}{S} P(z|w^s) \end{aligned} \tag{3.12}$$

where

$$w^s \sim P(W|M_1, C_1). \tag{3.13}$$

w is the latent random variable corresponding to the *test* pointset z. We performed the classification task by training using only three leaves from each group and testing on the remaining 20 leaves in each group. We obtained a correct-classification rate of 97.5%.

3.7.3 Hypothesis Testing

In the following sections, we show the proposed hierarchical shape model for hypothesis testing on 3D shapes.

3.7.3.1 Initialization

For the 3D data, we assume the input 3D images undergoing shape analysis to be binary or soft masks, having intensities in the range $[-1, 1]$, that segment the image into the object of interest and the background. For hypothesis testing, it is less interesting to compare the performances of methods when the two groups are 1) similar or 2) extremely different. The

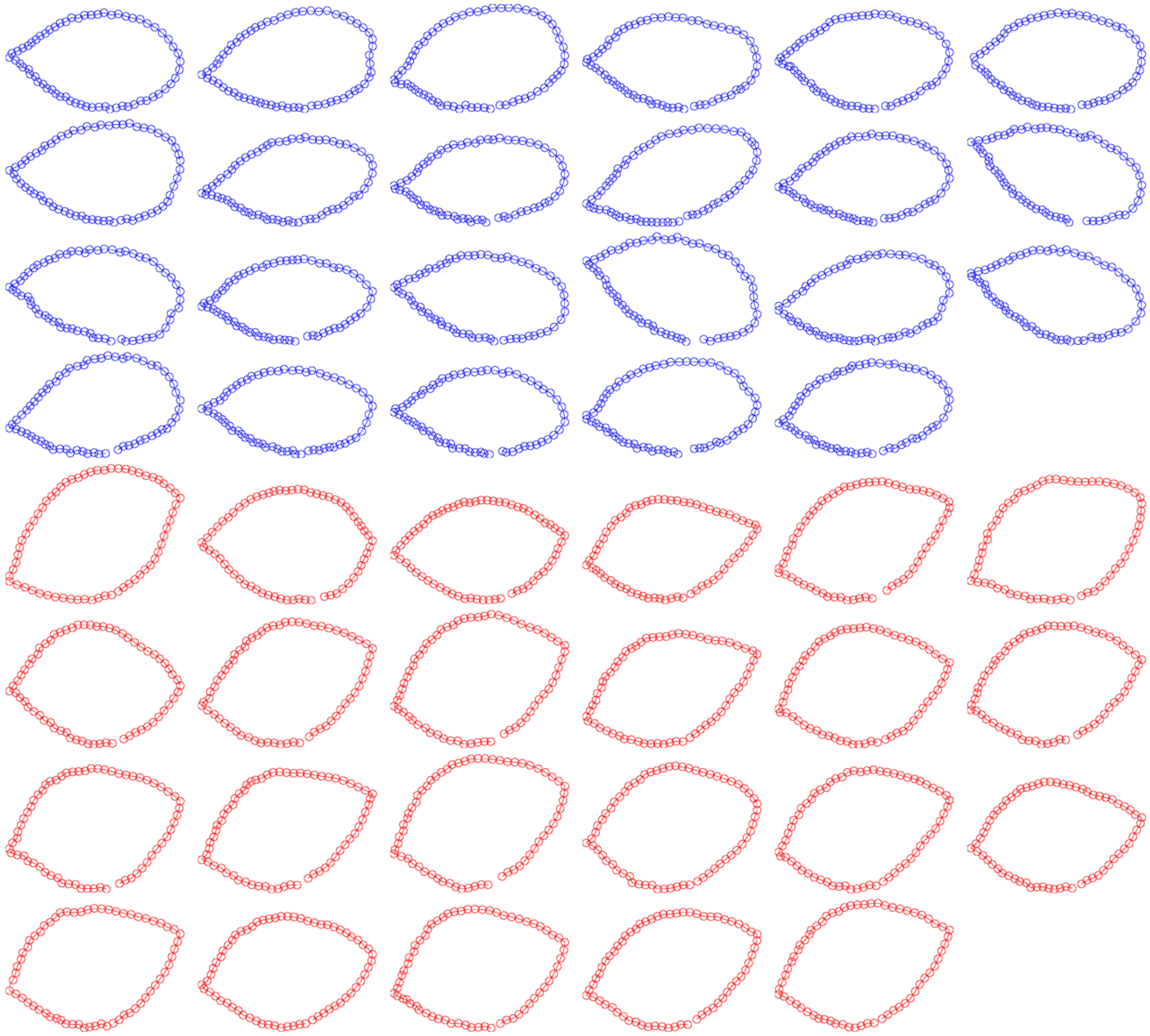


Fig. 3.7. **Tree leaf database:** The two groups of leaves from two species of trees; one in red and the other in blue. The blue groups is *Carpinus betulus*, with 23 leaf images and the red group is *Fagus sylvatica*, which also has 23 leaf images.

real challenge is being able to reject the null hypothesis when the two groups differ in subtle ways.

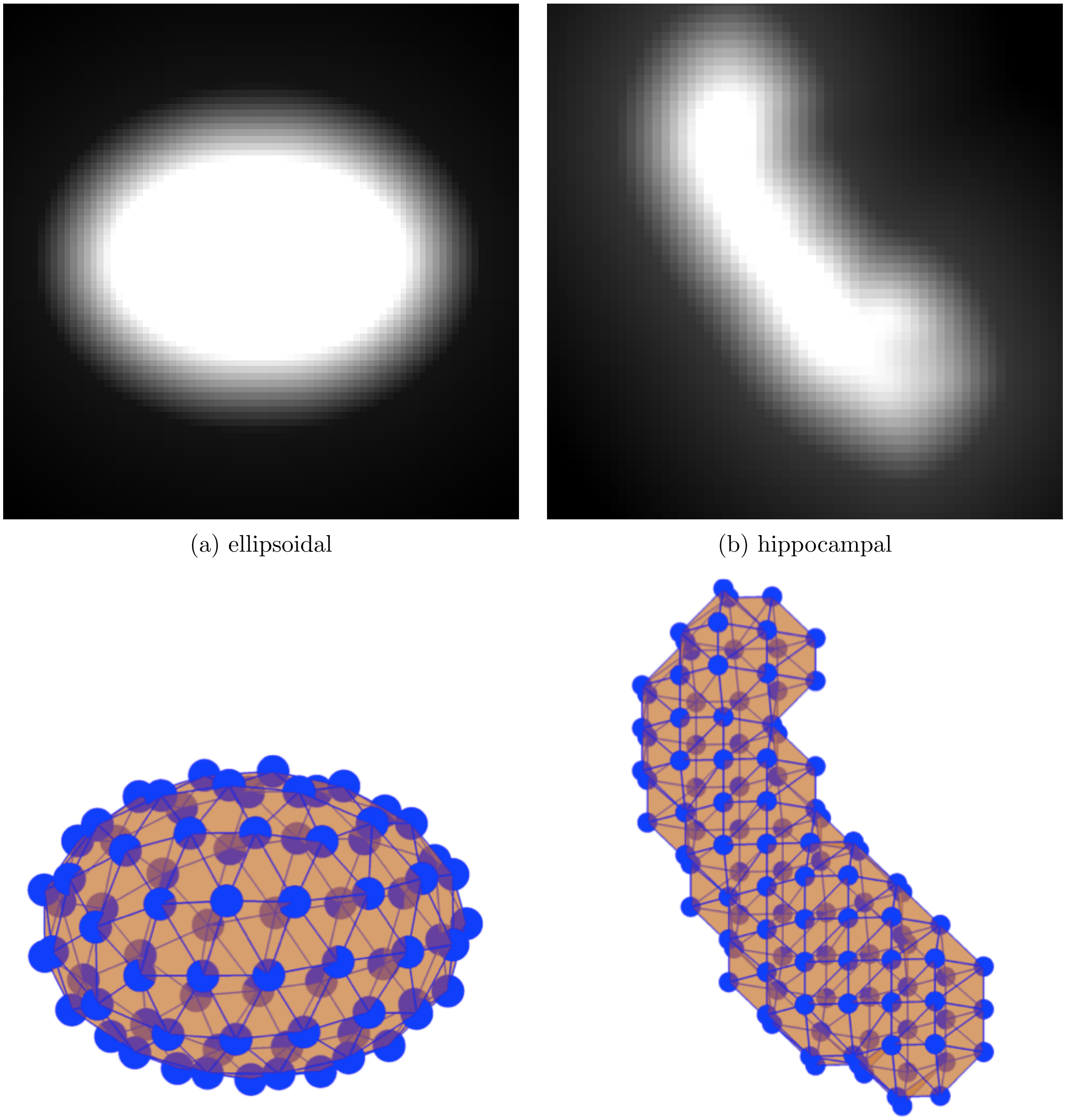
For the proposed hierarchical model, we initialize the pointsets that model shape as follows. First, we solve a groupwise registration problem on the mask images, using a similarity transform to 1) register the images, representing shape, to a common space and 2) find an average (mask) image in that space. We assume the data to be the set of voxels on the zero crossing of the mask images warped to the common space (Fig. 3.8-(a) and (b)). Then, we 1) threshold the average mask to get an object boundary, 2) embed it as the zero level set of a signed distance-transform image, and 3) generate a 3D triangular mesh for the zero level set using [121] (Fig. 3.8-(c) and (d)). Finally, we use this mesh-vertex pointset as the *initial* value for M, m_1^0, m_2^0, $\{u_i^0\}_{i=1}^{I}$, and $\{v_j^0\}_{j=1}^{J}$. We set C, C_1, C_2 to (scaled) identity. We set the σ for the Gaussian kernel, underlying the current distance, to be the average edge length in the mesh. With this initialization, we compare the proposed method with a state-of-the-art algorithm [117] implemented in the open-source software ShapeWorks [122].

3.7.3.2 Group Comparison Using Permutation Testing

After the model is fit to the data, we can perform hypothesis testing to compare any pair of groups; the null hypothesis is that the two groups of data were drawn from the same PDF. Since the shape PDF in each group is modeled using Mahalanobis distances based on means M_1, M_2 and covariances C_1, C_2, we use Hotelling's two-sample T^2 statistic to measure dissimilarity between any pair of groups. However, in 3D medical image data, the dimensionality TD can be very high compared to the number of individuals. Low sample sizes can render the F-distribution unusable. Simulating shapes with sample sizes higher than the dimensionality TD can be computationally expensive. Thus, we propose to employ distribution-free hypothesis testing, namely permutation testing, using Hotelling's T^2 as the test statistic. Permutation testing is conservative in rejecting the null hypothesis and enhances robustness to specific modeling choices, e.g., the cardinality of the shape-model pointsets and internal model-free parameters.

3.7.3.3 Result: Simulated 3D Shapes

We simulate two groups of ellipsoidal shapes (ellipsoids in canonical form; 20 pointsets per group), where the groups are subtly different from each other. Two of the axes have length 1. The lengths of the third axis for the 1) first group are drawn from a Gaussian with mean 0.9 and variance 0.01 and for the 2) second group are drawn from a Gaussian with mean 1.1 and variance 0.01. The pointsets are then rescaled to a constant norm.



(a) ellipsoidal

(b) hippocampal

(a) ellipsoidal

(b) hippocampal

Fig. 3.8. **Example for ellipsoidal shapes and hippocampal shapes:** (a) and (b) are one 2D slide of distance transform 3D image data and (c) and (d) are 3D point shape.

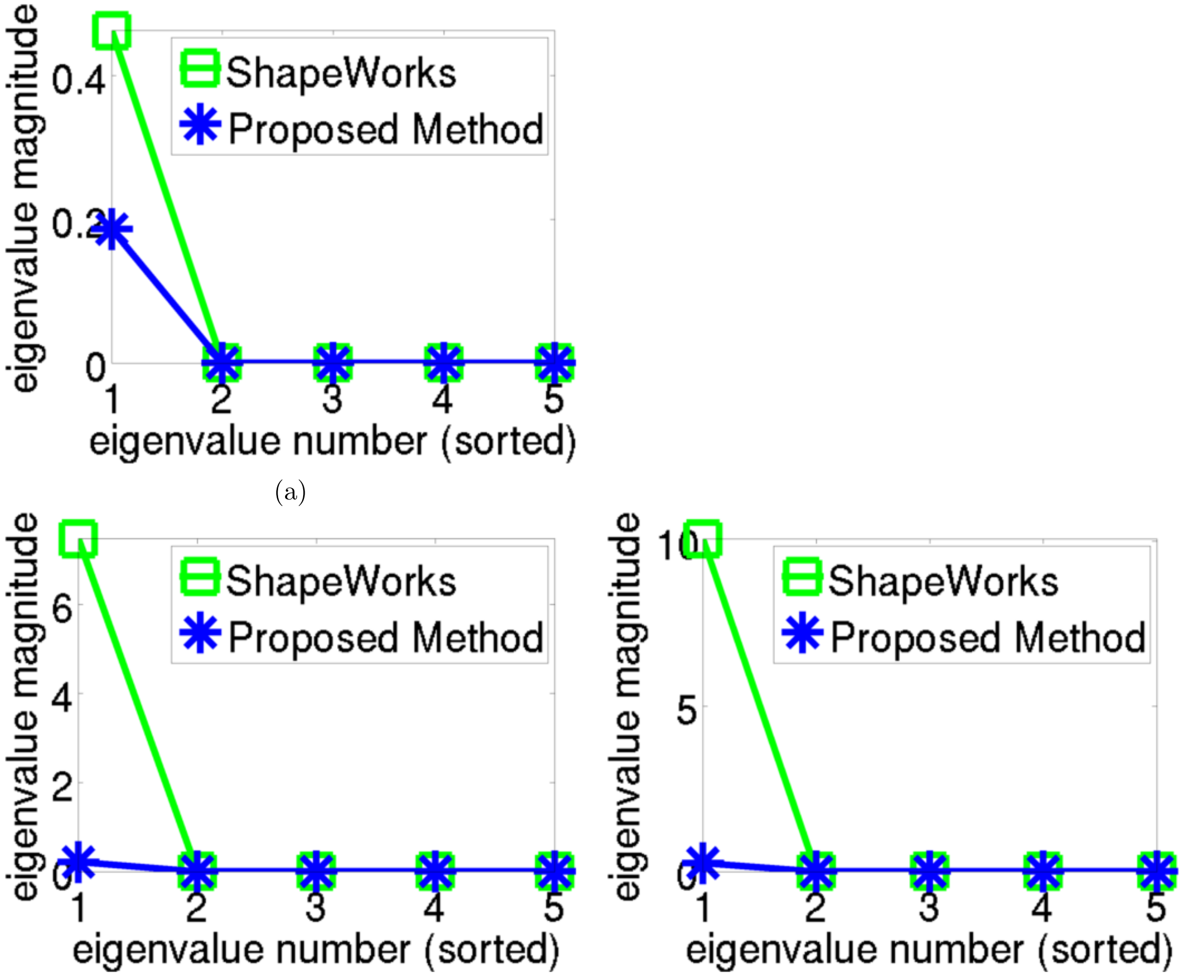
The proposed method as well as ShapeWorks 1) both employ $T = 64$ points per pointset for shape modeling and 2) both take as input equivalent information, i.e., whereas ShapeWorks takes as input a signed-distance-transform image (Fig. 3.8-(a)) representing the ellipsoids implicitly, the proposed method takes as input the corresponding zero-crossing image. With $T = 64$, the average distance between a point and its nearest neighbor, in the shape pointset, is around 10 voxels. For both methods, the covariance estimates are regularized by addition of a scaled identity matrix δI, where δ is a free parameter; the experiments explore the robustness of both approaches to changes in δ.

Fig. 3.9 and Fig. 3.10 show the results from the proposed method compared to ShapeWorks for the regularization parameter δ set to 10^{-4}. The proposed method leads to a fitted model that has smaller variances at the group level as well as the population level. This result indicates that the proposed method leads to a model that is more compact and fits the data better, which stems from improvements in optimal point placement and estimation of correspondences/parametrization. For the permutation distribution of the Hotelling's T^2 statistic, the p value for ShapeWorks is 0.05 and that for the proposed method is 0.001. Varying δ over $10^{-3}, 10^{-4}, \cdots, 10^{-10}$, we find that the p value for the proposed method stays at 0.001, but the p value of ShapeWorks varies and is never lower than 0.05. These results were unchanged when the value of the current-distance parameter σ was multiplied by factors $\in [0.5, 2]$. These results indicate that, compared to ShapeWorks, the proposed method was more robust to changes in δ and consistently produces a p value that tends to (correctly) reject the null hypothesis significantly more strongly.

3.7.3.4 Result: Hippocampal Shapes in Dementia

This section employs clinical brain magnetic resonance (MR) images from the OASIS [123] dataset. We use 10 randomly selected OASIS brains that uniformly sample the age span, including four cases with very mild to mild Alzheimer's dementia and six controls, having hippocampus segmentations manually performed by a radiologist [124], [125].

The proposed method and ShapeWorks both employ $T = 128$ points per pointset; the average distance between a point and its nearest neighbor is around five voxels. Fig. 3.11 and Fig. 3.12 show the results using $\delta = 10^{-4}$. These results were unchanged when the value of the current-distance parameter σ was multiplied by factors $\in [0.5, 2]$. The proposed method leads to a fitted model that has smaller variances, indicating a compact better-fitting model. The p value for ShapeWorks is 0.07. The p value for the proposed method is 0.03, which indicates a relatively stronger rejection of the null hypothesis.



(a)

(b)

(c)

Fig. 3.9. **Eigenspectra of the group covariance for ellipsoid shape:** (a) Eigenspectra of the population covariance C, (b) eigenspectra of the group covariance C_1, and (c) eigenspectra of the group covariance C_2.

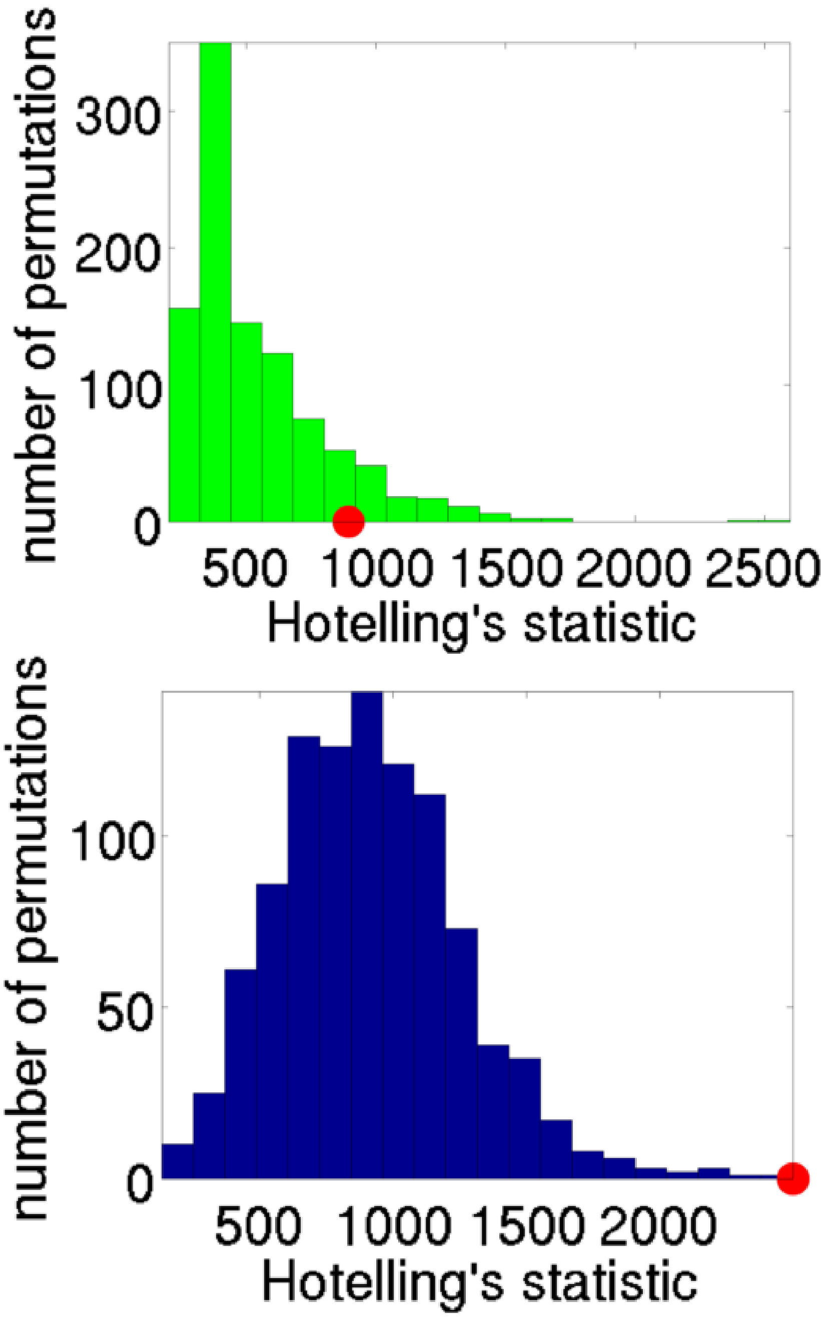
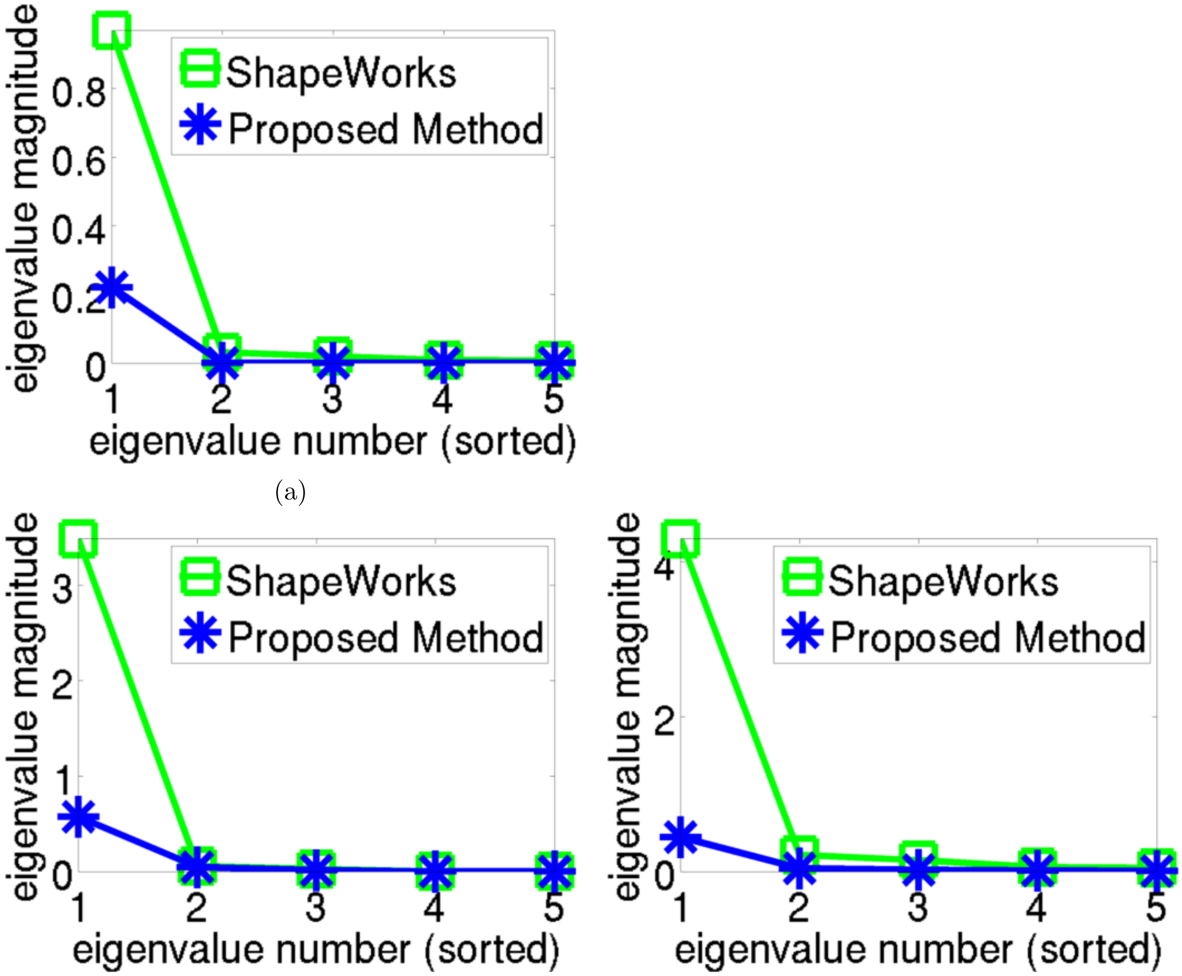


Fig. 3.10. **Permutation distribution of Hotellings T^2 test statistic for ellipsoid shape:** For ShapeWorks (**top**) and the proposed method (**bottom**); the red circle shows the value of the test statistic for the unpermuted group labeling.



(a)

(b)

(c)

Fig. 3.11. Eigenspectra of the group covariance for hippocampal shape: (a) Eigenspectra of the population covariance C, (b) eigenspectra of the group covariance C_1, and (c) eigenspectra of the group covariance C_2.

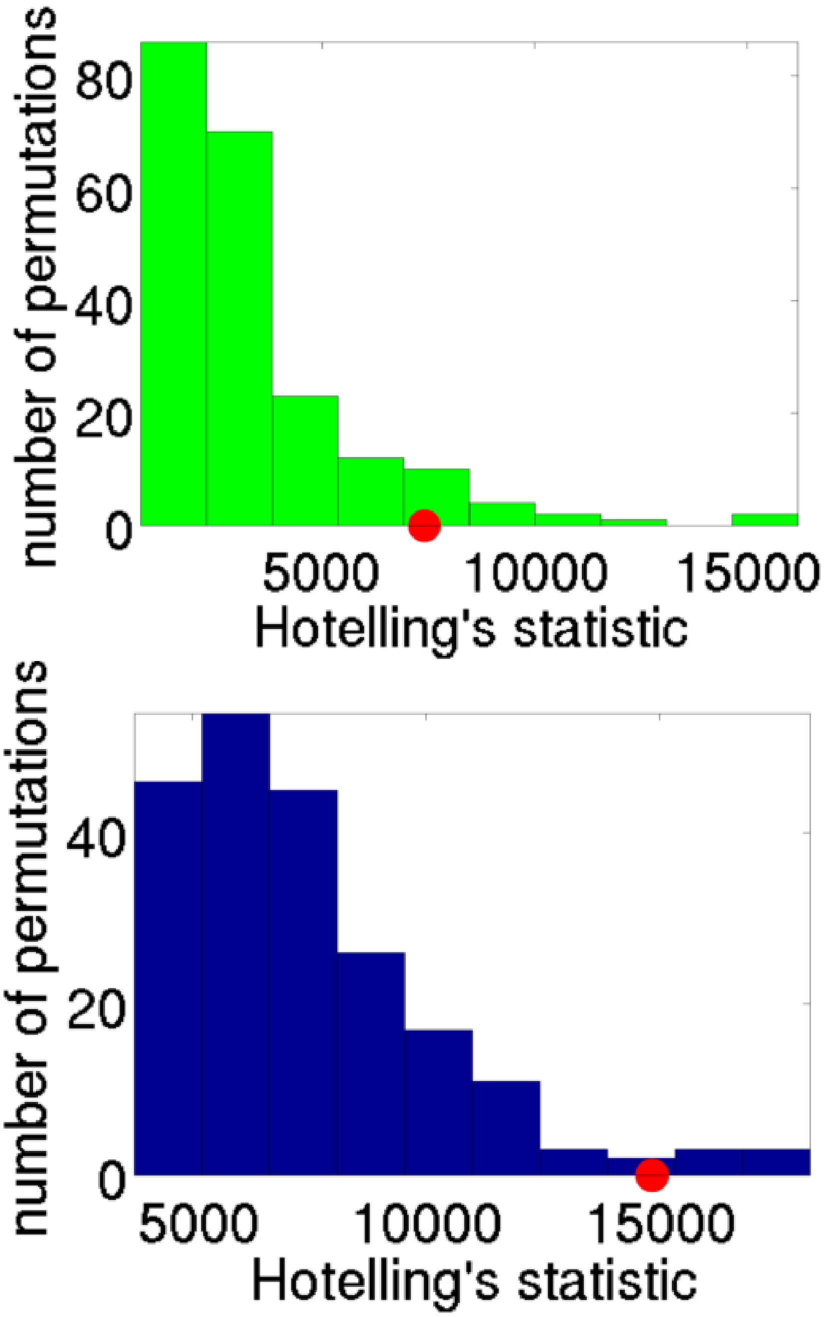


Fig. 3.12. Permutation distribution of Hotellings T^2 test statistic for hippocampal shape: For ShapeWorks (**top**) and the proposed method (**bottom**); the red circle shows the value of the test statistic for the unpermuted group labeling.

3.8 Conclusion

The results show that the proposed hierarchical model and unified-optimization approach lead to compact-fitting shape models that can differentiate subtle variations in hippocampal shapes (open-access data) better than the state-of-the-art (open-source software). The main originality in the paper is in being able to solve the three problems of point placement, correspondence, and model-parameter estimation (given data from one or more groups) as a single optimization problem. Another key originality is in being able to sample in Kendall shape space, using a novel adaptation of HMC sampling using restricted gradients. The proposed framework can benefit from more accurate and efficient schemes for modeling and estimation.

CHAPTER 4

CLUSTERING WITH PAIRWISE RELATIONSHIPS

4.1 Abstract

Semisupervised learning (SSL) has become important in current data analysis applications, where the amount of unlabeled data is growing exponentially and user input remains limited by logistics and expense. Constrained clustering, as a subclass of SSL, makes use of user input in the form of *relationships* between data points (e.g., pairs of data points belonging to the same class or different classes) and can remarkably improve the performance of unsupervised clustering in order to reflect user-defined knowledge of the relationships between particular data points. Existing algorithms incorporate such user input, heuristically, as either hard constraints or soft penalties, which are separate from any generative or statistical aspect of the clustering model; this results in formulations that are suboptimal and not sufficiently general. In this dissertation, we propose a principled, *generative approach* to probabilistically model, without ad hoc penalties, the joint distribution given by user-defined pairwise *relations*. The proposed model accounts for general underlying distributions without assuming a specific form and relies on expectation-maximization for model fitting. For distributions in a standard form, the proposed approach results in a closed-form solution for parameters updated. Results for real-world datasets demonstrate the effectiveness of the proposed generative model in reflecting user preferences with *fewer* user-defined relations compared to the state-of-the-art.

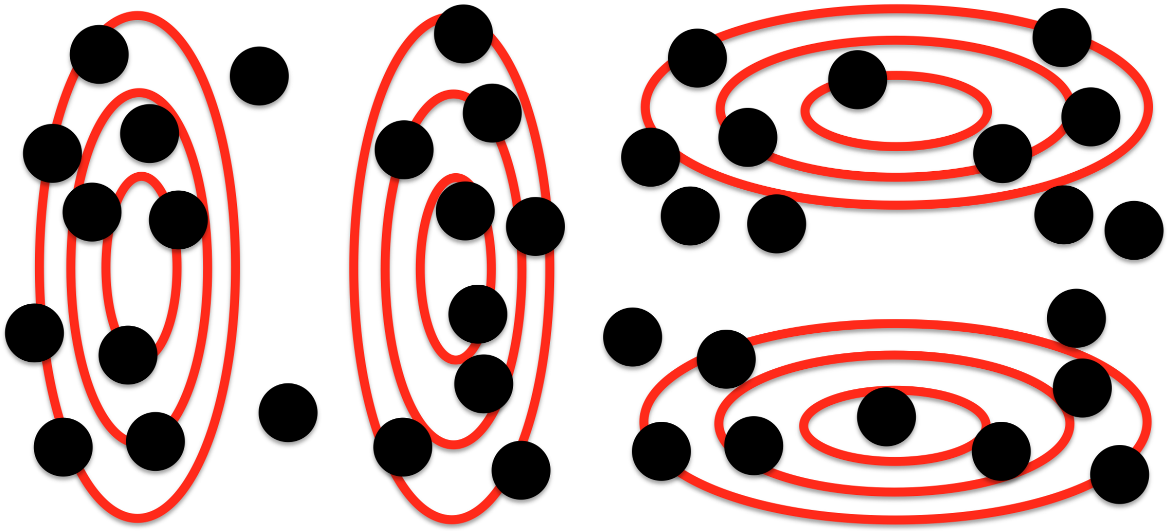
This chapter is a revised and adapted with permission (copyright by authors) from

> **Clustering With Pairwise Relationships: A Generative Approach**. *Yen-Yun Yu, Shireen Y. Elhabian, and Ross T. Whitaker.* The paper is under review by the Journal of Machine Learning Research.

4.2 Introduction

Semisupervised learning (SSL) has become a topic of significant recent interest in the context of applied machine learning, where per-class distributions are difficult to automatically separate due to limited sampling and/or limitations of the underlying mathematical model. Several applications, including content-based retrieval [126], email classification [127], gene function prediction [128], and natural language processing [129], [130], benefit from the availability of user-defined/application-specific knowledge in the presence of large amounts of complex unlabeled data, where labeled observations are often limited and expensive to acquire. In general, SSL algorithms fall into two broad categories: *classification* and *clustering.* Semisupervised classification is considered to improve supervised classification when small amounts of labeled data with large amounts of unlabeled data are available [9], [28]. For example, in a semisupervised email classification, one may wish to classify constantly increasing email messages into spam/nonspam with the knowledge of a limited amount of user-/human-based classified messages [127]. On the other hand, semisupervised clustering (SSC), also known as *constrained clustering* [131], aims to provide better performance for *unsupervised clustering* when user-based information about the *relationships* within a small subset of the observations becomes available. Such relations would involve data points belonging to the same or different classes. For example, a language-specific grammar is necessary in cognitive science when individuals are attempting to learn a foreign language efficiently. Such a grammar provides rules for prepositions that can be considered as user-defined knowledge for improving the ability to learn a new language.

To highlight the role of user-defined relationships for learning an application-specific data distribution, we consider the example in Fig. 4.1(a), which shows a maximum likelihood model estimate of a Gaussian mixture that is well supported by the data. However, an application may benefit from another good (but not optimal w.r.t. likelihood) solution as in Fig. 4.1(b), which is inconsistent with the data, but is optimal without some information in addition to the raw data points. Using a limited amount of *labeled* data and a large amount of unlabeled data could be difficult to guide the learning algorithm in the application-specific direction [28]–[31], because performance of a generative model depends on the ratio of the labeled data to unlabeled data. In contrast, previous works have shown that SSC achieves the estimate in Fig. 4.1(b), given the observed data and a small number of user-defined *relationships* that would *guide* the parameter estimation process toward a model [131] that is not only informed by the data, but also by this small amount of user input. This dissertation



(a) Mathematically Ideal Model

(b) Application-Specific Model

Fig. 4.1. **Generative model clustering example**: Because of finite sampling and modeling limitations, a distribution of points may give rise to optimal solutions that, depending on the model and the data, (a) are not well suited to the application and/or (b) are not consistent with the underlying generative model, which may require domain knowledge from a user.

addresses the problem of incorporating such user-specific relations into a clustering problem in an effective, general, and reliable manner.

Clustering data using a generative framework has some useful, important properties, including compact representations, parameter estimation for subsequent statistical analysis, and the ability to induce classifications of unseen data [132]. For the problem of estimating the parameters of generative models, the expectation-maximization (EM) algorithm [74] is particularly effective. The EM formulation is guaranteed to give maximum-likelihood (ML) estimates in the unimodal case and local maxima. Therefore, EM formulations of parameter estimation that properly account for user input in the context of SSC are of interest and are one of the contributions of this dissertation.

A flexible and efficient way to incorporate user input into SSC is in the form of *relations* between observed data points, in order to define statistical relationships among observations (rather than explicit labeling, as would be done in classification). A typical example would be for a user to examine a small subset of data and decide that some pairs of points should be in different classes, referred to as a *cannot-link* relation, and that other pairs of data points should be in the same class, i.e., *must-link*. Using these basic primitives, one may build up more complex relationships among sets of points. The concept of pairwise links was first applied to centroid-based clustering approaches, for instance, in the form of *constrained*

K-means [133], where each observation is assigned to the nearest cluster in a manner that avoids violating constraints.

Although some progress has been made in developing mechanisms for incorporating this type of user input into clustering algorithms, the need remains for a systematic, general framework that generalizes with a limited amount of user knowledge. Most state-of-the-art techniques propose adding *hard constraints* [134], where data points that violate the constraints do not contribute (i.e., all pairwise constraints must be satisfied), or *soft penalties* [135], which penalize the clustering results based on the number of violated constraints. These can lead to both a lack of generality and suboptimal solutions. For instance, in constrained K-means, introducing constraints by merely assigning a relatively small number of points to appropriate centroids does not ensure that the models (centroids) adequately respond to this user input.

In this chpater, we propose a novel, generative approach for clustering with pairwise relations that incorporates these relations into the estimation process in a precise manner. The parameters are estimated by optimizing the data likelihood under the *assumption* that individual data points are either independent samples (as in the unsupervised case) or that they have a nontrivial joint distribution, which is determined by user input. The proposed model explicitly incorporates the pairwise relationship as a property of the generative model that guides the parameter estimation process to reflect user preferences and estimates the global structure of the underlying distribution. Moreover, the proposed model is represented as a probability distribution that can take virtually any form. The results in this chapter demonstrate that the proposed optimal strategy pays off, and that it outperforms the state-of-the art on real-world datasets with significantly less user input.

4.3 Related Work

Semisupervised clustering methods typically fall into one of two categories [131]: *distance-based* methods and *constraint-based* methods. The distance-based approaches combine conventional clustering algorithms with distance metrics that are designed to satisfy the information given by user input [136]–[139]. The metrics effectively embed the points into spaces where the distances between the points with constraints are either larger or smaller to reflect the user-specified relationships. On the other hand, constraint-based algorithms incorporate the pairwise constraints into the clustering objective function, to either enforce the constraints or penalize their violation. For example, Wagstaff et al. [133] proposed the constrained K-means algorithm, which enforced user input as hard constraints in a nonprobabilistic manner as the part of the algorithm that assigns points to classes. Basu et al. [140] proposed a probabilistic framework based on a hidden Markov random field, with ad hoc soft penalties, which integrated metric learning with the constrained K-means approach, optimized by an EM-like algorithm. This work also can be applied to a kernel feature space as in [141]. Allab and Benabdeslem [142] adapted topological clustering to pairwise constraints using a self-organizing map in a deterministic manner.

Semisupervised clustering methods with generative, parametric clustering approaches have also been augmented to accommodate user input. Lu and Leen [135] proposed a penalized clustering algorithm using Gaussian mixture models (GMM) by incorporating the pairwise constraints as a prior distribution over the latent variable directly, resulting in a computationally challenging evaluation of the posterior. Such a penalization-based formulation results in a model with no clear generative interpretation and a stochastic expectation step that requires Gibbs sampling. Shental et al. [134] proposed a GMM with equivalence constraints that defines the data from either the same or a different source. However, for the *cannot-link* case, they used the Markov network to describe the dependence between a pair of latent variables and sought the optimal parameter by gradient ascent. Their results showed that the cannot-link relationship was unable to impact the final parameter estimation (i.e., such a relation was ineffective). Further, they imposed user input as *hard* constraints, where data points that violate the constraints did not contribute to the parameter estimation process. A similar approach in [143] proposed to treat the constraint as an additional random variable that increases the complexity of the optimization process. Further, their approach focused only on *must-link*. In this dissertation, we propose a novel solution to incorporating user-defined data relationships into clustering problems, so that cannot-link and must-link relations can be included in a unified framework in a way that they are computed efficiently using an EM algorithm with very modest computational demands. Moreover, the proposed formulation is general in that it can 1) accommodate any kind of relation that can be expressed as a joint probability and 2) incorporate, in principle, any probability distribution (generative model). For GMMs, however, this formulation results in a particularly attractive algorithm that entails a closed-form solution for the mean and covariance and a relatively inexpensive, iterative, constrained, nonlinear optimization for the mixing parameters.

Recently, EM-like algorithms for SSL (and clustering in particular) have received significant attention in natural language processing [144], [145]. Graca et al. [144] proposed an EM approach with a posterior constraint that incorporates the expected values of specially designed auxiliary functions of the latent variables to influence the posterior distribution to favor user input. Because of the lack of probabilistic interpretation, the expectation step is not influenced by user input, and the results are not optimal.

Unlike the generative approach, graph-based methods group the data points according to similarity and do not necessarily assume an underlying distribution. Graph-based, semi-supervised clustering methods have been demonstrated to be promising when user input is available [146]–[148]. However, graph-based methods are not ideal classifiers when a new data point is presented due to their transductive property, i.e., their inability to learn the general rule from the specific training data [132], [149]. In order to classify a new data point, other than rebuilding the graph with the new data point, one likely solution is to build a separate inductive model on top of the output of the graph-based method (e.g., K-means or GMM); user input would need to be incorporated into this new model.

The work in this dissertation is distinct from the aforementioned works in the following aspects:

- We present a *fully* generative approach, rather than a heuristic approach of imposing hard constraints or adding ad hoc penalties.
- The proposed generative model reflects user preferences while maintaining a probabilistic interpretation, which allows it to be generalized to take advantage of *alternative* density models or optimization algorithms.
- The proposed model clearly deals with the must-link *and* cannot-link cases in a unified framework and demonstrates that solutions using must-link and cannot-link together or independently are tractable and effective.
- Instead of pairwise constraints, the statistical interpretation of pairwise relationships allows the model estimation to converge to a distribution that follows user preferences with *less* domain knowledge.
- In the proposed algorithm, the parameter estimation is very similar to a standard EM in terms of ease of implementation and efficiency.

4.4 Clustering With Pairwise Relationships

The proposed model incorporates user input in the form of relations between pairs of points that are in the same class (*must-link*) or different classes (*cannot-link*). The *must-link* and *cannot-link* relationships are a natural and practical choice since the user can guide the clustering without having a specific preconceived notion of classes. These pairwise relationships are typically not sufficiently dense or complete to build a full discriminative model, and yet they may be helpful in discovering the underlying structure of the unlabeled data. For data points that have no user input, we assume that they are independent, random samples. The pairwise relationships give rise to an associate generative model with a joint distribution that reflects the nature of the user input.

The parameters are estimated as an ML formulation through an EM algorithm that discovers the global structure of the underlying distribution, reflecting the user-defined relations. Unlike previous works that include user input in a specific model (e.g., a GMM) through either hard constraints [134] or soft penalties [135], in this work we propose an ML estimation based on a generative model, without ad-hoc penalties.

4.4.1 Generative Models: Unsupervised Scenario

In this section, we first introduce generative models for an unsupervised scenario. Suppose the unconstrained generative model consists of M classes. $\mathcal{X} = \{\mathbf{x}_n \in \mathbb{R}^d\}_{n=1}^N$ denotes the observed dataset without user input. Dataset $\mathcal{X}$ is associated with *latent* set $\mathcal{Z} = \{\mathbf{z}_n\}_{n=1}^N$ where $\mathbf{z}_n = [z_n^1, ..., z_n^M]^T \in \{0,1\}^M$ with $z_n^m = 1$ if and only if the corresponding data point $\mathbf{x}_n$ was generated from the mth class, subject to $\sum_{m=1}^M z_n^m = 1$. Therefore, we can obtain the soft label for a data point $\mathbf{x}$ by estimating $p(z^m|\mathbf{x})$. The probability that a data point $\mathbf{x}$ is generated from a generative model with parameters $\boldsymbol{\vartheta}$ is

$$p(\mathbf{x}|\boldsymbol{\vartheta}) = \sum_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}, \boldsymbol{\vartheta})p(\mathbf{z}). \tag{4.1}$$

The likelihood of the observed data points governed by the model parameters is

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}, \boldsymbol{\vartheta}) := p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\vartheta}) = \prod_{m=1}^{M} \prod_{n\in[1,N]:z_n^m=1} p(\mathbf{x}_n) \tag{4.2}$$

$$= \prod_{m=1}^{M}\prod_{n=1}^{N} p(\mathbf{x}_n, z_n^m) = \prod_{m=1}^{M}\prod_{n=1}^{N} \Big[p(\mathbf{x}_n|z_n^m, \boldsymbol{\vartheta})p(z_n^m)\Big]^{z_n^m} \tag{4.3}$$

where the condition on the product term in equation (4.2) is restricted to data points $\mathbf{x}_n$ generated from the mth class. The joint probability in equation (4.3) is expressed, using Bayes' rule, in terms of the conditional probability $p(\mathbf{x}_n|z_n^m, \boldsymbol{\vartheta})$ and the mth class prior probability $p(z_n^m)$. In the rest of the formulation, to simplify the representation, we use $p(\mathbf{x}_n|z_n^m) = p(\mathbf{x}_n|z_n^m, \boldsymbol{\vartheta})$.

4.4.2 Generative Model With Pairwise Relationships

The definition of a pairwise relation in the proposed generative model is similar to that in the unsupervised case, yet such relations are propagated to the latent variables level. In particular, $\mathcal{M} = \{(i,j)\}$ denotes a set of must-link relations where the pair $\mathbf{x}_i$ and $\mathbf{x}_j$ was generated from the same class; hence, the pair $(\mathbf{x}_i, \mathbf{x}_j)$ shares a single latent variable $\mathbf{z}_{\{ij\}}$. The same logic is applied to the cannot-link relations where $\mathcal{C} = \{(a,b)\}$ denotes a set of cannot-link relations encoding that $\mathbf{x}_a$ and $\mathbf{x}_b$ were generated from distinct classes; therefore, $\mathbf{z}_a \neq \mathbf{z}_b$. Including $\mathcal{M}$ and $\mathcal{C}$, the data points are now expanded to be $\mathcal{X} := \{\mathbf{x}_1, \ldots \mathbf{x}_N, \mathcal{M}, \mathcal{C}\}$. Thus, the *modified* complete-data likelihood function $\mathcal{J}(\cdot)$ that would reflect user input is (refer to Fig. 4.2 for the graphical representation)

$$\begin{aligned}\mathcal{J}(\mathcal{X}, \mathcal{Z}, \mathcal{M}, \mathcal{C}, \boldsymbol{\vartheta}) &:= p(\mathcal{X}, \mathcal{Z} | \mathcal{M}, \mathcal{C}, \boldsymbol{\vartheta}) \\ &= \mathcal{L}(\mathcal{X}, \mathcal{Z}, \boldsymbol{\vartheta})\ \mathcal{S}(\mathcal{X}, \mathcal{Z}, \mathcal{M}, \boldsymbol{\vartheta})\ \mathcal{D}(\mathcal{X}, \mathcal{Z}, \mathcal{C}, \boldsymbol{\vartheta}). \qquad (4.4)\end{aligned}$$

$\mathcal{S}(\cdot)$ and $\mathcal{D}(\cdot)$ are the likelihood of pairwise data points. The likelihood of the set of all pairs of must-link data points $\mathcal{S}$ is, therefore,

$$\begin{aligned}\mathcal{S}(\mathcal{X}, \mathcal{Z}, \mathcal{M}, \boldsymbol{\vartheta}) &:= p(\mathcal{X}, \mathcal{Z} | \mathcal{M}, \boldsymbol{\vartheta}) \\ &= \prod_{m=1}^{M} \prod_{(i,j)\in\mathcal{M}} p(\mathbf{x}_i, \mathbf{x}_j, z^m_{\{ij\}}) \\ &= \prod_{m=1}^{M} \prod_{(i,j)\in\mathcal{M}} \left[p(\mathbf{x}_i | z^m_{\{ij\}}) p(\mathbf{x}_j | z^m_{\{ij\}}) p(z^m_{\{ij\}}) \right]^{z^m_{\{ij\}}}. \qquad (4.5)\end{aligned}$$

The likelihood of the cannot-link data points explicitly reflects the fact that they are drawn from distinct classes. Therefore, the joint probability of the labeling vectors $\mathbf{z}_a$ and $\mathbf{z}_b$ for all $(a,b) \in \mathcal{C}$ is as follows:

$$\begin{aligned}p(z^m_a, z^m_b) &:= p(z^m_a | z^m_b) p(z^m_b) = p(z^m_b | z^m_a) p(z^m_a) \qquad (4.6) \\ &= \begin{cases} \dfrac{p(z^m_a)^{z^m_a} p(z^m_b)^{z^m_b}}{1 - \sum_{m'=1}^{M} p(z^{m'}_a)^2} & z^m_a \neq z^m_b \\ 0 & z^m_a = z^m_b \end{cases} \qquad (4.7) \\ &= \frac{(1 - z^m_a z^m_b) p(z^m_a)^{z^m_a} p(z^m_b)^{z^m_b}}{1 - \sum_{m'=1}^{M} p(z^{m'}_a)^2}. \qquad (4.8)\end{aligned}$$

The proposed joint distribution reflects the cannot-link constraints by assigning a zero joint probability of $\mathbf{x}_a$ and $\mathbf{x}_b$ being generated from the same class, and takes into account the effect of this relation on the normalization term of the joint distribution $p(z^m_a, z^m_b)$. As such, the cannot-link relations contribute to the posterior distribution as follows:

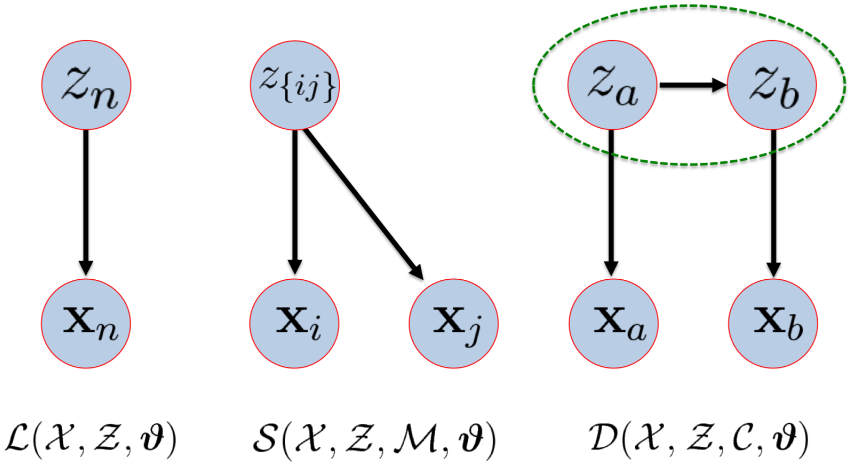

z_n
$\mathbf{x}_n$
$z_{\{ij\}}$
$\mathbf{x}_i$
$\mathbf{x}_j$
z_a
z_b
$\mathbf{x}_a$
$\mathbf{x}_b$
$\mathcal{L}(\mathcal{X}, \mathcal{Z}, \boldsymbol{\vartheta})$
$\mathcal{S}(\mathcal{X}, \mathcal{Z}, \mathcal{M}, \boldsymbol{\vartheta})$
$\mathcal{D}(\mathcal{X}, \mathcal{Z}, \mathcal{C}, \boldsymbol{\vartheta})$

Fig. 4.2. The graphical representation of the proposed generative model with complete data-likelihood. The $\mathcal{L}(\cdot)$ is from the standard generative model with independent samples. The $\mathcal{S}(\cdot)$ shows the must-link data points pair $\mathbf{x}_i$ and $\mathbf{x}_j$ shares a single latent variable $z_{\{ij\}}$. The $\mathcal{D}(\cdot)$ shows the cannot-link data points pair $\mathbf{x}_a$ and $\mathbf{x}_b$, where the green dashed line indicates the joint probability of z_a and z_b.

$$\begin{aligned}\mathcal{D}(\mathcal{X},\mathcal{Z},\mathcal{C},\boldsymbol{\vartheta}) &:= p(\mathcal{X},\mathcal{Z}|\mathcal{C},\boldsymbol{\vartheta})\\ &= \prod_{m=1}^{M}\prod_{(a,b)\in\mathcal{C}} p(\mathbf{x}_a,\mathbf{x}_b,z_a^m,z_b^m)\\ &= \prod_{m=1}^{M}\prod_{(a,b)\in\mathcal{C}} \Big[p(\mathbf{x}_a|z_a^m)\Big]^{z_a^m}\Big[p(\mathbf{x}_b|z_b^m)\Big]^{z_b^m} p(z_a^m,z_b^m). \end{aligned} \tag{4.9}$$

4.4.3 Expectation-Maximization With Pairwise Relationships

Given the joint distribution $p(\mathcal{X},\mathcal{Z}|\mathcal{M},\mathcal{C},\boldsymbol{\vartheta})$, the objective is to maximize the log-likelihood function $\log\mathcal{J}$ with respect to the parameters $\boldsymbol{\vartheta}$ of the generative process in a manner that would discover the global structure of the underlying distribution and reflect user input. This objective can be achieved using an EM algorithm.

4.4.3.1 E-Step

In the E-step, we estimate the posterior of the latent variables using the current parameter values $\boldsymbol{\vartheta}^{\text{old}}$.

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\vartheta},\boldsymbol{\vartheta}^{\text{old}}) &= \mathbb{E}_{\mathcal{Z}}[\log\ \mathcal{J}]\\ &= \sum_{\mathcal{Z}} p(\mathcal{Z}|\mathcal{X},\mathcal{M},\mathcal{C},\boldsymbol{\vartheta}^{\text{old}})\log p(\mathcal{X},\mathcal{Z}|\mathcal{M},\mathcal{C},\boldsymbol{\vartheta}).\end{aligned} \tag{4.10}$$

$\underline{\mathcal{L}\text{-term:}}$ Taking the expectation of $\log\mathcal{L}$ with respect to the posterior distribution of z_n^m and bearing in mind that the latent variable $\mathbf{z}$ is a binary variable,

$$\mathbb{E}_{z_n^m|\mathbf{x}_n}[z_n^m] = \frac{p(\mathbf{x}_n|z_n^m)p(z_n^m)}{\sum_{m'=1}^{M} p(\mathbf{x}_n|z_n^{m'})p(z_n^{m'})}. \tag{4.11}$$

$\underline{\mathcal{S}\text{-term:}}$ Taking the expectation of $\log\mathcal{S}$ with respect to the must-link posterior distribution of $z_{\{ij\}}^m$ results in

$$\mathbb{E}_{z_{\{ij\}}^m|\mathbf{x}_i,\mathbf{x}_j}[z_{\{ij\}}^m] = \frac{p(\mathbf{x}_i|z_{\{ij\}}^m)p(\mathbf{x}_j|z_{\{ij\}}^m)p(z_{\{ij\}}^m)}{\sum_{m'=1}^{M} p(\mathbf{x}_i|z_{\{ij\}}^{m'})p(\mathbf{x}_j|z_{\{ij\}}^{m'})p(z_{\{ij\}}^{m'})}. \tag{4.12}$$

$\underline{\mathcal{D}\text{-term:}}$ Because the proposed model does not allow $\mathbf{x}_a$ and $\mathbf{x}_b$ to be from the same class, the expectation of equation (4.8) in the $\log\mathcal{D}-$ that both will have the same class assignment vanishes, which can be shown using Jensen's inequality as follows:

$$\begin{aligned}\mathbb{E}_{z_a^m,z_b^m|\mathbf{x}_a,\mathbf{x}_b}[\log(1-z_a^m z_b^m)] &\le \log\left(1-\mathbb{E}_{z_a^m,z_b^m|\mathbf{x}_a,\mathbf{x}_b}[z_a^m z_b^m]\right)\\ &= \log(1-0) = 0.\end{aligned} \tag{4.13}$$

Hence, we can set $\log(1 - z_a^m z_b^m) = 0$ in equation (4.8). The expectation of the $\log \mathcal{D}$ term with respect to z_a^m is

$$\begin{aligned}\mathbb{E}_{z_a^m|\mathbf{x}_a,\mathbf{x}_b}[z_a^m] = p(z_a^m|\mathbf{x}_a, \mathbf{x}_b) &= \sum_{m'=1}^{M} p(z_a^m, z_b^{m'}|\mathbf{x}_a, \mathbf{x}_b) \\ &= \frac{\sum_{m'=1}^{M} p(\mathbf{x}_a|z_a^m)p(\mathbf{x}_b|z_b^{m'})p(z_a^m, z_b^{m'})}{\sum_{m''=1}^{M}\sum_{m'''=1}^{M} p(\mathbf{x}_a|z_a^{m''_k})p(\mathbf{x}_b|z_b^{m'''_k})p(z_a^{m''}, z_b^{m'''})}. \end{aligned} \tag{4.14}$$

In a like manner, we can write down the expectation of z_b^m.

4.4.3.2 M-Step

In the M-step, therefore, we update the $\boldsymbol{\vartheta}^{\text{new}}$ by maximizing equation (4.10) and fixing the posterior distribution that we estimated in the E-step.

$$\boldsymbol{\vartheta}^{\text{new}} = \arg\max_{\boldsymbol{\vartheta}} \quad \mathcal{Q}(\boldsymbol{\vartheta}, \boldsymbol{\vartheta}^{\text{old}}). \tag{4.15}$$

Different density models result in different update mechanisms for the respective model parameters. In the next subsection, we elaborate on an example of the proposed model to illustrate the idea of the M-step for the case of Gaussian mixture models.

4.4.4 Gaussian Mixture Model With Pairwise Relationships

Consider employing a single distribution (e.g., a Gaussian distribuion) for each class probability $p(\mathbf{x}|z^m)$. The proposed model, therefore, becomes the Gaussian mixture model (GMM) with pairwise relationships. The parameter of the GMM is $\boldsymbol{\vartheta} = \{\alpha_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}_{m=1}^M$, such that $\alpha_m \in [0, 1]$ is the mixing parameter for the *class* proportion subject to $\sum_{m=1}^M \alpha_m = 1$ and $p(z^m) = \alpha_m$. $\boldsymbol{\mu}_m \in \mathbb{R}^d$ is the mean parameter, and $\boldsymbol{\Sigma}_m \in \mathbb{R}^{d\times d}$ is the covariance associated with the mth class. By taking the derivative of equation (4.10) with respect to $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$, we can get

$$\boldsymbol{\mu}_m = \left(\sum_{n=1}^{N} \ell_n^m \mathbf{x}_n + \sum_{(i,j)\in\mathcal{M}} s_{ij}^m \left[\mathbf{x}_i + \mathbf{x}_j\right] + \sum_{(a,b)\in\mathcal{C}} \left[d_a^m \mathbf{x}_a + d_b^m \mathbf{x}_b\right]\right) \Big/ Z \tag{4.16}$$

$$\boldsymbol{\Sigma}_m = \left(\sum_{n=1}^{N} \ell_n^m \mathbf{S}_n^m + \sum_{(i,j)\in\mathcal{M}} s_{ij}^m \left[\mathbf{S}_i^m + \mathbf{S}_j^m\right] + \sum_{(a,b)\in\mathcal{C}} \left[d_a^m \mathbf{S}_a^m + d_b^k \mathbf{S}_b^m\right]\right) \Big/ Z \tag{4.17}$$

$$Z = \sum_{n=1}^{N} \ell_n^m + 2\sum_{(i,j)\in\mathcal{M}} s_{ij}^m + \sum_{(a,b)\in\mathcal{C}} \left[d_a^m + d_b^m\right] \tag{4.18}$$

where $\ell_n^m = p(z_n^m|\mathbf{x}_n)$, $s_{ij}^m = p(z_{\{ij\}}^m|\mathbf{x}_i, \mathbf{x}_j)$, $d_a^m = p(z_a^m|\mathbf{x}_a, \mathbf{x}_b)$ and the sample covariance $\mathbf{S}_n^m = (\mathbf{x}_n - \boldsymbol{\mu}_m)(\mathbf{x}_n - \boldsymbol{\mu}_m)^T$.

Estimating the mixing parameters α_m, on the other hand, entails the following constrained nonlinear optimization, which can be solved using sequential quadratic programming with Newton-Raphson steps [150], [151]. Let $\boldsymbol{\alpha} \in \mathbb{R}^M$ denote the vector of mixing parameters. Given the current estimate of the mean vectors and covariance matrices, the new estimate of the mixing parameters can be solved for using the optimization problem defined in (4.19),

$$\begin{aligned} &\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} -\mathcal{Q}(\boldsymbol{\vartheta}, \boldsymbol{\vartheta}^{\text{old}}) \\ &\text{s.t.} \qquad \mathbf{1}^T\boldsymbol{\alpha} - 1 = 0 \quad \text{and} \quad \alpha_m \geq 0 \ \ \forall m \in [1, M] \end{aligned} \tag{4.19}$$

where the initialization can be obtained using the closed-form solution obtained from discarding the nonlinear part, which ignores the normalization term $\log(1 - \sum_{m'=1}^{M} \alpha_{m'}^2)$. The energy function is convex, and we have found that this iterative algorithm typically converges in three to five iterations and does not represent a significant computational burden.

4.4.4.1 Multiple Mixture Clusters Per Class

In order to group the data that lie on the subspace (e.g., manifold structure) more explicitly, multiclusters to model per class have been widely used in unsupervised clustering by representing the density model in a hierarchical structure [152]–[158]. Because of its natural representation of data, the hierarchical structure can be built using either a top-down or bottom-up approach, in which the first approach tries to decompose one cluster into several small clusters, whereas the second starts with grouping several clusters into one cluster. The multicluster per class strategy also has been proposed when both labeled data and unlabeled data are available [159]–[166]. However, previous works indicated [29]–[31], [167] that the labeled data is unable to impact the final parameter estimation if the initial model assumption is incorrect. Moreover, it is not clear how to employ the previous works in regard to pairwise links instead of labeled data.

In this section, we propose to use the generative mixture of Gaussian distributions for each class probability $p(\mathbf{x}|z^m)$. In this form, we use multiclusters to model one class that overcomes data on a manifold structure. Therefore, in addition to the latent variable set $\mathcal{Z}$, $\mathcal{X}$ is also associated with the *latent* variable set $\mathcal{Y} = \{\mathbf{y}_n\}_{n=1}^N$ where $\mathbf{y}_n = [y_n^1, ..., y_n^{m_K}]^T \in \{0,1\}^{m_K}$ with $y_n^{m_k} = 1$ if and only if the corresponding data point $\mathbf{x}_n$ was generated from the kth cluster in the mth class, subject to $\sum_{m_k=1}^{m_K} y_n^{m_k} = 1$; m_K is the number of clusters in the mth class. The parameter of the generative mixture model is $\boldsymbol{\vartheta} = \{\alpha_m, \boldsymbol{\Theta}_m\}_{m=1}^M$, α_m is the mixing parameter for the *class* proportion and is the same as α_m in section 4.4.4. The parameter of the mth class is $\boldsymbol{\Theta}_m = \{\pi_{m_k}, \Theta_{m_k}\}_{m_k=1}^{m_K}$ where $\Theta_{m_k} = \{\boldsymbol{\mu}_{m_k}, \boldsymbol{\Sigma}_{m_k}\}$, such that $\pi_{m_k} \in [0,1]$ is the mixing parameter for the *cluster* proportion subject to $\sum_{m_k=1}^{m_k} \pi_{m_k} = 1$, $\boldsymbol{\mu}_{m_k} \in \mathbb{R}^d$ is the mean parameter, and $\boldsymbol{\Sigma}_{m_k} \in \mathbb{R}^{d\times d}$ is the covariance associated with the kth cluster in the mth class. The probability that an unsupervised data point $\mathbf{x}$ is generated from a generative mixture model given parameters $\boldsymbol{\vartheta}$ is

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \boldsymbol{\vartheta}) = \prod_{m=1}^{M} \prod_{m_k=1}^{m_K} \prod_{n=1}^{N} \left[\left[p(\mathbf{x}_n | y_n^{m_k}) p(y_n^{m_k} | z_n^m) \right]^{y_n^{m_k}} p(z_n^m) \right]^{z_n^m} \tag{4.20}$$

where

$$p(z_n^m) = \alpha_m;\ p(y_n^{m_k} | z_n^m) = \pi_{m_k};\ p(\mathbf{x} | y_n^{m_k}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{m_k}, \boldsymbol{\Sigma}_{m_k}), \tag{4.21}$$

and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{m_k}, \boldsymbol{\Sigma}_{m_k})$ is the Gaussian distribution. The definition of equation (4.21) can be used to describe the $\mathcal{S}(\cdot)$ in equation (4.5) and the $\mathcal{D}(\cdot)$ in equation (4.9). In the E-step, the posterior of latent variable $\mathcal{Z}$ can be estimated by marginalization of the $\mathcal{Y}$ directly. In the M-step, we update the parameters by maximizing equation (4.10), which is similar to the GMM case in section 4.4.4 (see the Appendix A for details). Last, if $m_K = 1$, we have $\mathbf{y}_n = [y_n^1]$ and equation (4.20) becomes the GMM, i.e., one cluster/single Gaussian distribution per class.

4.5 Experiment

In this section, we demonstrate the effectiveness of the proposed generative model on a synthetic dataset as well as on well-known datasets where the number of links can be significantly reduced compared to state-of-the-art.

4.5.1 Experimental Settings

To illustrate the method, we start with the case of $p(\mathbf{x}|z^m)$: a mixture of Gaussians ($m_K > 1$) and a single Gaussian distribution ($m_K = 1$) and with $m_K = 1$ (i.e., one cluster per class). To initialize the mean vectors for each class, we use *K-means++* [168], which is similar to the Gonzalez algorithm [169] without being completely greedy. Afterward, we assign every observed data point to its nearest initial mean where initial covariance matrices for each class are computed. We initially assume equally probable classes where the mixing parameters are set to $1/M$. When $m_K > 1$ (i.e., multiclusters per class), we initialize the parameters of the kth cluster in the mth class using the aforementioned strategy, but only on the data points that have been assigned to the mth class after the above initialization. To mimic user-preferences and assess the performance of the proposed model as a function of the number of available relations, pairwise relations are created by randomly selecting a pair of observed data points and using the knowledge of the distributions. If the points are assigned to the same cluster based on their ground-truth labeling, we move them to the *must-link* set, otherwise, to the *cannot-link* set. We perform 100 trials for all experiments. Each trial is constructed by the random initialization of the model parameters and random pairwise relations.

We compare the proposed model, *generative model with pairwise relation* (**GM-PR**), to the unconstrained **GMM**, unconstrained **spectral clustering (SC)**, and four other state-of-the-art algorithms: 1) **GMM-EC**: GMM with the equivalence constraint [134], 2) **EM-PC**: EM with the posterior constraint [144]; it is worth mentioning that **EM-PC** works only for *cannot-link*, 3) **SSKK**: Constrained kernel K-means [141], and 4) **CSC** : Flexible constrained spectral clustering [147]. For SC, SSKK, and CSC, the similarity matrix is computed by the RBF kernel, whose parameter is set by the average squared distance between all pairs of data points.

We use *purity* [170] for performance evaluation, which is a scalar value ranging from 0 to 1 where 1 is the best. Purity can be computed as follows: each class m is assigned to the most frequent ground-truth label $g(m)$, then, purity is measured by counting the number of correctly assigned observed data points in every ground truth class and dividing the total number of observed data. The assignment is according to the highest probability of the posterior distribution.

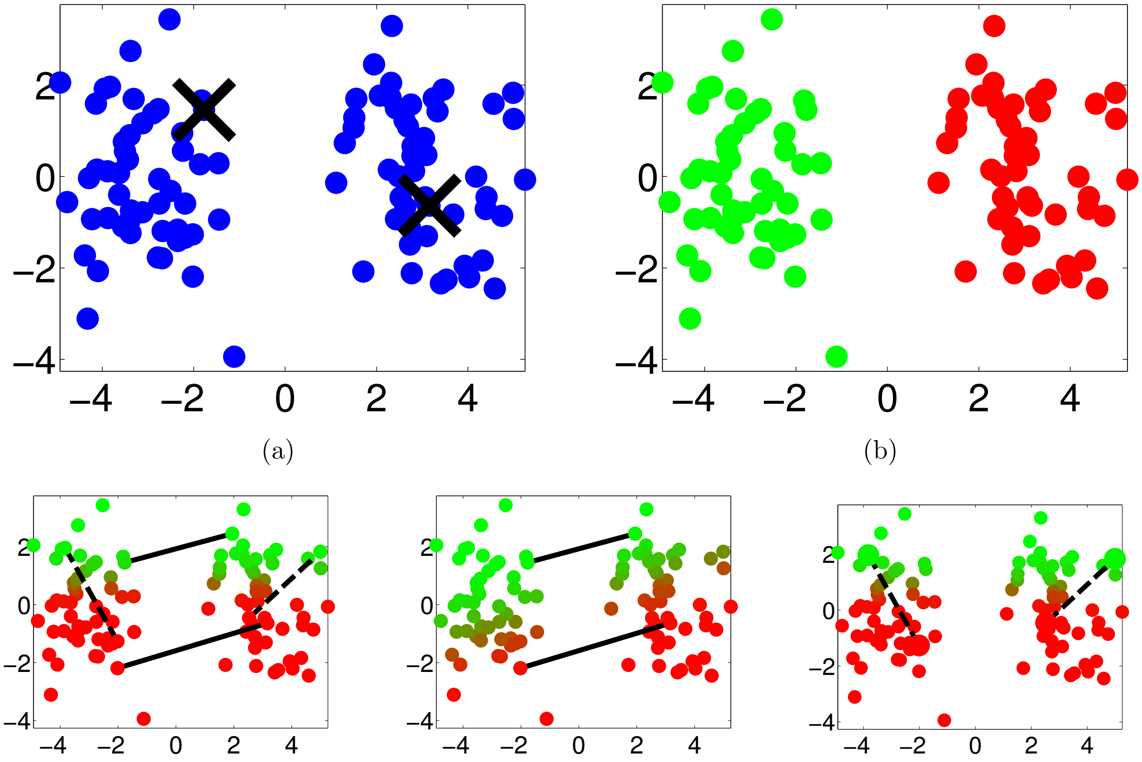
4.5.2 Results: Single Gaussian Distribution ($m_K = 1$)

In this section, we demonstrate the performance of the proposed model using a single Gaussian distribution on standard binary and multiclass problems.

4.5.2.1 Synthetic Data

We start off by evaluating the performance of **GM-PR**, which uses a single Gaussian distribution for $p(\mathbf{x}|z^m)$, on synthetic data. We generate a two-cluster toy example to mimic the example in Fig. 4.1, which is motivated by [28]. The correct decision boundary should be the horizontal line along the x-axis. Fig. 4.3(a) is the generated data with the initial means. Fig. 4.3(b) is the clustering result obtained from an unconstrained **GMM**. Fig. 4.3(c) shows that the proposed **GM-PR** can learn the desired model with only two must-link relations and two cannot-link relations. Fig. 4.3(d) shows that the proposed **GM-PR** can learn the

https://github.com/gnaixgnaw/CSP



(a)

(b)

(c)

(d)

(e)

Fig. 4.3. **Application-specific model synthetic data**: (a) Original data with initial two means marked by x. Results are represented as follows: (b) **GMM**, (c) **GM-PR** using two *must-links* (solid line) and two *cannot-links* (dashed line), (d) **GM-PR** using only two must-links, and (e) **GM-PR** using only two cannot-links. The saturation of the red/green points represents the value of the soft label.

desired model with only two must-links. Fig. 4.3(e) shows that the proposed **GM-PR** can learn the desired model with only two cannot-link relations. This experiment illustrates the advantage of the proposed method, which can perform well with only either must-links or cannot-links. This advantage makes the proposed model distinct from previous works [134], [143]

4.5.2.2 UCI Repository and Handwritten Digits

In this section, we report the performance of three real datasets: 1) the **Haberman's survival** dataset contains 306 instances, 3 attributes, and 2 classes; 2) the **MNIST** database contains images of handwritten digits. We used the test dataset, which contains 10000 examples, 784 attributes, and 10 classes [171]; and 3) the **Thyroid** dataset contains 215 instances, 5 attributes, and 2 classes.

We demonstrate the performance of **GM-PR** on two binary clustering tasks, Haberman and Thyroid, and two multiclass problems, digits 1, 2, 3 and 4, 5, 6, 7. For ease of visualization, we work with only the leading two principal components of the MNIST using principal component analysis (PCA). Fig. 4.4 shows two-dimensional inputs, color-coded by class label. Fig. 4.5 shows that **GM-PR** significantly outperforms **GMM-EC** regardless of the available number of links on all datasets. Moreover, Fig. 4.6 shows that **GM-PR** performs well even if only the **must-links** are available. Compared to **EM-PC**, which uses only the *cannot-links*, Fig. 4.7 shows the performance of **GM-PR** is always greater than or comparable to **EM-PC**. Fig. 4.7 also shows that the performance of **EM-PC** decreases when the number of classes increases. Notice that all the experiments indicate that **GM-PR** has a lower variance over 100 random initializations, which implies **GM-PR** stability regardless of the number of available pairwise links.

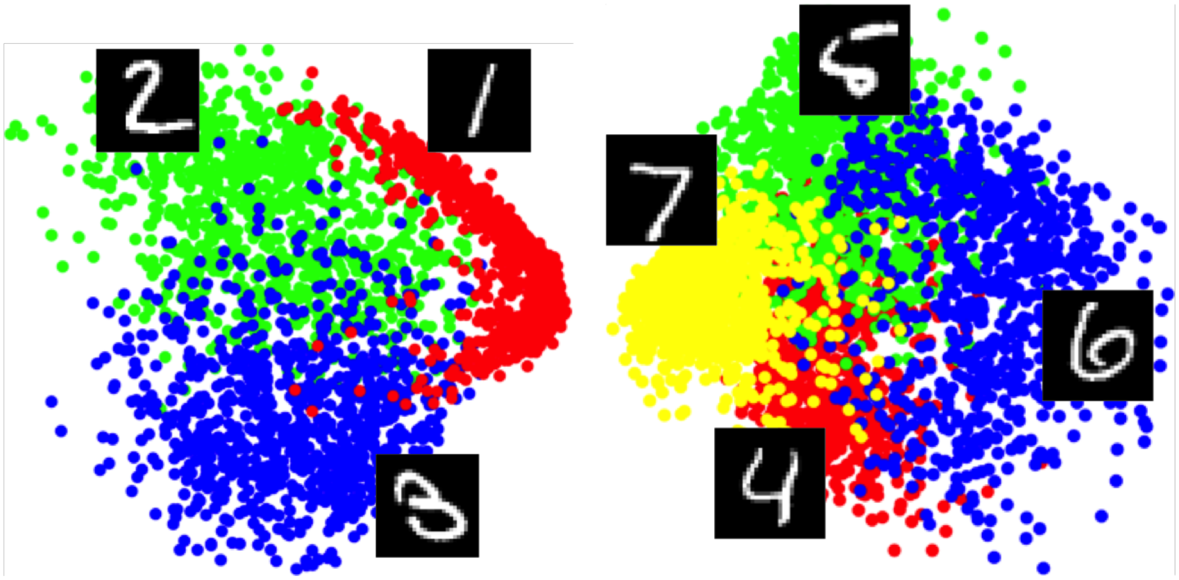
4.5.3 Results: Mixture of Gaussians ($m_K > 1$)

In this section, we demonstrate the performance of the proposed model using a mixture of Gaussians on the datasets that have local manifold structure.

https://archive.ics.uci.edu/ml/datasets.html

http://yann.lecun.com/exdb/mnist/

http://www.raetschlab.org/Members/raetsch/benchmark



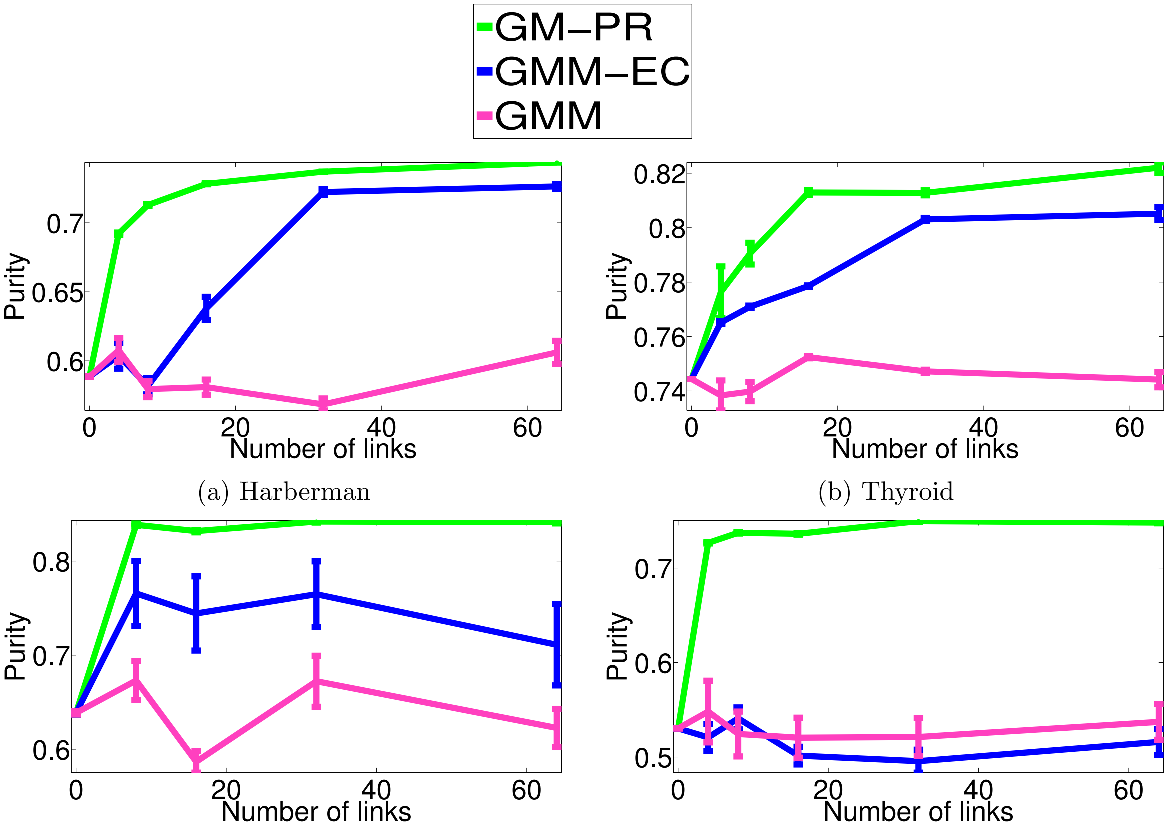
(a) digits 1, 2, and 3

(b) digits 4, 5, 6, and 7

Fig. 4.4. **Visualization of MNIST:** Digits 1, 2, and 3, and digits 4, 5, 6, and 7 visualized by the first two principal components of PCA.

4.5.3.1 Synthetic Data: Two Moons Dataset

Data points in two moons are on a moon-like manifold structure (Fig. 4.8(a)), which allows us to show the advantage of the proposed method using a mixture of Gaussians as a distribution instead of a single Gaussian distribution. Fig. 4.8(a) shows the data with initial means for the **GMM** and the **GM-PR** using a single Gaussian. Fig. 4.8(b) shows the data with initial means for **GM-PR** using a mixture of Gaussians ($m_K = 2$). Fig. 4.8(c) is the clustering result obtained from the unconstrained **GMM**, in which three points were assigned to the wrong class. Fig. 4.8(c) also shows that the performance of the **GMM** relied on the parameter initialization. Fig. 4.8(d) shows that the proposed **GM-PR**, which used a single cluster for each class, tried to learn the manifold structure via two must-link and two cannot-link relations. However, two points were still assigned to the incorrect class. Fig. 4.8(e) shows that the **GM-PR** can trace the manifold structure but used the same links in (d) with two clusters for each class. This experiment illustrates the advantage of the proposed model with a mixture of distributions that traces the local data structure by every single cluster and describes the global data structure using the mixture of clusters.



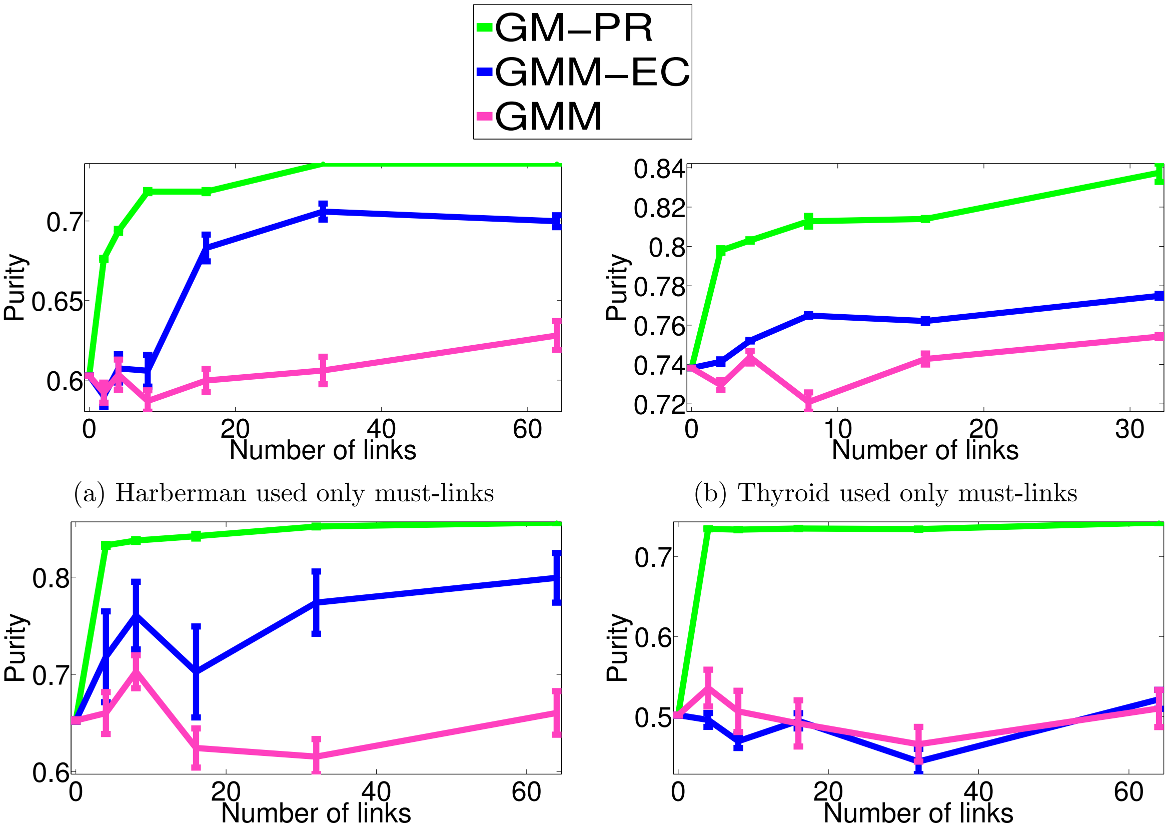
(a) Harberman

(b) Thyroid

(c) digits 1, 2, and 3

(d) digits 4, 5, 6, and 7

Fig. 4.5. **Result of MNIST and UCI:** The performance of **GM-PR** compared to **GMM-EC** [134] with a different number of pairwise links on (a) Harberman, (b) Thyroid, (c) digits 1, 2, and 3, and (d) digits 4, 5, 6, and 7.



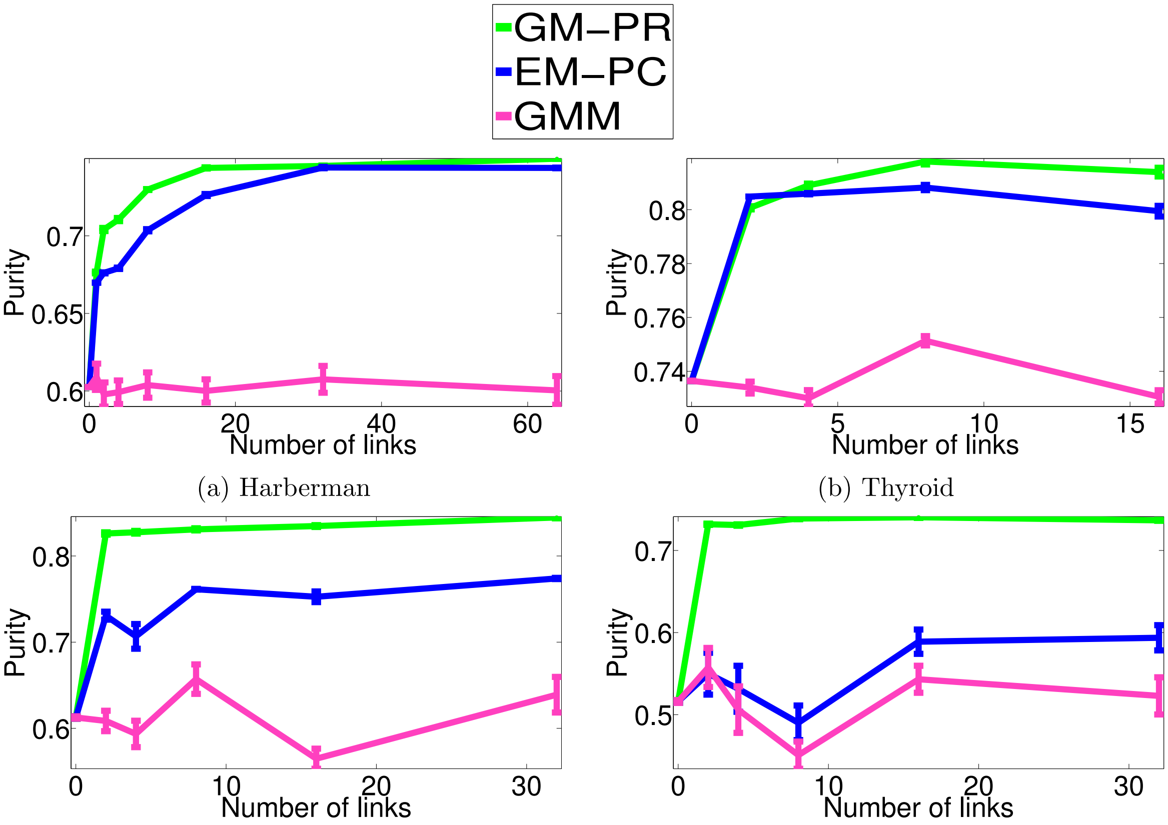
(a) Harberman used only must-links

(b) Thyroid used only must-links

(c) digits 1, 2, and 3

(d) digits 4, 5, 6, and 7

Fig. 4.6. **Result of MNIST and UCI with only must-link relations:** The performance of **GM-PR** compared to **GMM-EC** [134] with a different number of must-links on (a) Harberman, (b) Thyroid, (c) digits 1, 2, and 3, and (d) digits 4, 5, 6, and 7.



(a) Harberman

(b) Thyroid

(c) digits 1, 2, and 3

(d) digits 4, 5, 6, and 7

Fig. 4.7. **Result of MNIST and UCI with only cannot-link relations:** The performance of **GM-PR** compared to **EM-PC** [144] with a different number of cannot-links on (a) Harberman, (b) Thyroid, (c) digits 1, 2, and 3, and (d) digits 4, 5, 6, and 7.

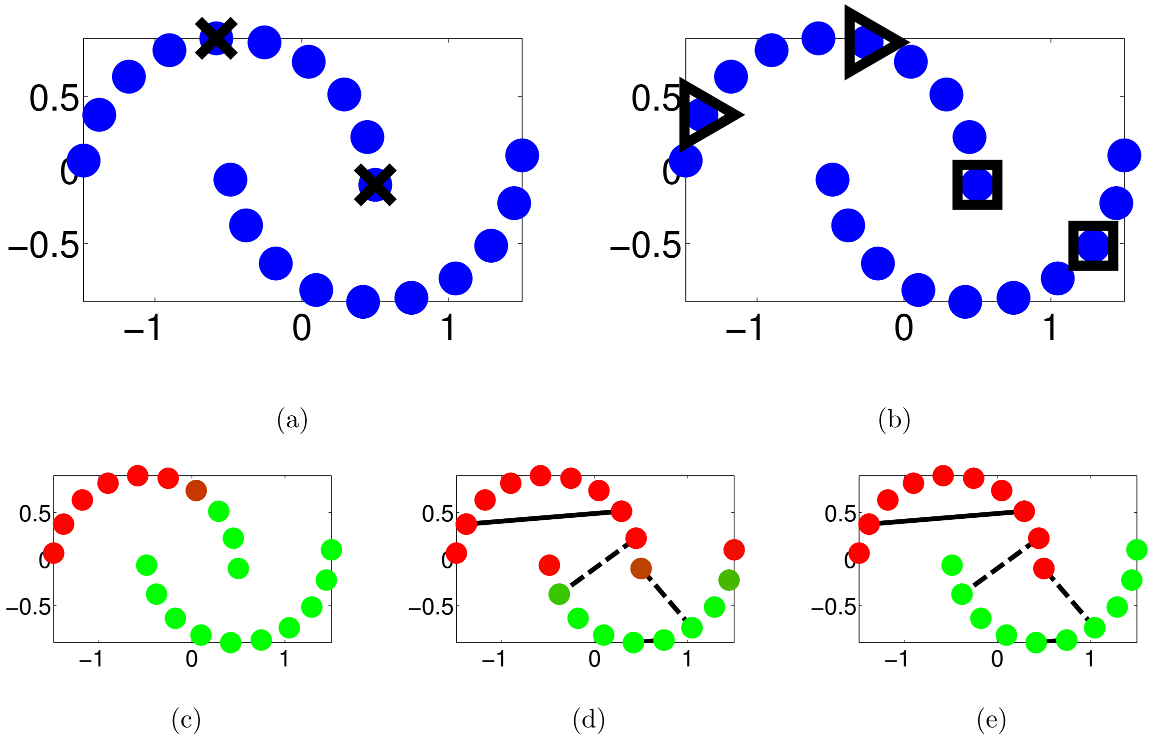


Fig. 4.8. **Two moons synthetic data**: (a) Original data with initial two means marked by x. (b) Original data with initial means marked by triangles for class 1 and squares for class 2. Results are represented as follows: (c) **GMM**, (d) **MM-PR** used one cluster for each class, and two *must-links* (solid line) and two *cannot-links* (dashed line), and (e) **MM-PR** used two clusters for each class and used the same links as in (d).

4.5.3.2 COIL 20

In this section, we report the performance of COIL 20 datasets, which contain images of 20 objects in which each object was placed on a turntable and rotated 360 degrees to be captured with different poses via a fixed camera (Fig. 4.9). The COIL 20 dataset contains 1440 instances and 1024 attributes. We set the number of multiclusters per class by cross-validation to $m_K = 2$. Previous studies have shown that the intrinsic dimension of many high-dimensional real-world datasets is often quite small ($d \leq 20$) [172], [173]; therefore, each image is first projected onto the low-dimensional subspace (d = 20). Fig. 4.9 shows that the **GM-PR** provides higher purity values compared to the **SSKK** and the **CSC** with number of links ≥ 100. In these experiments, we found that the proposed model can outperform the graph-based method with fewer links.

4.5.4 Result: Sensitivity to Number of Clusters Per Class

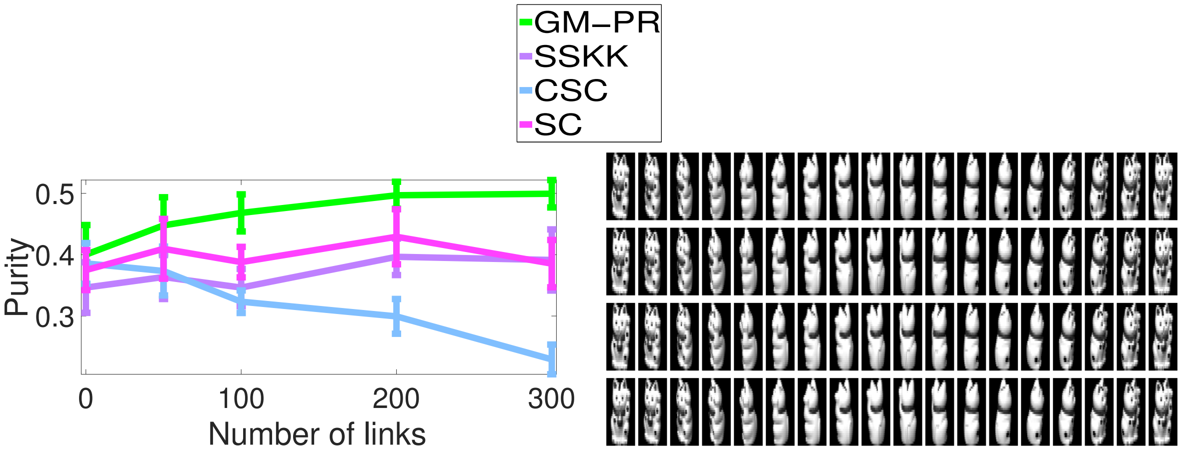
Lastly, we demonstrated the performance of the proposed model in regard to different values of m_K. First, we used the same dataset (MNIST) that is used in section 4.5.2.2. In Fig. 4.4(a), we observed digit 1, which clearly lay on a moon-like structure. Therefore, Fig. 4.10(a) shows that the performance of $m_K = 2, 3$, or 4 is better than $m_K = 1$ when the number of links is greater than 64. However, in Fig. 4.4(b), we observe hardly any manifold structure for digits 4, 5, 6, and 7. This observation also applies to the results in Fig. 4.10(b). The performances of $m_K = 1, 2, 3$, and 4 are very similar to each other, e.g., increasing the value of m_K does not help. However, we also notice that the increase in the number of m_K does not hurt the performance of the model and might even enhance the performance, depending on the dataset.

4.5.5 Application: Image Segmentation

In this subsection, we demonstrate the effectiveness of the proposed generative model for an application, image segmentation.

The goal of image segmentation is to simplify the representation of an image, an important preprocess for medical imaging analysis, image retrieval, or object tracking [174]. Therefore, the procedure in image segmentation is to partition image elements (e.g., pixel or voxel) into several different categories, i.e., a 2D image segmentation problem can be viewed as a pixel clustering problem. Different from the standard clustering problem, pixel clustering needs to ensure that spatial connectivity (i.e., pixel neighborhoods) is formulated

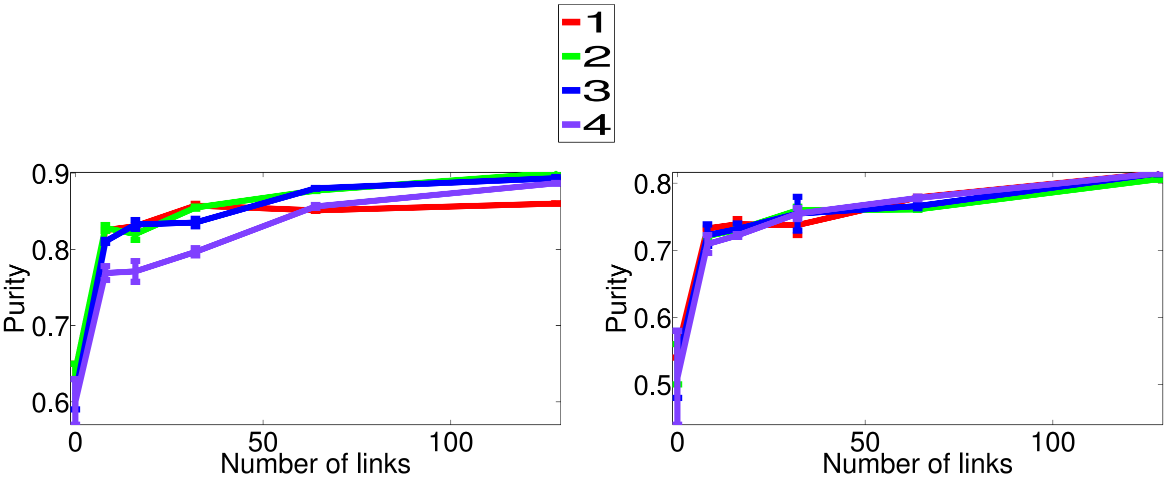
http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php



(a) result

(b) COIL-20

Fig. 4.9. **Result of COIL 20:** The performance of **GM-PR** compared to **SSKK** [141] and **CSC** [147] with a different number of links on COIL-20 (d).



(a) digits 1, 2, and 3

(b) digits 4, 5, 6, and 7

Fig. 4.10. **Result of sensitivity to number of clusters per class:** The performance of **GM-PR** uses different values of m_K on (a) digits 1, 2, and 3 and (b) digits 4, 5, 6, and 7.

during segmentation. Suppose we have a set of observed data P including every single pixel p and a pixel neighborhood set NB representing all pairs (p,q) in p. In order to model pixel neighborhoods, we can construct a graph with an image (e.g., pixels are associated with nodes and the edges) by

$$E(L) = \sum_{p \in P} D_p(L_p) + \sum_{p \in P} \sum_{(p,q) \in NB} V_{p,q}(L_p, L_q), \tag{4.22}$$

where $L = \{L_p | p \in P\}$ is a labeling of image P, and $D_p(\cdot)$ is a penalty function, which is defined by an individual label preference of pixels based on observed intensities given a pretrained likelihood function (e.g., the GMM and the proposed model). $V_{p,q}(\cdot)$ is the interaction potential, which indicates spatial coherence by penalizing discontinuities between neighboring pixels (e.g., Pottes model [175]). Hence, we can partition the vertices in the graph into disjoint subsets. The technique used to find the energy minimization in equation (4.22) is the max flow/min cut algorithm, which is an efficient solver for many low-level computer vision problems (e.g., image segmentation) that can be formulated in terms of energy minimization [175]–[179].

In this experiment, we consider the foreground and background segmentation ($M = 2$). We first trained the proposed model and then used it as $D_p(\cdot)$ in equation (4.22). The algorithm we used for solving the max flow problem is that proposed by [179]. The parameter initialization of the proposed model is the same as that in section 4.5.1. In image segmentation, instead of generating random pairwise relationships, we can also *manually* add some meaningful links to see if the results of the proposed model conform to our expectation. The intensity of the pixel is used for the feature vector. We used the images from the Berkeley segmentation dataset and benchmark. For efficiency, we compressed the images to 30% of the original size.

Fig. 4.11(a), (b) and (c) demonstrate that pairwise relations were created manually. We manually selected the two red blocks (size is 10x10) and added 100 pairwise relations between each pixel in the two blocks. The pairwise relations between each pixel in the two blocks can be either must-link relations or cannot-link relations. We set $m_K = 5$. In Fig. 4.11(a), when the pairwise relationship was a must-link relation, both grass and sky were recognized as background. However, when the pairwise relationship was cannot-link, only sky was recognized as background. Fig. 4.11(b) demonstrates that if the pairwise

http://vision.csd.uwo.ca/code/

https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

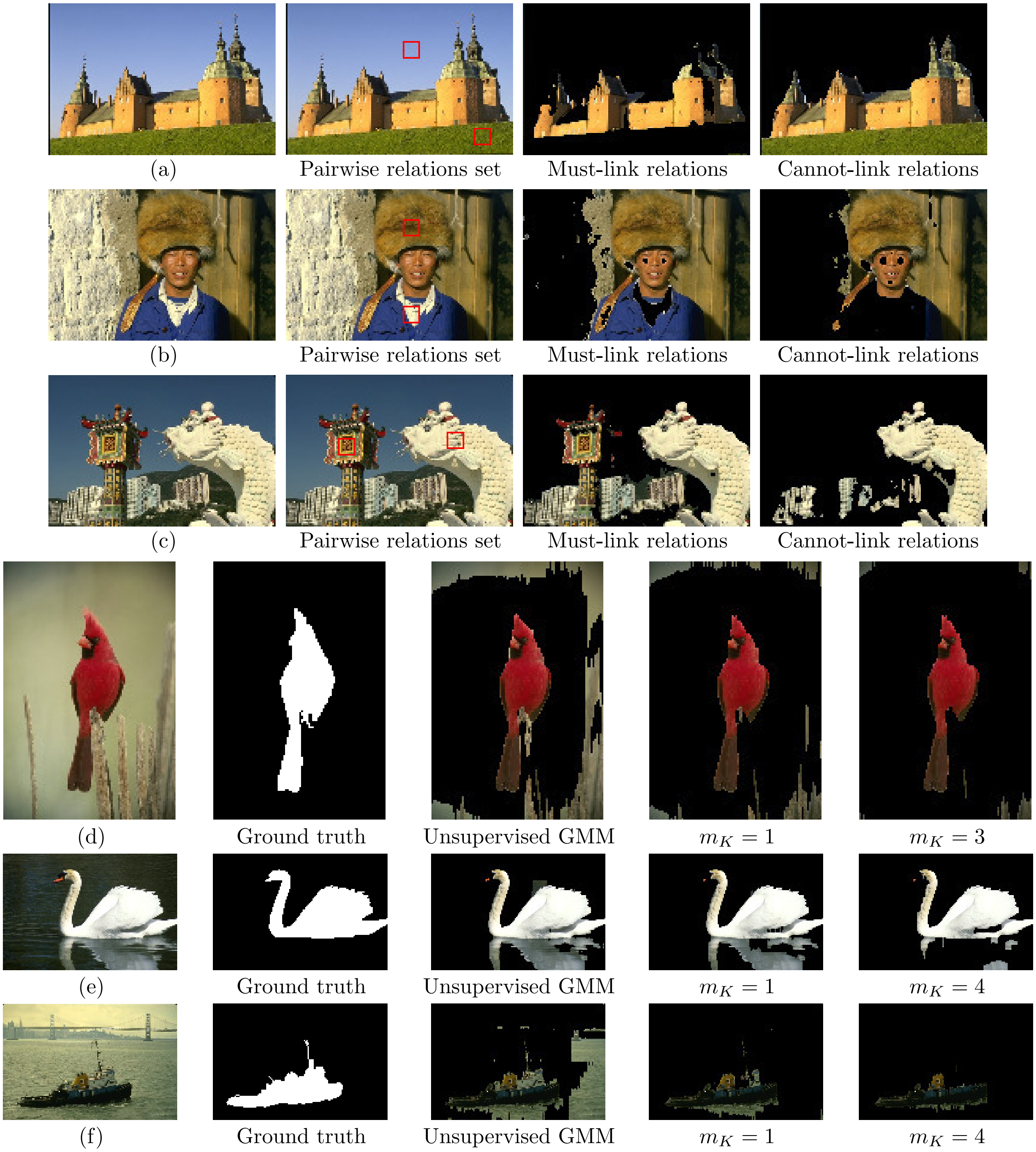


Fig. 4.11. **Results of image segmentation.** (a), (b), and (c) used the manually selected pairwise relations. (c), (d), and (e) created the pairwise relations by randomly selecting a pair of pixels.

relationship was the must-link relation, both hat and clothes were recognized as foreground. If, however, the pairwise relationship was cannot-link, clothes was recognized as background. Fig. 4.11(c) demonstrates that if the pairwise relationship was the must-link relation, both temple tower and statue were recognized as foreground. Only the statue was recognized as foreground if the pairwise relationship was cannot-link.

In Fig. 4.11(d), (e), and (f), pairwise relations were created by randomly selecting a pair of pixels. If the pixels have the same ground-truth labeling, we move them to the must-link set, otherwise, to the cannot-link set. We created 500 pairwise relations (i.e., around 5% of pixels were used as user-input) of which 250 relations were must-links and the others were cannot-links. We compared the proposed model, **GM-PR** with different values of m_K, to the unsupervised **GMM**. We set m_K by cross-validation. Fig. 4.11(d) shows that the task was to recognize the bird. The result proved that the unsupervised GMM can recognize the bird; however, many pixels in the background were still assigned to be foreground. The result further demonstrated that the **GM-PR** with $m_K = 1$ can remove more background pixels than the unsupervised GMM. When $m_K = 3$, we can recognize the bird more precisely. The performance of the **GM-PR** is similar in Fig. 4.11(e), where the swan can be segmented precisely when $m_K = 4$. Finally, in Fig. 4.11(f), **GM-PR** ($m_K = 1$ and 4) can segment the cargo ship in a manner that is more close to the ground-truth than unsupervised GMM.

4.6 Conclusion

This paper proposed a fully generative approach, **GM-PR**. In our formulation, cannot-link relationships also contribute to the **GM-PR** to increase the performance of clustering, which is distinct from previous works. We saw that in a distribution in a location-scale family (e.g., mixture of Gaussians), the updated parameters of **GM-PR** are in a closed form with an inexpensive, nonlinear optimization for the mixing parameters in terms of ease of implementation. Moreover, the statistical interpretation of pairwise relationships is more suited to the generative model, in which a pair of data is connected by *similarity* (e.g., similar or dissimilar) instead of either hard constraints or heuristic soft penalties. The results, therefore, demonstrated that the **GM-PR** can outperform the state-of-the-art with fewer pairwise relations. In addition, the result also demonstrated when $p(\mathbf{x}|z^m)$ is a mixture model, the **GM-PR** can discover the data on a manifold structure. When the data do not lie on a manifold structure, we found that different values of m_K do not hurt the performance of the model. Lastly, we showed the **GM-PR** can be used for image segmentation.

The **GM-PR** is a useful tool because it is in a generalized form that can be represented as any alternative distribution and model. Different from the mixture model, the hidden Markov model (HMM) is widely employed for analyzing biological sequences data and time sequential data. A semisupervised approach for clustering data in a sequential order is still an interesting problem. In the future, if the **GM-PR** uses the HMM, the **GM-PR** could be used to solve this type of problem. Furthermore, we can assume that a pair of data is generated from a different model. For example, images with text descriptions (e.g., tag) are very common in social media. Instead of modeling image and text in a single model, a more efficient approach is to use the **GM-PR** with the mixture model for images and HMM for text.

APPENDIX

EXPECTATION-MAXIMIZATION: MIXTURE OF GAUSSIANS WITH PAIRWISE RELATIONSHIPS

A.1 Likelihood: Must-link Relationships

The likelihood of the $\mathcal{S}(\cdot)$ is

$$\begin{aligned}&\mathcal{S}(\mathcal{X},\mathcal{Y},\mathcal{Z},\mathcal{M},\boldsymbol{\vartheta}) := p(\mathcal{X},\mathcal{Y},\mathcal{Z}|\mathcal{M},\boldsymbol{\vartheta})\\ &= \prod_{m=1}^{M}\prod_{m_k=1}^{m_K}\prod_{(i,j)\in\mathcal{M}}\left[\alpha_m\Big[\pi_{m_k}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_{m_k},\boldsymbol{\Sigma}_{m_k})\Big]^{y_i^{m_k}}\Big[\pi_{m_k}\mathcal{N}(\mathbf{x}_j|\boldsymbol{\mu}_{m_k},\boldsymbol{\Sigma}_{m_k})\Big]^{y_j^{m_k}}\right]^{z^m_{\{ij\}}}. \qquad (A.1)\end{aligned}$$

A.2 Likelihood: Cannot-link Relationships

The likelihood of the $\mathcal{D}(\cdot)$ is

$$\begin{aligned} p(z_a^m, z_b^m) &= p(z_a^m|z_b^m)p(z_b^m) := p(z_b^m|z_a^m)p(z_a^m) && (A.2)\\ &= \begin{cases} \dfrac{(\alpha_m)^{z_a^m}(\alpha_m)^{z_b^m}}{1-\sum_{m'=1}^{M}\alpha_{m'}^2} & z_a^m \neq z_b^m \\ 0 & z_a^m = z_b^m \end{cases}. && (A.3)\end{aligned}$$

and

$$\begin{aligned}&\mathcal{D}(\mathcal{X},\mathcal{Y},\mathcal{Z},\mathcal{C},\boldsymbol{\vartheta}) := p(\mathcal{X},\mathcal{Y},\mathcal{Z}|\mathcal{C},\boldsymbol{\vartheta})\\ &= \prod_{m=1}^{M}\prod_{m_k=1}^{m_K}\prod_{(a,b)\in\mathcal{C}}\Big[\pi_{m_k}\mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{m_k},\boldsymbol{\Sigma}_{m_k})\Big]^{z_a^m\; y_a^{m_k}}\Big[\pi_{m_k}\mathcal{N}(\mathbf{x}_b|\boldsymbol{\mu}_{m_k},\boldsymbol{\Sigma}_{m_k})\Big]^{z_b^m\; y_b^{m_k}} p(z_a^m, z_b^m).\\ &\qquad (A.4)\end{aligned}$$

A.3 E-Step: Unsupervised Scenario

The expatiation $\mathcal{L}(\cdot)$ is

$$\begin{aligned}\mathbb{E}_{z_n^m, y_n^{m_k}|\mathbf{x}_n}[z_n^m\; y_n^{m_k}] &= p(z_n^m, y_n^{m_k}|\mathbf{x}_n) \qquad (A.5)\\ &= \frac{\alpha_m\pi_{m_k}\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{m_k},\boldsymbol{\Sigma}_{m_k})}{\sum_{m'=1}^{M}\sum_{m'_k=1}^{m_K}\alpha_{m'}\pi_{m'_k}\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{m'_k},\boldsymbol{\Sigma}_{m'_k})},\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}_{z_n^m|\mathbf{x}_n}[z_n^m] &= p(z_n^m|\mathbf{x}_n)\\ &= \frac{\alpha_m \sum_{m_k=1}^{m_K} \pi_{m_k}\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{m_k}, \boldsymbol{\Sigma}_{m_k})}{\sum_{m'=1}^{M}\sum_{m'_k=1}^{m_K} \alpha_{m'}\pi_{m'_k}\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_{m'_k}\boldsymbol{\Sigma}_{m'_k})}.\end{aligned} \tag{A.6}$$

A.4 E-Step: Must-link Scenario

The $\mathcal{S}(\cdot)$ is

$$\begin{aligned}&\mathbb{E}_{z_{\{ij\}}^m, y_i^{m_k}|\mathbf{x}_i,\mathbf{x}_j}[z_{\{ij\}}^m y_i^{m_k}] = p(z_{\{ij\}}^m, y_i^{m_k}|\mathbf{x}_i, \mathbf{x}_j)\\ &= \frac{\alpha_m \pi_{m_k}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_{m_k}, \boldsymbol{\Sigma}_{m_k}) \sum_{m_{k'}=1}^{m_K} \pi_{m_{k'}}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_{m_{k'}}, \boldsymbol{\Sigma}_{m_{k'}})}{\sum_{m'=1}^{M}\sum_{m'_k=1}^{m_K}\sum_{m'_{k'}=1}^{m_K} \alpha_{m'}\pi_{m'_k}\pi_{m'_{k'}}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_{m'_k}, \boldsymbol{\Sigma}_{m'_k})\mathcal{N}(\mathbf{x}_j|\boldsymbol{\mu}_{m'_{k'}}, \boldsymbol{\Sigma}_{m'_{k'}})},\end{aligned} \tag{A.7}$$

and

$$\begin{aligned}&\mathbb{E}_{z_{\{ij\}}^m|\mathbf{x}_i,\mathbf{x}_j}[z_{\{ij\}}^m] = p(z_{\{ij\}}^m|\mathbf{x}_i, \mathbf{x}_j)\\ &= \frac{\alpha_m \sum_{m_k=1}^{m_K}\sum_{m_{k'}=1}^{m_K} \pi_{m_k}\pi_{m_{k'}}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_{m_k}, \boldsymbol{\Sigma}_{m_k})\mathcal{N}(\mathbf{x}_j|\boldsymbol{\mu}_{m_{k'}}, \boldsymbol{\Sigma}_{m_{k'}})}{\sum_{m'=1}^{M}\sum_{m'_k=1}^{m_K}\sum_{m'_{k'}=1}^{m_K} \alpha_{m'}\pi_{m'_k}\pi_{m'_{k'}}\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_{m'_k}, \boldsymbol{\Sigma}_{m'_k})\mathcal{N}(\mathbf{x}_j|\boldsymbol{\mu}_{m'_{k'}}, \boldsymbol{\Sigma}_{m'_{k'}})}.\end{aligned} \tag{A.8}$$

A.5 E-Step: Cannot-link Scenario

The $\mathcal{D}(\cdot)$ is

$$\begin{aligned}&\mathbb{E}_{z_a^m, y_a^{m_k}|\mathbf{x}_a,\mathbf{x}_b}[z_a^m\ y_a^{m_k}]\\ &= p(z_a^m, y_a^{m_k}|\mathbf{x}_a, \mathbf{x}_b)\\ &= \frac{\pi_{m_k}\mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{m_k}, \boldsymbol{\Sigma}_{m_k}) \sum_{m'=1}^{M}\sum_{m'_k=1}^{m_K} \pi_{m'_k}\mathcal{N}(\mathbf{x}_b|\boldsymbol{\mu}_{m'_k}, \boldsymbol{\Sigma}_{m'_k})p(z_a^m, z_b^{m'})}{Z_{\mathcal{D}}},\end{aligned} \tag{A.9}$$

and

$$\begin{aligned}&\mathbb{E}_{z_a^m|\mathbf{x}_a,\mathbf{x}_b}[z_a^m]\\ &= p(z_a^m|\mathbf{x}_a, \mathbf{x}_b)\\ &= \frac{\sum_{m_k=1}^{m_K} \pi_{m_k}\mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{m_k}, \boldsymbol{\Sigma}_{m_k}) \sum_{m'=1}^{M}\sum_{m'_k=1}^{m_K} \pi_{m'_k}\mathcal{N}(\mathbf{x}_b|\boldsymbol{\mu}_{m'_k}, \boldsymbol{\Sigma}_{m'_k})p(z_a^m, z_b^{m'})}{Z_{\mathcal{D}}},\end{aligned} \tag{A.10}$$

where

$$\begin{aligned}&Z_{\mathcal{D}} =\\ &\quad \sum_{m''=1}^{M}\sum_{m'''=1}^{M}\sum_{m''_k=1}^{m_K}\sum_{m'''_k=1}^{m_K} \pi_{m''_k}\pi_{m'''_k}\mathcal{N}(\mathbf{x}_a|\boldsymbol{\mu}_{m''_k}, \boldsymbol{\Sigma}_{m''_k})\mathcal{N}(\mathbf{x}_b|\boldsymbol{\mu}_{m'''_k}, \boldsymbol{\Sigma}_{m'''_k})p(z_a^{m''}, z_b^{m'''}).\end{aligned} \tag{A.11}$$

A.6 M-Step

The mean and covariance in the kth cluster in the mth class are

$$\boldsymbol{\mu}_{m_k} = \frac{\sum_n \ell_n^{m_k}\mathbf{x}_n + \sum_{(i,j)\in\mathcal{M}} \big[s_i^{m_k}\mathbf{x}_i + s_j^{m_k}\mathbf{x}_j + \sum_{(a,b)\in\mathcal{C}} \big[d_a^{m_k}\mathbf{x}_a + d_b^{m_k}\mathbf{x}_b\big]}{\sum_n \ell_n^{m_k} + \sum_{(i,j)\in\mathcal{M}} \big[s_i^{m_k} + s_j^{m_k}\big] + \sum_{(a,b)\in\mathcal{C}} \big[d_a^{m_k} + d_b^{m_k}\big]}, \tag{A.12}$$

$$\boldsymbol{\Sigma}_{m_k} = \frac{\sum_n \ell_n^{m_k}\mathbf{S}_n^{m_k} + \sum_{(i,j)\in\mathcal{M}} \big[s_i^{m_k}\mathbf{S}_i^{m_k} + s_j^{m_k}\mathbf{S}_j^{m_k}\big] + \sum_{(a,b)\in\mathcal{C}} \big[d_a^{m_k}\mathbf{S}_a^{m_k} + d_b^{m_k}\mathbf{S}_b^{m_k}\big]}{\sum_n \ell_n^{m_k} + \sum_{(i,j)\in\mathcal{M}} \big[s_i^{m_k} + s_j^{m_k}\big] + \sum_{(a,b)\in\mathcal{C}} \big[d_a^{m_k} + d_b^{m_k}\big]}, \tag{A.13}$$

where

$$\begin{aligned} \ell_n^{m_k} &= p(z_n^m, y_n^{m_k}|\mathbf{x}_n), \\ s_i^{m_k} &= p(z_{\{ij\}}^m, y_i^{m_k}|\mathbf{x}_i, \mathbf{x}_j), \\ d_a^{m_k} &= p(z_a^m, y_a^{m_k}|\mathbf{x}_a, \mathbf{x}_b), \end{aligned} \tag{A.14}$$

and

$$\mathbf{S}_n^{m_k} = (\mathbf{x}_n - \boldsymbol{\mu}_{m_k})(\mathbf{x}_n - \boldsymbol{\mu}_{m_k})^T. \tag{A.15}$$

Because the mixing parameter for the cluster π_{m_k} satisfies the summation to one, the determination can be achieved by the Lagrange multiplier.

$$\mathcal{Q}_{\mathcal{J}} + \lambda\bigg(\sum_{m_k=1}^{m_K} \pi_{m_k} - 1\bigg) \tag{A.16}$$

λ is the Lagrange multiplier. Taking the derivative of equation (A.16) with respect to π_{m_k},

$$\frac{\sum_{n=1}^{N} \ell_n^{m_k} + \sum_{(i,j)\in\mathcal{M}} \big[s_i^{m_k} + s_j^{m_k}\big] + \sum_{(a,b)\in\mathcal{C}} \big[d_a^{m_k} + d_b^{m_k}\big]}{\pi_{m_k}} + \lambda = 0 \tag{A.17}$$

By taking the derivative of equation (A.16) with respect to λ and equal to zero, we then can get $\sum_{m_k=1}^{m_K} \pi_{m_k} = 1$ and use it to eliminate the λ in equation (A.17). The mixing parameter for the kth cluster in the mth mixture is given by

$$\pi_{m_k} = \frac{\sum_{n=1}^{N} \ell_n^{m_k} + \sum_{(i,j)\in\mathcal{M}} \big[s_i^{m_k} + s_j^{m_k}\big] + \sum_{(a,b)\in\mathcal{C}} \big[d_a^{m_k} + d_b^{m_k}\big]}{\sum_{m_k=1}^{m_K} \Big(\sum_{n=1}^{N} \ell_n^{m_k} + \sum_{(i,j)\in\mathcal{M}} \big[s_i^{m_k} + s_j^{m_k}\big] + \sum_{(a,b)\in\mathcal{C}} \big[d_a^{m_k} + d_b^{m_k}\big]\Big)} \tag{A.18}$$

Lastly, estimating the mixing parameters for mixture α_m is the same as in equation (4.19).

REFERENCES

[1] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. Int. Conf. on Pattern Recognition*, vol. 2, 2004, pp. 28–31.

[2] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Comput. Soc. Conf. Computer Vision and Pattern Recogntion*, vol. 2, 1999.

[3] M. Ouyang, W. J. Welsh, and P. Georgopoulos, "Gaussian mixture clustering and imputation of microarray data," *Bioinformatics*, vol. 20, no. 6, pp. 917–923, 2004.

[4] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "Smem algorithm for mixture models," *Neural Computation*, vol. 12, no. 9, pp. 2109–2128, 2000.

[5] P. E. Rossi and G. M. Allenby, "Bayesian statistics and marketing," *Marketing Sci.*, vol. 22, no. 3, pp. 304–328, 2003.

[6] F. Niu, C. Zhang, C. Ré, and J. Shavlik, "Elementary: Large-scale knowledge-base construction via machine learning and statistical inference," *Int. J. Semantic Web and Infor. Syst. (IJSWIS)*, vol. 8, no. 3, pp. 42–73, 2012.

[7] C. Bishop and J. Lasserre, "Generative or discriminative? Getting the best of both worlds," *Bayesian Statist.*, vol. 8, pp. 3–24, 2007.

[8] C. M. Bishop *et al.*, *Pattern Recognition and Machine Learning.* New York, NY: Springer, 2006, vol. 1.

[9] O. Chapelle, B. Schölkopf, A. Zien *et al.*, *Semi-supervised Learning.* Cambridge, MA: MIT Press, 2006, vol. 2.

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

[11] V. N. Vapnik and V. Vapnik, *Statistical Learning Theory.* New York, NY: John Wiley and Sons, 1998, vol. 2.

[12] T. S. Jaakkola, D. Haussler *et al.*, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Inform. Process. Syst.*, vol. 11. MIT; 1998, 1999, pp. 487–493.

[13] 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. Intell.*, vol. 23, no. 6, pp. 643–660, 2001.

[14] J. M. Bennett and J. A. Flueck, "An evaluation of major league baseball offensive performance models," *Amer. Statistician*, vol. 37, no. 1, pp. 76–82, 1983.

[15] S. T. Jensen, K. E. Shirley, and A. J. Wyner, "Bayesball: A Bayesian hierarchical model for evaluating fielding in major league baseball," *Ann. of Appl. Statist.*, pp. 491–520, 2009.

[16] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Computational Learning Theory.* ACM, 1998, pp. 92–100.

[17] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. 33rd Annu. Meeting Assoc. Computational Linguistics.* Association for Computational Linguistics, 1995, pp. 189–196.

[18] T. Mitchell, "The role of unlabeled data in supervised learning," in *Proc. 6th Int. Colloq. Cognitive Sci.* Citeseer, 1999, pp. 2–11.

[19] V. Sindhwani, P. Niyogi, and M. Belkin, "A co-regularization approach to semi-supervised learning with multiple views," in *Proc. of Int. Conf. on Mach. Learning Workshop Learning with Multiple Views.* Citeseer, 2005, pp. 74–79.

[20] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," *Int. Conf. Mach. Learning*, vol. 18, pp. 19–26, 2001.

[21] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Inform. Process. Syst.*, vol. 16, 2003, pp. 321–328.

[22] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. of Mach. Learning Res.*, vol. 7, pp. 2399–2434, 2006.

[23] O. Delalleau, Y. Bengio, and N. Le Roux, "Efficient non-parametric function induction in semi-supervised learning," in *Proc.10th Int. Workshop on Artificial Intelligence and Stat.*, 2005, pp. 96–103.

[24] K. Yu, V. Tresp, and D. Zhou, "Semi-supervised induction with basis functions," *Max Planck Institute Technical Report*, vol. 141, 2004.

[25] T. Joachims, "Transductive inference for text classification using support vector machines," in *Int. Conf. on Mach. Learning*, vol. 99, 1999, pp. 200–209.

[26] T. De Bie and N. Cristianini, "Convex methods for transduction," in *Advances in Neural Inform. Process. Syst.*, vol. 16, 2003, pp. 73–80.

[27] N. D. Lawrence and M. I. Jordan, "Semi-supervised learning via Gaussian processes," in *Advances in Neural Inform. Process. Syst.*, vol. 17, 2004, pp. 753–760.

[28] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis Lectures on Artificial Intell. and Mach. Learning*, vol. 3, no. 1, pp. 1–130, 2009.

[29] F. G. Cozman, I. Cohen, M. C. Cirelo *et al.*, "Semi-supervised learning of mixture models," in *Int. Conf., on Mach. Learning*, 2003, pp. 99–106.

[30] M. Loog, "Semi-supervised linear discriminant analysis through moment-constraint parameter estimation," *Pattern Recognition Letters*, vol. 37, pp. 24–31, 2014.

[31] T. Yang and C. E. Priebe, "The effect of model misspecification on semi-supervised classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2093–2103, 2011.

[32] C. Goodall, "Procrustes methods in the statistical analysis of shape," *J. Royal Stat. Soc.*, vol. 53, no. 2, pp. 285–339, 1991.

[33] D. Kendall, "Shape manifolds, Procrustean metrics, and complex projective spaces," *Bull. London Math. Soc.*, vol. 16, no. 2, pp. 81–121, 1984.

[34] F. L. Bookstein, *The Measurement of Biological Shape and Shape Change.* New York, NY: Springer Science & Business Media, 2013.

[35] J. Coughlan and S. Ferreira, "Finding deformable shapes using loopy belief propagation," in *Euro. Conf. Comp. Vis.*, 2002, pp. 453–468.

[36] L. Gu and T. Kanade, "A generative shape regularization model for robust face alignment," in *Euro. Conf. Comp. Vis.*, 2008, pp. 413–426.

[37] A. Rangarajan, J. Coughlan, and A. Yuille, "A Bayesian network framework for relational shape matching," in *Int. Conf. Comp. Vis.*, 2003, pp. 671–678.

[38] A. Neumann, "Graphical Gaussian shape models and their application to image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 3, pp. 316–329, 2003.

[39] S.-C. Zhu, "Embedding Gestalt laws in Markov random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1170–1187, 1999.

[40] B. Scholkopf, A. Smola, and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[41] K. Grauman and T. Darrell, "The pyramid match kernel: Efficient learning with sets of features," *J. of Mach. Learning Res.*, vol. 8, pp. 925–760, 2007.

[42] K. Carter, R. Raich, and A. Hero, "On local intrinsic dimension estimation and its applications," *IEEE Trans. Signal Proc.*, vol. 58, no. 2, pp. 650–663, 2010.

[43] D. de Ridder, O. Kuoropteva, O. Okun, M. Pietikainen, and R. Duin, "Supervised locally linear embedding," in *Proc. Int. Conf. Artificial Neural Networks*, 2003, pp. 333–341.

[44] M. Felsberg, S. Kalkan, and N. Krueger, "Continuous dimensionality characterization of image structures," *Image and Vision Computing*, vol. 27, no. 6, pp. 628–636, 2009.

[45] M. Hein and J.-Y. Audibert, "Intrinsic dimensionality estimation of submanifolds in R^d," in *Int. Conf. on Mach. Learning.*, 2005, pp. 289–296.

[46] M. Raginsky and S. Lazebnik, "Estimation of intrinsic dimensionality using high-rate vector quantization," in *Advances in Neural Inform. Process. Syst.*, 2005, pp. 1–8.

[47] J. Ahn, J. S. Marron, K. Muller, and Y.-Y. Chi, "The high-dimension, low-sample-size geometric representation holds under mild conditions," *Biometrika*, vol. 94, no. 3, pp. 760–766, 2007.

[48] J. Ah-Pine, "Normalized kernels as similarity indices," in *Proc. Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining*, vol. 2, 2010, pp. 362–373.

[49] N. Courty, T. Burger, and P. Marteau, "Geodesic analysis on the Gaussian RKHS hypersphere," in *Euro. Conf. Mach. Learn. Prac. Knowl. Disc. Data.*, vol. 1, 2012, pp. 299–313.

[50] M. Eigensatz, "Insights into the geometry of the Gaussian kernel and an application in geometric modeling," Master's thesis, Swiss Federal Institute of Technology, 2006.

[51] A. Graf, A. Smola, and S. Borer, "Classification in a normalized feature space using support vector machines," *IEEE Trans. Neural Networks*, vol. 14, no. 3, pp. 597–605, 2003.

[52] F. Nielsen and R. Bhatia, *Matrix Information Geometry.* New York, NY: Springer, 2013.

[53] S. Buss and J. Fillmore, "Spherical averages and applications to spherical splines and interpolation," *ACM Trans. Graph.*, vol. 20, no. 2, pp. 95–126, 2001.

[54] K. Krakowski, K. Huper, and J. Manton, "On the computation of the Karcher mean on spheres and special orthogonal groups," in *Proc. Workshop Robotics Mathematics*, 2007, pp. 1–6.

[55] T. Fletcher, C. Lu, S. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Trans. Med. Imag.*, vol. 23, no. 8, pp. 995–1005, 2004.

[56] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-Euclidean metrics for fast and simple calculus on diffusion tensors," *Mgn. Reson. Med.*, vol. 56, no. 2, pp. 411–421, 2006.

[57] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos, "Jensen-Bregman logDet divergence with application to efficient similarity search for covariance matrices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2161–2174, 2012.

[58] S. Sommer, F. Lauze, S. Hauberg, and M. Nielsen, "Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations," in *Proc. Euro. Conf. Computer Vision*, 2010, pp. 43–56.

[59] D. C. Hoyle and M. Rattray, "Limiting form of the sample covariance eigenspectrum in PCA and kernel PCA," in *Int. Conf. Neural Info. Proc. Sys.*, 2003.

[60] P. Bühlmann and S. Van De Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications.* New York, NY: Springer, 2011.

[61] S. Berman, "Isotropic Gaussian processes on the Hilbert sphere," *Ann. of Probability*, vol. 8, no. 6, pp. 1093–1106, 1980.

[62] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Cambridge, MA: Academic Press, 1986, vol. 120.

[63] S. Kakutani *et al.*, "Topological properties of the unit sphere of a Hilbert space," *Proc. of the Imperial Academy*, vol. 19, no. 6, pp. 269–271, 1943.

[64] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society*, vol. B, no. 39, pp. 1–38, 1977.

[65] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Trans. on Comput.*, no. 5, pp. 401–409, 1969.

[66] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Sci.*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[67] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *J. of Mach. Learning Res.*, vol. 6, pp. 1783–1816, 2005.

[68] C. Walder and B. Schölkopf, "Diffeomorphic dimensionality reduction," in *Advances in Neural Inform. Process. Syst.*, vol. 21, 2008, pp. 1713–1720.

[69] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. of Mach. Learning Res.*, vol. 1, pp. 211–244, 2001.

[70] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *J. of Mach. Learning Res.*, vol. 3, pp. 1–48, 2003.

[71] B. Scholkopft and K.-R. Mullert, "Fisher discriminant analysis with kernels," in *Proc. 1999 IEEE Signal Processing Society Workshop Neural Networks for Signal Processing*, 1999, pp. 23–25.

[72] J. Wang, J. Lee, and C. Zhang, "Kernel trick embedded Gaussian mixture model," in *Proc. Algorithmic Learning Theory*, 2003, pp. 159–174.

[73] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 20, no. 1, pp. 39–51, 1998.

[74] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. of the Royal Stat. Soc.: Series B (Methodological)*, pp. 1–38, 1977.

[75] M. Jamshidian and R. I. Jennrich, "Acceleration of the em algorithm by using quasi-newton methods," *J. of the Royal Stat. Soc.: Series B (Statistical Methodology)*, vol. 59, no. 3, pp. 569–587, 1997.

[76] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models.* New York, NY: Springer, 1998, pp. 355–368.

[77] C. J. Wu, "On the convergence properties of the em algorithm," *Ann. of Statist.*, pp. 95–103, 1983.

[78] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[79] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods.* Cambridge, England: Cambridge University Press, 2000.

[80] J. Mercer, “Functions of positive and negative type, and their connection with the theory of integral equations,” *Philosoph. Trans. of the Royal Soc. of London. Series A, Containing Papers of a Mathematical or Physical Character*, pp. 415–446, 1909.

[81] M. Aizerman, E. Braverman, and L. Rozonoer, “Probability problem of pattern recognition learning and potential functions method,” *Avtomat. i Telemekh*, vol. 25, no. 9, pp. 1307–1323, 1964.

[82] N. Aronszajn, “Theory of reproducing kernels,” *Trans. of the Amer. Math. Soc.*, pp. 337–404, 1950.

[83] F. Girosi, “An equivalence between sparse approximation and support vector machines,” *Neural Computation*, vol. 10, no. 6, pp. 1455–1480, 1998.

[84] S. Saitoh, *Integral transforms, reproducing kernels and their applications.* Boca Raton, FL: CRC Press, 1997, vol. 369.

[85] G. Wahba, *Spline models for observational data.* SIAM, 1990.

[86] N. Aronszajn, “Theory of reproducing kernels,” *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, 1950.

[87] G. Blanchard, O. Bousquet, and L. Zwald, “Statistical properties of kernel principal component analysis,” *Mach. Learning*, vol. 66, no. 3, pp. 259–294, 2007.

[88] S. Amari and H. Nagaoka, *Methods of Information Geometry.* New York, NY: Oxford Univ. Press, 2000.

[89] M. Berger, *A Panoramic View of Riemannian Geometry.* New York, NY: Springer, 2007.

[90] R. Bhattacharya and V. Patrangenaru, “Large sample theory of intrinsic and extrinsic sample means on manifolds. I.” *Annals Statist.*, vol. 31, no. 1, pp. 1–29, 2005.

[91] ——, “Large sample theory of intrinsic and extrinsic sample means on manifolds. II.” *Ann Statist.*, vol. 33, no. 3, pp. 1225–1259, 2005.

[92] B. Afsari, “Riemannian L^p center of mass: Existence, uniqueness, and convexity,” *Proc. Am. Math. Soc.*, vol. 139, no. 2, pp. 655–673, 2011.

[93] H. Karcher, “Riemannian center of mass and mollifier smoothing,” *Comm. Pure Appl. Math.*, vol. 30, no. 5, pp. 509–41, 1977.

[94] W. S. Kendall, “Probability, convexity and harmonic maps with small image I: uniqueness and fine existence,” *Proc. London Math. Soc.*, vol. 61, pp. 371–406, 1990.

[95] B. Afsari, R. Tron, and R. Vidal, “On the convergence of gradient descent for finding the Riemannian center of mass,” *SIAM J. Control and Optimization*, vol. 51, no. 3, pp. 2230–2260, 2013.

[96] B. Charlier, “Necessary and sufficient condition for the existence of a Frechet mean on the circle,” *ESAIM: Probability and Statist.*, vol. 17, pp. 635–649, 2013.

[97] J. Shawe-Taylor, C. Williams, N. Cristianini, and J. Kandola, “On the eigenspectrum of the Gram matrix and the generalisation error of kernel PCA,” *IEEE Trans. Info. Theory*, vol. 51, no. 7, pp. 2510–2522, 2005.

[98] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von Mises-Fisher distributions," *J. Mach. Learning Res.*, vol. 6, pp. 1345–1382, 2005.

[99] K. Mardia and P. Jupp, *Directional Statistics.* New York, NY: John Wiley and Sons, 2000.

[100] D. Peel, W. Whiten, and G. McLachlan, "Fitting mixtures of Kent distributions to aid in joint set identification," *J. Amer. Statist. Assoc.*, vol. 96, pp. 56–63, 2001.

[101] X. Pennec, "Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements," *J. Math. Imaging and Vision*, vol. 25, no. 1, pp. 127–154, 2006.

[102] A. Mas, "Weak convergence in the function autoregressive model," *J. Multiv. Anal.*, vol. 98, pp. 1231–1261, 2007.

[103] T. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theor. Comp. Sci.*, vol. 38, pp. 293–306, 1985.

[104] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop on Applications of Comput. Vision*, 1994, pp. 138–142.

[105] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[106] J. A. Lee and M. Verleysen, "Quality assessment of dimensionality reduction: Rank-based criteria," *Neurocomputing*, vol. 72, pp. 1432–1433, 2009.

[107] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[108] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *2009 IEEE Conf. Comput. Vision and Pattern Recognition.* IEEE, 2009, pp. 2004–2011.

[109] J. C. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *2007 IEEE Conf. Comput. Vision and Pattern Recognition.* IEEE, 2007, pp. 1–8.

[110] S. Joshi, R. Kommaraji, J. Phillips, and S. Venkatasubramanian, "Comparing distributions and shapes using the kernel distance," in *ACM Symp. Comp. Geom.*, 2011, pp. 47–56.

[111] M. Vaillant and J. Glaunes, "Surface matching via currents," in *Info. Proc. Med. Imag.*, vol. 19, 2005, pp. 381–92.

[112] S. Duane, A. Kennedy, B. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Phys. Lett.*, vol. 195, pp. 216–222, 1987.

[113] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models - their training and application," *Comp. Vis. Imag. Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[114] A. Kotcheff and C. Taylor, "Automatic construction of eigen shape models by direct optimization," *Med. Imag. Anal.*, vol. 2, no. 4, pp. 303–314, 1998.

[115] R. Davies, C. Twining, T. Cootes, and C. Taylor, “A minimum description length approach to statistical shape modeling,” *IEEE Trans. Med. Imag.*, vol. 21, no. 5, pp. 525–537, 2002.

[116] H. Thodberg, “MDL shape and appearance models,” in *Info. Proc. Med. Imag.*, vol. 2, 2003, pp. 251–260.

[117] J. Cates, P. T. Fletcher, M. Styner, H. Hazlett, and R. Whitaker, “Particle-based shape analysis of multi-object complexes,” in *Info. Proc. Med. Imag.*, 2008, pp. 477–485.

[118] I. Dryden and K. Mardia, *Statistical Shape Analysis.* New York, NY: John Wiley and Sons, 1998.

[119] B. K. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[120] LEAF, “Tree Leaf Database: Inst. of Information Theory and Automation ASCR, Prague, Czech Republic, http://zoi.utia.cas.cz/tree_leaves,” 2012.

[121] P. Persson and G. Strang, “A simple mesh generator in Matlab,” *SIAM Rev.*, vol. 46, no. 2, pp. 329–45, 2004.

[122] SCI Institute, 2013, ShapeWorks: An open-source tool for constructing compact statistical point-based models of ensembles of similar shapes that does not rely on specific surface parameterization. http://www.sci.utah.edu/software/shapeworks.html.

[123] D. Marcus, T. Wang, J. Parker, J. Csernansky, J. Morris, and R. Buckner, “Open access series of imaging studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults,” *J. Cogn. Neuro.*, vol. 19, no. 9, pp. 1498–1507, 2007.

[124] D. Zarpalas, P. Gkontra, P. Daras, and N. Maglaveras, “Gradient-based reliability maps for acm-based segmentation of hippocampus,” *IEEE Trans. Biomed. Eng.*, vol. 61, no. 4, pp. 1015–1026, 2014.

[125] ——, “Accurate and fully automatic hippocampus segmentation using subject-specific 3d optimal local maps into a hybrid active contour model,” *IEEE J. Transl. Eng. Health Med.*, vol. 2, pp. 1–16, 2014.

[126] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan, “A multimedia retrieval framework based on semi-supervised ranking and relevance feedback,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 723–742, 2012.

[127] A. Kyriakopoulou and T. Kalamboukis, “The impact of semi-supervised clustering on text classification,” in *Proc. 17th Panhellenic Conf. on Informatics.* ACM, 2013, pp. 180–187.

[128] T.-P. Nguyen and T.-B. Ho, “Detecting disease genes based on semi-supervised learning and protein–protein interaction networks,” *Artificial Intell. in Med.*, vol. 54, no. 1, pp. 63–71, 2012.

[129] K. Sirts and S. Goldwater, “Minimally-supervised morphological segmentation using adaptor grammars,” *Trans. of the Assoc. for Computational Linguistics*, vol. 1, pp. 255–266, 2013.

[130] M. Le Nguyen and A. Shimazu, "A semi-supervised learning model for mapping sentences to logical forms with ambiguous supervision," *Data and Knowledge Eng.*, vol. 90, pp. 1–12, 2014.

[131] S. Basu, I. Davidson, and K. Wagstaff, *Constrained clustering: Advances in algorithms, theory, and applications.* Boca Raton, FL: CRC Press, 2008.

[132] X. Zhu and J. Lafferty, "Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning," in *Int. Conf. on Mach. Learning*, 2005, pp. 1052–1059.

[133] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl *et al.*, "Constrained k-means clustering with background knowledge," in *Int. Conf. on Mach. Learning*, vol. 1, 2001, pp. 577–584.

[134] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, "Computing Gaussian mixture models with em using equivalence constraints," *Advances in Neural Inform. Process. Syst.*, vol. 16, no. 8, pp. 465–472, 2004.

[135] Z. Lu and T. K. Leen, "Semi-supervised learning with penalized probabilistic clustering," in *Advances in Neural Inform. Process. Syst.*, 2004, pp. 849–856.

[136] E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in Neural Inform. Process. Syst.*, 2002, pp. 505–512.

[137] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a mahalanobis metric from equivalence constraints," *J. of Mach. Learning Res.*, vol. 6, no. 6, pp. 937–965, 2005.

[138] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in Neural Inform. Process. Syst.*, 2005, pp. 1473–1480.

[139] D. Cohn, R. Caruana, and A. McCallum, "Semi-supervised clustering with user feedback," *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, vol. 4, no. 1, pp. 17–32, 2003.

[140] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," in *Int. Conf. on Knowledge Discovery and Data Mining.* ACM, 2004, pp. 59–68.

[141] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," *Mach. Learning*, vol. 74, no. 1, pp. 1–22, 2009.

[142] K. Allab and K. Benabdeslem, "Constraint selection for semi-supervised topological clustering," in *Machine Learning and Knowledge Discovery in Databases.* New York, NY: Springer, 2011, pp. 28–43.

[143] M. H. Law, A. P. Topchy, and A. K. Jain, "Model-based clustering with probabilistic constraints," in *SDM.* SIAM, 2005, pp. 641–645.

[144] J. Graca, K. Ganchev, and B. Taskar, "Expectation maximization and posterior constraints," in *Advances in Neural Inform. Process. Syst*, 2007.

[145] G. S. Mann and A. McCallum, "Generalized expectation criteria for semi-supervised learning with weakly labeled data," *J. of Mach. Learning Res.*, vol. 11, pp. 955–984, 2010.

[146] J. Yi, L. Zhang, R. Jin, Q. Qian, and A. Jain, "Semi-supervised clustering by input pattern assisted pairwise similarity matrix completion," in *Int. Conf. on Mach. Learning*, 2013, pp. 1400–1408.

[147] X. Wang and I. Davidson, "Flexible constrained spectral clustering," in *Proc. of the 16th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining.* ACM, 2010, pp. 563–572.

[148] C. Xiong, D. Johnson, and J. J. Corso, "Spectral active clustering via purification of the k-nearest neighbor graph," in *Proc. European Conf. on Data Mining*, 2012.

[149] A. Gammerman, V. Vovk, and V. Vapnik, "Learning by transduction," in *Proc. 14th Conf. on Uncertainty in Artificial Intell.*, 1998, pp. 148–155.

[150] R. Fletcher, *Practical methods of optimization.* New York, NY: John Wiley & Sons, 2013.

[151] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables.* Courier Corporation, 1964, vol. 55.

[152] E. Coviello, G. R. Lanckriet, and A. B. Chan, "The variational hierarchical em algorithm for clustering hidden markov models," in *Advances in Neural Inform. Process. Syst.*, 2012.

[153] C. K. Williams, "A mcmc approach to hierarchical mixture modelling," in *Advances in Neural Inform. Process. Syst.*, 1999, pp. 680–686.

[154] N. Vasconcelos and A. Lippman, "Learning mixture hierarchies," in *Advances in Neural Inform. Process. Syst.* Citeseer, 1998, pp. 606–612.

[155] J. Goldberger and S. T. Roweis, "Hierarchical clustering of a mixture model," in *Advances in Neural Inform. Process. Syst.*, 2004, pp. 505–512.

[156] M. Meila and M. I. Jordan, "Learning with mixtures of trees," *J. of Mach. Learning Res.*, vol. 1, pp. 1–48, 2001.

[157] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994.

[158] M. K. Titsias and A. Likas, "Mixture of experts classification using a hierarchical mixture model," *Neural Computation*, vol. 14, no. 9, pp. 2221–2244, 2002.

[159] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Mach. Learning*, vol. 39, no. 2-3, pp. 103–134, 2000.

[160] J. Liu, D. Cai, and X. He, "Gaussian mixture model with local consistency," in *AAAI*, vol. 10, 2010, pp. 512–517.

[161] J. Shen, J. Bu, B. Ju, T. Jiang, H. Wu, and L. Li, "Refining Gaussian mixture model based on enhanced manifold learning," *Neurocomputing*, vol. 87, pp. 19–25, 2012.

[162] X. He, D. Cai, Y. Shao, H. Bao, and J. Han, "Laplacian regularized Gaussian mixture model for data clustering," *IEEE Trans. Knowl. Data. Eng.*, vol. 23, no. 9, pp. 1406–1418, 2011.

[163] X. Xing, Y. Yu, H. Jiang, and S. Du, "A multi-manifold semi-supervised gaussian mixture model for pattern classification," *Pattern Recognition Lett.*, vol. 34, no. 16, pp. 2118–2125, 2013.

[164] A. Demiriz, K. P. Bennett, and M. J. Embrechts, "Semi-supervised clustering using genetic algorithms," *Artificial Neural Networks in Eng.*, pp. 809–814, 1999.

[165] R. Dara, S. C. Kremer, D. Stacey *et al.*, "Clustering unlabeled data with soms improves classification of labeled real-world data," in *Proc. Int. Joint Conf. on Neural Networks*, vol. 3, 2002, pp. 2237–2242.

[166] A. B. Goldberg, X. Zhu, A. Singh, Z. Xu, and R. Nowak, "Multi-manifold semi-supervised learning," in *In Twelfth International Conference on Artificial Intelligence-and Statistics (AISTATS)*, 2009, pp. 169–176.

[167] A. Singh, R. Nowak, and X. Zhu, "Unlabeled data: Now it helps, now it doesn't," in *Advances in Neural Inform. Process. Syst.*, 2009, pp. 1513–1520.

[168] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. on Discrete Algorithms.* Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[169] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Comput. Sci.*, vol. 38, pp. 293–306, 1985.

[170] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval.* Cambridge, MA: Cambridge University Press, 2008, vol. 1.

[171] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[172] M. Raginsky and S. Lazebnik, "Estimation of intrinsic dimensionality using high-rate vector quantization," in *Advances in Neural Inform. Process. Syst.*, 2005, pp. 1105–1112.

[173] M. Felsberg, S. Kalkan, and N. Krüger, "Continuous dimensionality characterization of image structures," *Image and Vision Comput.*, vol. 27, no. 6, pp. 628–636, 2009.

[174] D. A. Forsyth and J. Ponce, "A modern approach," *Comput. Vision: A Modern Approach*, pp. 88–101, 2003.

[175] D. Greig, B. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *J. Royal Stat. Soc. Series B (Methodological)*, pp. 271–279, 1989.

[176] S. Roy and I. J. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem," in *Int. Conf. on Comput. Vision.* IEEE, 1998, pp. 492–499.

[177] H. Ishikawa, "Exact optimization for markov random fields with convex priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1333–1336, 2003.

[178] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, 2004.

[179] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, 2004.