Probabilistic Dimensionality Reduction via Structure Learning

Li Wang and Qi Mao

Abstract—We propose an alternative probabilistic dimensionality reduction framework that can naturally integrate the generative model and the locality information of data. Based on this framework, we present a new model, which is able to learn a set of embedding points in a low-dimensional space by retaining the inherent structure from high-dimensional data. The objective function of this new model can be equivalently interpreted as two coupled learning problems, i.e., structure learning and the learning of projection matrix. Inspired by this interesting interpretation, we propose another model, which finds a set of embedding points that can directly form an explicit graph structure. We proved that the model by learning explicit graphs generalizes the reversed graph embedding method, but leads to a natural interpretation from Bayesian perspective. This can greatly facilitate data visualization and scientific discovery in downstream analysis. Extensive experiments are performed that demonstrate that the proposed framework is able to retain the inherent structure of datasets and achieve competitive quantitative results in terms of various performance evaluation criteria.

Index Terms—Nonlinear dimensionality reduction, structure learning, probabilistic models, latent variable model

1 Introduction

Contemporary simulation and experimental data acquisition technologies enable scientists and engineers to generate progressively large and inherently high-dimensional data sampled from unknown multivariate probability distributions. Data expressed with many degrees of freedom imposes serious problems for data analysis. It is often difficult to directly analyze these datasets in the high-dimensional space, and is desirable to reduce the data dimensionality in order to overcome the curse of dimensionality and associate data with intrinsic structures for data visualization and subsequent scientific discovery.

Dimensionality reduction is a learning paradigm that transforms high-dimensional data into a low-dimensional representation, which can be of advantage in practical applications. Examples include clustering of gene expression data and text documents [1], and high-dimensional data visualization of image datasets [2], [3]. In general, the low-dimensional representations learned by dimensionality reduction methods are used for specific purposes and evaluated in terms of the performance criterion of related tasks, such as clustering or classification [4]. Similarly, the visualization results usually are expected to demonstrate the pattern of clusters, that is, points of the same class are close to each other, while points within different classes are distant. In addition, some methods are designed to preserve the locality information of data, such as preserving pairwise distances, so that rank-based measures [5] become useful which are based on distance rank errors and concepts such as neighborhood intrusions and extrusions.

From the methodology of modeling the problem of dimensionality reduction, existing methods can be divided into two categories: deterministic and probabilistic. Existing deterministic methods for dimensionality reduction have been proposed to capture certain information of data by catering for different criteria. Principal component analysis (PCA) [6] learns a subspace linearly spanned over some orthonormal bases by minimizing the reconstruction error [7]. Kernel PCA [8] first maps the input space to a reproducing kernel Hilbert space (RKHS) by a kernel function and then performs PCA in the RKHS space. Manifold learning [9] aims to find a manifold close to the intrinsic structure of data. By projecting data onto a manifold, the low-dimensional representation of data can be obtained by unfolding the manifold. Local linear embedding (LLE) [10] finds a mapping that preserves local geometry where local patches based on K-nearest neighbors are nearly linear and overlap with one another to form a manifold. Laplacian eigenmap (LE) [11] is proposed based on spectral graph theory where a K-nearest neighbor graph is constructed. Other methods related to neighborhood graphs are referred to survey papers [4], [9], including Isometric feature mapping (Isomap) [3], maximum variance unfolding (MVU) [12], diffusion maps (DM) [13], Hession LLE [14], and local tangent space analysis (LTSA) [15].

Probabilistic models have also been studied for dimensionality reduction. Probabilistic PCA (PPCA) [16] generalizes PCA by applying the latent variable model to the representation of linear relationship between data and its embeddings. Gaussian process latent variable model (GPLVM) [17] takes an alternative approach to marginalize the linear projection matrix, and then parametrizes covariance matrix using a kernel function. GPLVM with a linear kernel is the dual interpretation of PPCA, while its nonlinear generalization is related to KPCA. Bayesian GPLVM [18] maximizes the likelihood of data by marginalizing out both projection matrix and embeddings using variational inference. Maximum entropy unfolding (MEU) [19] is proposed to directly model the density of observed data by minimizing

- *L. Wang is with Department of Mathematics, University of Texas at Arlington, Texas, USA. E-mail: li.wang@uta.edu*
- *Q. Mao is a senior research engineer at HERE North America LLC, Chicago, USA. E-mail: qimao.here@gmail.com*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2785402, IEEE Transactions on Pattern Analysis and Machine Intelligence

Kullback-Leibler (KL) divergence under a set of constraints, and embedding points of data are obtained by maximizing the likelihood of the learned density. t-distributed stochastic neighbor embedding (tSNE) [20] employs a heavy-tailed distribution in the low-dimensional space to alleviate both the crowding problem and the optimization problem of SNE [21], which converts the high-dimensional Euclidean distances between data points into conditional probabilities that represent similarities.

In this paper, we are interested in developing methods that can naturally integrate information from both global and local perspectives of data for dimensionality reduction. For instance, the global information corresponds to the parameterized probability functions used to model the data generation in the above probabilistic models. In contrast, local information is related to the relationship between points in a small neighborhood as used in the above manifold learning methods. The integration of global and local information benefits dimensionality reduction with many good properties, including 1) the likelihood function and prior distribution can be used to model the noise of data for the robust learning; 2) a neighborhood graph is useful to capture the nonlinear representation of data if distances between data points are reliable in the neighborhood; 3) local information is also important to unveil a latent structure in a low-dimensional space.

Although the above existing methods work well under certain conditions, they lack the ability to learn robust embeddings from noisy data by applying neighborhood graph to capture the locality information. On the one hand, probabilistic models such as PPCA and GPLVM can deal with noisy data, but they are difficult to incorporate the neighborhood manifold, which has been proved to be effective for nonlinear dimensionality reduction. On the other hand, methods based on neighborhood manifold, such as MVU, LE and LLE, either are hard to learn the manifold structure of a smooth skeleton, or cannot be interpreted as probabilistic models for model selection and noise tolerance.

In addition to a given neighborhood graph, learning a latent graph structure from data becomes important if the locality information calculated in the original space is not reliable. Neighborhood graphs that are commonly used in graph based clustering and semi-supervised learning are the K-nearest neighbor graph and the ϵ-neighborhood graph [22]. Dramatic influences of these two graphs on clustering techniques have been studied in [23]. However, it is improper to use a fixed neighborhood size since the curvature of manifold and the density of data points may be different in different regions of the manifold [24]. Moreover, most distance-based manifold learning methods suffer from the curse of dimensionality, i.e., there is little difference in the distances between different pairs of data points [25]. Furthermore, if the data is noisy, a precomputed neighborhood graph to approximate the manifold of data is not reliable any more. As a result, it is less reliable to directly construct a neighborhood graph in a high-dimensional space.

To overcome the issues of constructing graphs, structure learning has had a great success in automatically learning explicit structures from data. A sparse manifold clustering and embedding (SMCE) [24] is proposed using ℓ_1 norm over the edge weights and ℓ_2 norm over the errors that measure the linear representation of every data point by using its neighborhood information. Similarly, ℓ_1 graph is learned for image analysis using ℓ_1 norm over both the edge weights and the errors for enhancing the robustness of the learned graph [26]. Instead of learning directed graphs by using the above two methods, an integrated model for learning an undirected graph by imposing the sparsity on a symmetric similarity matrix and a positive semidefinite constraint on the Laplacian matrix is proposed [27]. These are discriminative models, so they lack the ability to model noise of data. In addition to learning a general graph structure, a simple principal tree learning algorithm (SimplePPT) [28] aims to learn a spanning tree from data by minimizing data quantization error and the length of the tree. However, SimplePPT generates a principal tree in the input space, so it is not applicable for dimensionality reduction.

To take the advantages from the integrated modeling of both global and local information of data, we propose an alternative probabilistic dimensionality reduction framework that can naturally integrate the generative model and the locality information of data. This framework is formulated in terms of the empirical Bayesian inference. Based on the proposed framework, two different models are studied. One is the model with a given neighborhood graph, where the likelihood function models the data generation process using noise model and the loss function penalizes the violations of expected pairwise distances between two original data points and their corresponding embedded points in the given neighborhood. The other is inspired to learn a latent structure for dimensionality reduction, which learns a mapping function that transforms data points in a high-dimensional space to latent points in a low-dimensional space such that these latent points directly forms a graph. Extensive experiments are conducted to validate our proposed methods for the visualization of learned embeddings for latent graph structures and quantitative performance of various evaluation criteria. The main contributions of this paper are summarized as follows:

1) We propose an alternative probabilistic framework for dimensionality reduction, which not only takes the noise of data into account, but also utilizes the neighborhood graph as the locality information. To the best of our knowledge, there is no prior work that can model data generation error and pairwise distance constraints in a unified framework for dimensionality reduction.

2) We present a new model under the proposed framework using ℓ_2 loss function over the expected distances. Given a neighborhood graph, this model is able to learn a smooth skeleton structure of embedded points and retain the inherent structure from noisy data by imposing the shrinkage of the pairwise distances between data points.

3) Inspired by the interesting interpretation of the new model as two coupled learning problems, we propose another model, which finds a set of embedding points that can directly form an explicit graph structure. We proved that the model by learning explicit graphs generalizes the reversed graph embedding method [29], but leads to a natural interpretation from Bayesian perspective.

4) The connections between the proposed models and various existing methods are discussed including reversed graph embedding, MEU, and structure learning methods.

The rest of the paper is organized as follows. We first briefly introduce several existing methods from deterministic and probabilistic perspectives in Section 2. In Section 3, we propose a unified probabilistic framework for dimensionality reduction and a new model for learning a smooth skeleton from noisy, high-dimensional data. We further generalize this framework for learning an explicit graph structure in Section 4 and discuss the connections to various existing works in Section 5. Extensive experiments are conducted in Section 6. We conclude this work in Section 7. Proofs and more experimental results are given in the supplementary materials.

2 Related Work

Let $\mathbb{Y} = \{\mathbf{y}_i\}_{i=1}^N$ be a set of data points where $\mathbf{y}_i \in \mathbb{R}^D$. The goal of dimensionality reduction is to find a set of embedded data points $\mathbb{X} = \{\mathbf{x}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $d < D$, satisfying certain assumptions. Next, we briefly introduce several existing dimensionality reduction methods from both deterministic and probabilistic perspectives.

2.1 Deterministic Methods

The classic deterministic method for dimensionality reduction is MVU [12]. Its objective is to maximize the variance of the embedded points subject to constraints such that distances between nearby inputs are preserved. MVU consists of three steps. The first step is to compute the K-nearest neighbors $\mathcal{N}_i$ of data point $\mathbf{y}_i, \forall i$. The second step is to solve the following optimization problem

$$\max_{\mathbb{X}} \sum_{i=1}^N ||\mathbf{x}_i||^2 \tag{1}$$

$$\text{s.t. } ||\mathbf{x}_i - \mathbf{x}_j||^2 = ||\mathbf{y}_i - \mathbf{y}_j||^2, \forall i, j \in \mathcal{N}_i, \tag{2}$$

$$\sum_{i=1}^N \mathbf{x}_i = 0, \tag{3}$$

where constraints (2) preserve distances between K-nearest neighbors and constraint (3) eliminates the translational degree of freedom on the embedded data points by constraining them to be centered at the origin. Instead of optimizing over $\mathbb{X}$, MVU reformulates (1) as a semidefinite programming by learning a kernel matrix $\mathbf{K}$ with the (i,j)th element denoted by $\kappa_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle_{\mathcal{H}}$ with a semidefinite constraint $\mathbf{K} \succeq 0$ for a valid kernel [30] where the corresponding mapping function lies in a RKHS $\mathcal{H}$. Define $\phi_{i,j} = ||\mathbf{y}_i - \mathbf{y}_j||^2$ and $\zeta_{i,j} = ||\mathbf{x}_i - \mathbf{x}_j||^2 = \kappa_{i,i} + \kappa_{j,j} - 2\kappa_{i,j}$. The resulting semidefinite programming is

$$\max_{\mathbf{K}} \text{Tr}(\mathbf{K}) : \text{s.t.} \sum_{i,j} \kappa_{i,j} = 0, \mathbf{K} \succeq 0, \zeta_{i,j} = \phi_{i,j}, \forall i, j \in \mathcal{N}_i,$$

where $\langle \sum_{i=1}^N \mathbf{x}_i, \sum_{j=1}^N \mathbf{x}_j \rangle_{\mathcal{H}} = \sum_{i,j} \kappa_{i,j} = 0$ is a relaxation of (3) for the ease of kernelization. The last step is to obtain the embedding $\mathbb{X}$ by applying KPCA on the optimal $\mathbf{K}$. The distance/similarity information on a neighborhood graph is widely used in the manifold-based dimensionality reduction methods such as locally linear embedding (LLE) and its variants [10], and Laplacian Eigenmap (LE) [11].

A duality view of MVU problem has been studied in [31]. Define $\mathbf{E}^{i,j}$ as an $N \times N$ matrix consisting of only four nonzero elements: $\mathbf{E}^{i,j}[i,i] = \mathbf{E}^{i,j}[j,j] = 1, \mathbf{E}^{i,j}[i,j] = \mathbf{E}^{i,j}[j,i] = -1$. The preserving constraints can be rewritten as $\text{Tr}(\mathbf{K}\mathbf{E}^{i,j}) = \phi_{i,j}, \forall i, j \in \mathcal{N}_i$. Thus, the dual problem of the above semidefinite programming is given by

$$\min_{\{w_{i,j}\}} \sum_{i,j \in \mathcal{N}_i} w_{i,j}\phi_{i,j} : \text{s.t. } \lambda_{N-1}(\mathbf{L}) \geq 1, \mathbf{L} = \sum_{i,j \in \mathcal{N}_i} w_{i,j}\mathbf{E}^{i,j}, \tag{4}$$

where $w_{i,j}$ is the dual variable subject to the preserving constraint associated to edge (i,j), and λ_{N-1} denotes the second smallest eigenvalue of a symmetric matrix [31].

2.2 Probabilistic Models

Probabilistic models are able to take the noise model of data generation into consideration. The observed data $\mathbb{Y}$ and the embedding $\mathbb{X}$ are treated as random variables. For dimensionality reduction, we associate matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times d}$ to $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T \in \mathbb{R}^{N \times D}$.

Latent variable models for dimensionality reduction generally assume a linear relationship between $\mathbf{x}_i$ and $\mathbf{y}_i$ with noise given by

$$\mathbf{y}_i = \mathbf{W}\mathbf{x}_i + \boldsymbol{\epsilon}_i, \forall i, \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{D \times d}$ is a linear projection matrix, and $\boldsymbol{\epsilon}_i \in \mathbb{R}^D$ is the vector of noise values. Noise is independently sampled from a spherical Gaussian distribution with mean zero and covariance $\gamma^{-1}\mathbf{I}_D$ where $\gamma > 0$ and $\mathbf{I}_D$ is a $D \times D$ identity matrix. Thus, the likelihood of data point $\mathbf{y}_i$ is

$$p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{W}, \gamma) = \mathcal{N}(\mathbf{y}_i|\mathbf{W}\mathbf{x}_i, \gamma^{-1}\mathbf{I}_D), \tag{6}$$

and the likelihood of the whole data is $p(\mathbf{Y}|\mathbf{X}, \mathbf{W}, \gamma) = \prod_{i=1}^N p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{W}, \gamma)$ due to the independently and identically distributed (i.i.d.) assumption of data.

PPCA [16] further assumes that the latent variables $\{\mathbf{x}_i\}_{i=1}^N$ follow a unit covariance zero mean Gaussian distribution

$$\pi(\mathbf{x}_i) = \mathcal{N}(\mathbf{x}_i|\mathbf{0}_d, \mathbf{I}_d). \tag{7}$$

The projection matrix $\mathbf{W}$ is then obtained by maximizing the likelihood of the given data

$$\max_{\mathbf{W}} p(\mathbf{Y}|\mathbf{W}, \gamma) \tag{8}$$

where $p(\mathbf{Y}|\mathbf{W}, \gamma) = \prod_{i=1}^N p(\mathbf{y}_i|\mathbf{W}, \gamma)$ and the marginal likelihood of each data is obtained by marginalizing out the latent variable $\mathbf{X}$ given by

$$\begin{aligned} p(\mathbf{y}_i|\mathbf{W}, \gamma) &= \int p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{W}, \gamma)\pi(\mathbf{x}_i)d\mathbf{x}_i \\ &= \mathcal{N}(\mathbf{y}_i|\mathbf{0}, \mathbf{W}\mathbf{W}^T + \gamma^{-1}\mathbf{I}_D). \end{aligned} \tag{9}$$

Tipping and Bishop [16] showed that the principal subspace of the data is the optimal solution of problem (8) if γ approaches to infinity. Therefore, this model is viewed as a probabilistic version of PCA.

GPLVM [17] takes an alternative way to obtain marginal likelihood of data by marginalizing out $\mathbf{W}$ and optimizing with respect to $\mathbf{X}$. Assume the prior distribution of $\mathbf{W}$ as

$$\pi(\mathbf{W}) = \prod_{j=1}^D \mathcal{N}(\mathbf{w}_j|\mathbf{0}_d, \mathbf{I}_d), \tag{10}$$

where $\mathbf{w}_j$ is the jth row of $\mathbf{W}$. The marginal likelihood of the data is obtained by marginalizing out $\mathbf{W}$ given by

$$\begin{aligned} p(\mathbf{Y}|\mathbf{X}, \gamma) &= \int \prod_{i=1}^N p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{W}, \gamma)\pi(\mathbf{W})d\mathbf{W} \\ &= \mathcal{MN}_{N,D}(\mathbf{0}, \widehat{\mathbf{K}}, \mathbf{I}_D) \end{aligned} \tag{11}$$

where $\mathcal{MN}_{N,D}$ is the matrix normal distribution with zero mean, sample-based covariance matrix $\widehat{\mathbf{K}} = \mathbf{X}\mathbf{X}^T + \gamma^{-1}\mathbf{I}_N$ and feature-based covariance matrix $\mathbf{I}_D$ [32]. GPLVM obtains $\mathbf{X}$ by maximizing the marginal likelihood of the data

$$\max_{\mathbf{X}} p(\mathbf{Y}|\mathbf{X},\gamma). \tag{12}$$

Lawrence [17] showed that the optimal solution is equivalent to that obtained by PCA. The merit of this model is that different covariance functions can be incorporated for nonlinear representations since $\widehat{\mathbf{K}}$ is of the form of inner product matrix. Thus, the linear model is called the dual interpretation of PPCA [16], and the nonlinear model is related to KPCA [8].

3 Structured Dimensionality Reduction

We propose an alternative dimensionality reduction framework based on the regularized empirical Bayesian inference [33], where the unknown embedded data is not only decided by the observed data, but also regulated by manifold structures. Next, we first present the regularized empirical Bayesian inference, and expectation constraints for capturing manifold structure are then presented. With these two ingredients, we formulate the proposed framework for dimensionality reduction.

3.1 Regularized Empirical Bayesian Inference

Regularized empirical Bayesian inference is an optimization formulation of a richer type of posterior inference, by replacing the standard normality constraint with a wide spectrum of knowledge-driven and/or data-driven constraints or regularization. Following the notation in [33], denote $\mathcal{M}$ as the space of feasible models, which is a complete separable metric space endowed with its Borel σ-algebra $\mathcal{B}(\mathcal{M})$, and $\mathbf{M} \in \mathcal{M}$ represents an atom in this space. Moreover, denote Π as a distribution on the measurable space $(\mathcal{M}, \mathcal{B}(\mathcal{M}))$. We assume that Π is absolutely continuous with respect to some background measure μ, so that there exists a density π such that $\mathrm{d}\Pi = \pi \mathrm{d}\mu$. Given a model, let $\mathcal{D}$ be a collection of observed data points, which are assumed to be i.i.d. Define $\mathrm{KL}(q(\mathbf{M})||\pi(\mathbf{M})) = \int_{\mathcal{M}} q(\mathbf{M}) \log(q(\mathbf{M})/\pi(\mathbf{M}))\mathrm{d}\mu(\mathbf{M})$ as the Kullback-Leibler (KL) divergence from $q(\cdot)$ to $\pi(\cdot)$.

In the presence of unknown parameters, e.g., hyperparameters, empirical Bayesian inference is necessary where an estimation procedure such as maximum likelihood estimation is needed. Here, we focus on the expectation constraints, of which each one is a function of $q(\mathbf{M})$ through an expectation operator. For example, let $\boldsymbol{\psi} = \{\psi_1, \ldots, \psi_T\}$ be a set of feature functions, each of which is $\psi_t(\mathbf{M}; \mathcal{D})$ defined on $\mathcal{M}$ and possibly data dependent. With unknown parameter $\boldsymbol{\Theta}$, regularized empirical Bayesian inference is formulated by solving the following optimization problem

$$\begin{aligned} \inf_{\boldsymbol{\Theta}, q(\mathbf{M})} \; & \mathrm{KL}(q(\mathbf{M})||\pi(\mathbf{M})) - \int_{\mathcal{M}} \log p(\mathcal{D}|\mathbf{M},\boldsymbol{\Theta}) q(\mathbf{M}) \mathrm{d}\mu(\mathbf{M}) \\ & + U(\{\mathbb{E}_{q(\mathbf{M})}[\psi_t(\mathbf{M};\mathcal{D})]\}_{t=1}^T) \\ \text{s.t. } & q(\mathbf{M}) \in \mathcal{P}_{prob}, \boldsymbol{\Theta} \in \Theta, \end{aligned} \tag{13}$$

where $\mathbb{E}_{q(\mathbf{M})}[\psi_t(\mathbf{M};\mathcal{D})]$ is the expectation of $\psi_t(\mathbf{M};\mathcal{D})$ over $q(\mathbf{M})$, U is a function of $\{\mathbb{E}_{q(\mathbf{M})}[\psi_t(\mathbf{M};\mathcal{D})]\}_{t=1}^T$, Θ is the feasible set of the unknown parameter $\boldsymbol{\Theta}$, and $\mathcal{P}_{prob}$ is a subspace of distributions. Note that minimizing the first two terms of (13) with respect to $q(\mathbf{M})$ and $\boldsymbol{\Theta}$ leads to an optimal solution $q(\mathbf{M}) \propto \pi(\mathbf{M}) p(\mathcal{D}|\mathbf{M}, \boldsymbol{\Theta}^*)$ and $\boldsymbol{\Theta}^*$, which is equivalent to the maximum likelihood estimation,

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}\in\Theta} \log p(\mathcal{D}|\boldsymbol{\Theta}) = \log \int_{\mathcal{M}} \pi(\mathbf{M}) p(\mathcal{D}|\mathbf{M},\boldsymbol{\Theta}) \mathrm{d}\mu(\mathbf{M}).$$

Hence, problem (13) is called the regularized empirical Bayesian inference where the regularization term is useful to capture domain knowledge or structure information of data.

3.2 Expectation Constraints over Pairwise Distances

Expectation constraints are widely employed for classification problems in the generalized maximum entropy model [34] and regularized Bayesian inference model [33]. Here, we are particularly interested in defining expectation constraints for dimensionality reduction.

Given a probabilistic density function $q(\mathbf{M})$, the definition of feature function over data is one of the necessary ingredients to form an expectation constraint. For classification, the feature-label pair of one instance is naturally treated as ψ_t. However, this is not suitable for dimensionality reduction, where features from a single instance are not enough to determine the embedding of the whole data. As discussed before, most discriminant methods take pairwise distance as the key information provided by the data. For example, MVU [12] takes pairwise distances over a neighborhood graph, and LE [11] transforms pairwise distances over a neighborhood graph to similarity. Thus, pairwise distance can be reasonably considered as the factor of the feature function for dimensionality reduction. Specifically, in this paper, the feature function $\psi_{i,j}$ represents the difference between pairwise distance of embedding points $\mathbf{x}_i$ and $\mathbf{x}_j$ and their corresponding distance $\phi_{i,j}$, i.e., $\psi_{i,j}(\mathbf{X};\mathbf{Y}) = ||\mathbf{x}_i - \mathbf{x}_j||^2 - \phi_{i,j}$.

Another necessary ingredient is to determine function U, which has significant influence on density function $q(\mathbf{X})$. By incorporating specific prior information of data, we have various choices. One choice is to strictly preserve the pairwise distances over a given neighborhood graph, which corresponds to equality constraints (2) in MVU. To achieve this, we define $U(q(\mathbf{X})) = \sum_{i,j\in\mathcal{N}_i} \mathbb{I}(\mathbb{E}_{q(\mathbf{X})}[\psi_{i,j}(\mathbf{X};\mathbf{Y})] = 0)$, where the indicator function $\mathbb{I}(a)$ equals to 0 if condition a holds, and ∞ otherwise. As a result, the optimal $q(\mathbf{X})$ must satisfy $\mathbb{E}_{q(\mathbf{X})}[||\mathbf{x}_i - \mathbf{x}_j||^2] = \phi_{i,j}, \forall i,j \in \mathcal{N}_i$.

As discussed in [34], we can formulate different expectation constraints over pairwise distances. In this paper, we are more interested in the shrinkage effect of pairwise distances of data to form a smooth skeleton structure in the embedding space, which has been demonstrated very useful in many applications [35], [36]. This can be achieved by defining function $U(\boldsymbol{\xi}) = \sum_{i,j\in\mathcal{N}_i} \xi_{i,j}^2$ where the closeness tolerance $\xi_{i,j}$ is constrained by $\mathbb{E}_{q(\mathbf{X})}[||\mathbf{x}_i - \mathbf{x}_j||^2] - \phi_{i,j} \leq \xi_{i,j}, \forall i,j \in \mathcal{N}_i$. If $\xi_{i,j} \leq 0$, we have $\mathbb{E}_{q(\mathbf{X})}[||\mathbf{x}_i - \mathbf{x}_j||^2] \leq \phi_{i,j} + \xi_{i,j} \leq \phi_{i,j}$ so that $\mathbb{E}_{q(\mathbf{X})}[||\mathbf{x}_i - \mathbf{x}_j||^2]$ cannot be bigger than $\phi_{i,j}$. On the other hand, if $\xi_{i,j} > 0$, we probably have $\mathbb{E}_{q(\mathbf{X})}[||\mathbf{x}_i - \mathbf{x}_j||^2] \geq \phi_{i,j}$ but $\mathbb{E}_{q(\mathbf{X})}[||\mathbf{x}_i - \mathbf{x}_j||^2]$ cannot be bigger than $\phi_{i,j} + \xi_{i,j}$. Thus, the above function $U(\boldsymbol{\xi}) = \sum_{i,j\in\mathcal{N}_i} \xi_{i,j}^2$ prefers to shrink the pairwise distance of two original points as the pairwise distance of the corresponding two embedding points, but these constraints allow

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2785402, IEEE Transactions on Pattern Analysis and Machine Intelligence

the violation of changing pairwise distance no more than $\xi_{i,j}$ if $\xi_{i,j} > 0$. These constraints are quite different from the expectation constraints used in [33], [34]. Moreover, they allow us to use efficient optimization tools, which will be illustrated in Section 3.3.

3.3 Structured Projection Learning

We propose a new model by incorporating the newly defined expectation constraints into the regularized empirical Bayesian inference framework. Following PPCA, we treat $\mathbf{X}$ as a random variable and $\mathbf{W}$ as an unknown parameter. Given a neighborhood graph with a set $\mathcal{N}_i$ as the neighbors of the ith vertex of the graph. According to the regularized empirical Bayesian inference, we formulate the following optimization problem

$$\begin{aligned}\min_{\mathbf{W}} \min_{q(\mathbf{X}),\boldsymbol{\xi}} & \mathrm{KL}(q(\mathbf{X})||\pi(\mathbf{X})) - \int \log p(\mathbf{Y}|\mathbf{X},\mathbf{W},\gamma)q(\mathbf{X})d\mathbf{X} \\ & + C||\boldsymbol{\xi}||_F^2 \qquad (14) \\ \text{s.t. } & \mathbb{E}_{q(\mathbf{X})}\left[||\mathbf{x}_i - \mathbf{x}_j||^2\right] - \phi_{i,j} \leq \xi_{i,j}, \forall i,j \in \mathcal{N}_i \\ & q(\mathbf{X}) \in \mathcal{P}_{prob} \\ & \mathbf{W}^T\mathbf{W} = \mathbf{I}_d,\end{aligned}$$

where $C > 0$ is a regularization parameter, the orthogonal constraint is added for preventing arbitrarily scaling of the variable $\mathbf{W}$, $\boldsymbol{\xi}$ is an $N \times N$ matrix with (i,j)th element as $\xi_{i,j}$ and $\xi_{i,j} = 0$ if $j \notin \mathcal{N}_i$, and $\mathcal{P}_{prob}$ represents the feasible set of all density functions over $\mathbf{X}$.

The following proposition shows that problem (14) has interesting property in terms of its partial dual problem. The proof is given in the supplementary materials.

Proposition 1. *Problem (14) has an analytic solution given by*

$$q(\mathbf{X}) \propto \pi(\mathbf{X})p(\mathbf{Y}|\mathbf{X},\mathbf{W},\gamma)\exp\Big(-\sum_{i,j\in\mathcal{N}_i} s_{i,j}||\mathbf{x}_i - \mathbf{x}_j||^2\Big) \quad (15)$$

and $\xi_{i,j} = \frac{s_{i,j}}{2C}$ *where* $\mathbf{S}$ *and* $\mathbf{W}$ *can be obtained by solving the following optimization problem*

$$\begin{aligned}\max_{\mathbf{S}} \min_{\mathbf{W}} & \frac{d}{2}\log\det((\gamma+1)\mathbf{I}_N + 4\mathbf{L}) - \sum_{i,j\in\mathcal{N}_i} s_{i,j}\phi_{i,j} - \frac{1}{4C}||\mathbf{S}||_F^2 \\ & - \frac{\gamma^2}{2}Tr(\mathbf{W}^T\mathbf{Y}^T((\gamma+1)\mathbf{I}_N + 4\mathbf{L})^{-1}\mathbf{Y}\mathbf{W}) \qquad (16) \\ \text{s.t. } & s_{i,j} = 0, \forall i,j \notin \mathcal{N}_i \\ & s_{i,j} \geq 0, s_{i,j} = s_{j,i}, \forall i,j, \\ & \mathbf{W}^T\mathbf{W} = \mathbf{I}_d,\end{aligned}$$

$s_{i,j}$ *is the* (i,j)*th element of* $\mathbf{S}$*, and* $\mathbf{L} = diag(\mathbf{S1}) - \mathbf{S}$ *is the Laplacian matrix with* $\mathbf{1}$ *as the column vector of all ones.*

According to the property of Laplacian matrix, we have $\mathbf{L} = \mathrm{diag}(\mathbf{S1}) - \mathbf{S} \succeq 0$ if $\mathbf{S} \geq 0$. In other words, $\mathbf{L}$ is guaranteed to be positive semidefinite for any nonnegative $\mathbf{S}$. Define $\mathbf{Q} = \mathrm{diag}(\mathbf{S1}) - \mathbf{S} + \frac{(\gamma+1)}{4}\mathbf{I}_N$ and $\mathcal{W} = \{\mathbf{W} \in \mathbb{R}^{D\times d}|\mathbf{W}^T\mathbf{W} = \mathbf{I}_d\}$. Due to the inversion of $\mathbf{Q}$ and nonconvexity of the objective function, it is challenging to solve problem (16) globally. In order to reach a stationary point, we take the projected subgradient ascend method [37] to solve problem (16). First, we denote a function with respect to $\mathbf{S}$ as

$$g(\mathbf{S}) = \max_{\mathbf{W}\in\mathcal{W}} h(\mathbf{S},\mathbf{W}) = \frac{\gamma^2}{8}\mathrm{Tr}(\mathbf{W}^T\mathbf{Y}^T\mathbf{Q}^{-1}\mathbf{Y}\mathbf{W}). \quad (17)$$

It is worth noting that $h(\mathbf{S},\mathbf{W})$ is convex over $\mathbf{W}$ given any $\mathbf{S}$ since $\mathbf{Q}$ is positive definite. Thus, we can obtain the subgradient of $g(\mathbf{S})$ as the convex hull of union of subdifferentials of active functions at $\mathbf{S}$ given by

$$\partial g(\mathbf{S}) = \mathbf{Co}\bigcup\{\partial h(\mathbf{S},\mathbf{W})|h(\mathbf{S},\mathbf{W}) = h(\mathbf{S},\mathbf{W}^*), \forall\mathbf{W}\in\mathcal{W}\}, \quad (18)$$

where $\mathbf{W}^* = \arg\max_{\mathbf{W}\in\mathcal{W}} h(\mathbf{S},\mathbf{W})$. In this paper, we take $\partial h(\mathbf{S},\mathbf{W}^*)$ as the ascend direction. According to (17), given an $\mathbf{S}$ or $\mathbf{Q}$, maximizing $h(\mathbf{S},\mathbf{W})$ with respect to $\mathbf{W}$ is equivalent to the problem of classical PCA where the covariance matrix is $\mathbf{Y}^T\mathbf{Q}^{-1}\mathbf{Y}$. Hence, $\mathbf{W}^*$ consists of the eigenvectors corresponding to the d largest eigenvalues of the covariance matrix. Let

$$f(\mathbf{S}) = \frac{d}{2}\log\det(\mathbf{Q}) - \sum_{i,j\in\mathcal{N}_i} s_{i,j}\phi_{i,j} - \frac{1}{4C}||\mathbf{S}||_F^2 - \max_{\mathbf{W}\in\mathcal{W}} h(\mathbf{S},\mathbf{W})$$

Thus, we can compute subgradient $\partial f(\mathbf{S})$ based on the optimal $\mathbf{W}^*$. Considering the symmetric property of $\mathbf{S}$, the derivative of the $\log\det$ term with respect to $s_{i,j}$ and a pair of indexes $(i,j) \in \mathcal{I}_\mathcal{N} = \{j < i \wedge j \in \mathcal{N}_i, \forall i\}$ is obtained by

$$\begin{aligned}\frac{\partial\log\det(\mathbf{Q})}{\partial s_{i,j}} &= \mathrm{Tr}\left(\left[\frac{\partial\log\det(\mathbf{Q})}{\partial\mathbf{Q}}\right]^T \frac{\partial[\mathrm{diag}(\mathbf{S1}) - \mathbf{S} + \frac{(\gamma+1)}{4}\mathbf{I}_N]}{\partial s_{i,j}}\right) \\ &= \mathrm{Tr}(\mathbf{Q}^{-1}\mathbf{A}_{i,j})\end{aligned}$$

where the matrix $\mathbf{A}_{i,j}$ can be represented by

$$[\mathbf{A}_{i,j}](m,n) = \begin{cases} 1, & m = n = i \text{ or } m = n = j \\ -1, & m = i \wedge n = j \text{ or } m = j \wedge n = i \\ 0, & \text{otherwise.}\end{cases}$$

Let $\mathbf{s}$ be the vectorization of matrix $\mathbf{S}$ with indexes $(i,j) \in \mathcal{I}_\mathcal{N}$. As a result, the subgradient of the objective function $f(\mathbf{S})$ can be computed as, $\forall(i,j) \in \mathcal{I}_\mathcal{N}$,

$$\partial_{s_{i,j}} f(\mathbf{s}) = \frac{1}{2}\mathrm{Tr}(\mathbf{Q}^{-T}(d\mathbf{I}_N + \frac{\gamma^2}{4}\mathbf{P})\mathbf{A}_{i,j}) - \frac{1}{C}s_{i,j} - 2||\mathbf{y}_i - \mathbf{y}_j||^2 \quad (19)$$

where $\mathbf{P} = \mathbf{Y}\mathbf{W}^*\mathbf{W}^{*T}\mathbf{Y}^T\mathbf{Q}^{-1}$. Finally, we can solve (16) by using the projected subgradient ascend method as

$$\mathbf{s}^{(t+1)} = \Pi_{\mathbf{s}\geq 0}\left(\mathbf{s}^{(t)} + \alpha_t\partial_{\mathbf{s}} f(\mathbf{s}^{(t)})\right),$$

where $\Pi_{\mathbf{s}\geq 0}$ is the projection on the non-negative set, and α_t is the step size in the tth iteration. In order to guarantee the convergence, the step size should have the properties: $\sum_{t=1}^{\infty}\alpha_t^2 < \infty$ and $\sum_{t=1}^{\infty}\alpha_t = \infty$. In the experiments, we take a typical example as $\alpha_t = 1/t$ [37].

After learning the posterior distribution (15) with optimal $\mathbf{S}$ and $\mathbf{W}$, we obtain the embedded points $\mathbf{X}$ by maximizing the logarithm of (15) as the maximum a posteriori estimation, i.e., $\max_{\mathbf{X}}\log q(\mathbf{X})$, which can be rewritten as

$$\min_{\mathbf{X}} \frac{1}{2}\mathrm{Tr}(\mathbf{X}^T(4\mathbf{L} + (\gamma+1)\mathbf{I}_N)\mathbf{X}) - \gamma\mathrm{Tr}(\mathbf{X}^T\mathbf{Y}\mathbf{W}). \quad (20)$$

Problem (20) is a quadratic optimization problem since matrix $4\mathbf{L} + (\gamma+1)\mathbf{I}_N$ is positive definite. Thus, we can obtain an analytic solution by setting its derivative with respect to $\mathbf{X}$ to zero, given by

$$\mathbf{X} = \gamma(4\mathbf{L} + (\gamma+1)\mathbf{I}_N)^{-1}\mathbf{Y}\mathbf{W} = \frac{\gamma}{4}\mathbf{Q}^{-1}\mathbf{Y}\mathbf{W}. \quad (21)$$

As observed, there is a trivial solution for above objective, i.e., $\mathbf{X} = 0$ if $\gamma = 0$. To overcome this issue, we have the following observation: the posterior distribution is the matrix normal distribution [32] given by

$$q(\mathbf{X}) \sim \mathcal{MN}_{N,d}(\mathbf{0}, \Sigma, \mathbf{I}_d) \quad (22)$$

Algorithm 1 Structured Projection Learning (SPL)

```
Input: Data $\mathbf{Y}$, neighbors $\mathcal{N}_i, \forall i$, reduced dimension $d$ parameters $\gamma$ and $C$
$t = 1$, $\mathbf{S}^{(t)} = \mathbf{0}$
repeat
    $\mathbf{Q} = \text{diag}(\mathbf{S}^{(t)}\mathbf{1}) - \mathbf{S}^{(t)} + \frac{(\gamma+1)}{4}\mathbf{I}_N$
    Perform eigendecomposition $\mathbf{Y}^T\mathbf{Q}^{-1}\mathbf{Y} = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^T$ with diag($\mathbf{\Gamma}$) sorted in a descending order
    $\mathbf{W} = \mathbf{U}(:, 1\!:\!d)$
    Compute subgradient $\partial_{\mathbf{s}} f(\mathbf{s}^{(t)})$ using (19)
    $\mathbf{s}^{(t+1)} = \Pi_{\mathbf{s} \geq 0}\left(\mathbf{s}^{(t)} + \frac{1}{t}\partial_{\mathbf{s}} f(\mathbf{s}^{(t)})\right)$
    $t = t + 1$
until Convergence
Output: Embedding $\mathbf{X} = \frac{\gamma}{4}\mathbf{Q}^{-1}\mathbf{Y}\mathbf{W}$
```

where $\Sigma = (4\mathbf{L} + \mathbf{I}_N)^{-1}$ is the sample-based covariance matrix and can also be interpreted as a regularized Laplacian kernel [38]. As a result, we can apply KPCA on Σ to achieve the embedded data points if $\gamma = 0$ to avoid trivial solution. The pseudo-code of our proposed structured projection learning (SPL) is described in Algorithm 1.

3.4 Algorithm Analysis

The computational complexity of Algorithm 1 can be estimated as follows: solving problem (16) takes approximately $O(N^{2.37})$ for computing logdet and inversion of matrix $\mathbf{Q}$ at each iteration; computing subgradient and function value of $f(\mathbf{S})$ takes $O(N^3)$ due to the eigendecomposition. The time complexity of Algorithm 1 takes the order of $O(N^3)$. Our method can also leverage the fast eigendecomposition methods for finding a small number of large eigenvalues and eigenvectors. Thus, the computational complexity of Algorithm 1 is the same as that in most of spectral based methods, but is much faster than semidefinite programming used in MVU. The theoretical convergence analysis of Algorithm 1 follows the projected subgradient method [37].

Algorithm 1 takes two parameters into account, C and γ, except the low-dimensional parameter d, which is a common parameter for dimensionality reduction and will not be discussed in this paper. Parameter C regulates the error tolerance of pairwise distances between original points and embedding points. The larger the C is, the smaller the error tolerance is imposed. In the case of $C = \infty$, the model does not allow the error. Parameter γ controls the noise of data in the generative model (5). More interestingly, this parameter plays an important role on balancing two distinct models: deterministic model (1) and generative model (5). The role becomes clear by investigating the proposed unified model (16). Specifically, problem (16) only learns the similarity matrix $\mathbf{S}$ using pairwise distances of original data as input if $\gamma = 0$. On the other hand, if $\gamma > 0$, the random noise of original points is simultaneously incorporated by regulating similarity matrix learning of deterministic model and data reconstruction of generative model. The merit of non-zero γ leads to an easy embedding process and meanwhile maintaining the intrinsic structure of data in low-dimensional space, which will be investigated in Section 6.1.

The proposed embedding framework provides a novel way to automatically learn a sparse positive similarity matrix $\mathbf{S}$ from a set of pairwise distances, and the sparse positive similarity matrix is purposely designed for learning the embedded points. This also provides a probabilistic interpretation why MVU takes KPCA as the embedding method after learning a kernel matrix.

4 Learning Explicit Graph Structure

For certain applications, we know the explicit representation of latent graph structure that generates the observed data, but both the embedded points and the correspondence between vertexes of the graph and the observed data are unknown. In this section, we adapt structured projection learning to learn an explicit graph structure, so that the learned embedded points reside on the optimal graph inside a set of feasible graphs with the given graph representation in the latent space.

4.1 Explicit Graph Structure Learning

Before presenting the model for explicitly learning a graph structure, we first introduce an important result.

Proposition 2. *Given an* $\mathbf{S}$, $\min_{\mathbf{W}} -\frac{4}{\gamma^2}h(\mathbf{S}, \mathbf{W})$ *is equivalent to the following optimization problem*

$$\begin{aligned}\min_{\mathbf{W}\in\mathcal{W},\mathbf{Z}} & \frac{1}{2(\gamma+1)}||\mathbf{Y} - \mathbf{Z}\mathbf{W}^T||_F^2 + \frac{2}{(\gamma+1)^2}\mathit{Tr}(\mathbf{Z}^T\mathbf{L}\mathbf{Z}) \\ & - \frac{1}{2(\gamma+1)}||\mathbf{Y}||_F^2,\end{aligned} \tag{23}$$

where $\mathbf{Z} = \frac{\gamma+1}{4}\mathbf{Q}^{-1}\mathbf{Y}\mathbf{W}$.

According to the property of Laplacian matrix and the above proposition, we reformulate the problem (16) as

$$\begin{aligned}\max_{\mathbf{S}} \min_{\mathbf{W}\in\mathcal{W}} & \frac{d}{2}\log\det((\gamma+1)\mathbf{I}_N + 4\mathbf{L}) - \langle\mathbf{S}, \Phi_{\mathbf{Y}}\rangle - \frac{1}{4C}||\mathbf{S}||_F^2 \\ & + \frac{\gamma^2}{8(\gamma+1)}||\mathbf{Y} - \mathbf{Z}\mathbf{W}^T||_F^2 + \frac{\gamma^2}{4(\gamma+1)^2}\text{Tr}(\mathbf{Z}^T\mathbf{L}\mathbf{Z}),\end{aligned} \tag{24}$$

where $\mathbf{Z}$ is a matrix according to Proposition 2 and $\Phi_{\mathbf{Y}}$ is a distance matrix with the (i, j)th element as $||\mathbf{y}_i - \mathbf{y}_j||^2$. $||\mathbf{Y}||_F^2$ is removed because it is a constant with respect to $\mathbf{S}$ and $\mathbf{W}$. From (24), given an $\mathbf{S}$, we have

$$\begin{aligned}\min_{\mathbf{W},\mathbf{Z}} & \frac{\gamma^2}{8(\gamma+1)}||\mathbf{Y} - \mathbf{Z}\mathbf{W}^T||_F^2 + \frac{\gamma^2}{4(\gamma+1)^2}\text{Tr}(\mathbf{Z}^T\mathbf{L}\mathbf{Z}) \\ \text{s.t. } & \mathbf{W}^T\mathbf{W} = \mathbf{I}_d,\end{aligned} \tag{25}$$

by removing the following three terms that regulate $\mathbf{S}$, i.e., $\frac{d}{2}\log\det((\gamma+1)\mathbf{I}_N + 4\mathbf{L})$, $-\langle\mathbf{S}, \Phi_{\mathbf{Y}}\rangle$, and $-\frac{1}{4C}||\mathbf{S}||_F^2$. Thus, we can view model (16) as an approach to learn $\mathbf{S}$ from data by simultaneously preserving expected distances and optimizing (25) as the learning criterion for dimensionality reduction based on a graph. Similarly, we can also learn an explicit graph structure by incorporating known constraints of certain graph structures and minimizing criterion (25).

Following the above annotations, we can define a general graph representation. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V} = \{V_1, \ldots, V_N\}$ is a set of vertices and $\mathcal{E}$ is a set of edges. Suppose that every vertex V_i corresponds to a point $\mathbf{z}_i \in \mathcal{Z} \subset \mathbb{R}^d$, which lies in an intrinsic space of dimension d. Denote the weight of edge (V_i, V_j) as $s_{i,j}$, which represents the similarity (or connection indicator) between $\mathbf{z}_i$ and $\mathbf{z}_j$ in the intrinsic space $\mathcal{Z}$. We assume that matrix $\mathbf{S} \in \mathcal{S}$ with the (i, j)th element as $s_{i,j}$ can be used to define the representation of a latent graph, where $\mathcal{S}$ is a set of feasible graphs with the given graph representation.

By combining the above ingredients, we formulate the following optimization problem, given by

$$\min_{\mathbf{S},\mathbf{W},\mathbf{Z}} \frac{\gamma^2}{8(\gamma+1)}\left\{||\mathbf{Y}-\mathbf{Z}\mathbf{W}^T||_F^2+\lambda\mathrm{Tr}(\mathbf{Z}^T\mathbf{L}\mathbf{Z})\right\} \quad (26)$$
$$\text{s.t. } \mathbf{W}^T\mathbf{W}=\mathbf{I}_d, \mathbf{S}\in\mathcal{S},$$

where $\lambda=\frac{2}{\gamma+1}$. Problem (26) can be used to learn a graph structure represented by $\mathbf{S}$ in (26) in the feasible set $\mathcal{S}$, but problem (25) is based on a given $\mathbf{S}$. Thus, formulation (26) is a general framework for dimensionality reduction by learning an intrinsic graph structure in a low-dimensional space. In order to instantiate a new method, we have to specify the feasible set $\mathcal{S}$ of graphs.

4.2 Dimensionality Reduction via Learning a Tree

We investigate a family of tree structures, which can be used to deal with various real world problems.

Given a connected undirected graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ with a cost $c_{i,j}$ associated with edge $(V_i,V_j)\in\mathcal{E},\forall i,\forall j$, let $\mathcal{T}=(\mathcal{V},\mathcal{E}_\mathcal{T})$ be a tree with the minimum total cost and $\mathcal{E}_\mathcal{T}$ be the edges forming the tree. In order to represent and learn a tree, we consider $\{s_{i,j}\}$ as binary variables where $s_{i,j}=1$ if $(V_i,V_j)\in\mathcal{E}_\mathcal{T}$, and 0 otherwise. Denote $\mathbf{S}=[s_{i,j}]\in\{0,1\}^{N\times N}$. The integer linear programming formulation of minimum spanning tree (MST) can be written as: $\min_{\mathbf{S}\in\mathcal{S}_0}\sum_{i,j}s_{i,j}c_{i,j}$, where $\mathcal{S}_0=\{\mathbf{S}\in\{0,1\}^{N\times N}\}\cap\mathcal{S}'$ and $\mathcal{S}'=\{\mathbf{S}=\mathbf{S}^T\}\cap\{\frac{1}{2}\sum_{i,j}s_{i,j}=|\mathcal{V}|-1\}\cap\{\frac{1}{2}\sum_{V_i\in\mathcal{A},V_j\in\mathcal{A}}s_{i,j}\le|\mathcal{A}|-1,\forall\mathcal{A}\subseteq\mathcal{V}\}$. The first constraint of $\mathcal{S}'$ enforces the symmetric connection of an undirected graph, e.g. $s_{i,j}=s_{j,i}$. The second constraint states that the spanning tree only contains $|\mathcal{V}|-1$ edges. The third constraint imposes the acyclicity and connectivity properties of a tree. Instead of solving the above integer programming problem, we resort to a relaxed problem by letting $s_{i,j}\ge 0$, that is,

$$\min_{\mathbf{S}\in\mathcal{S}_\mathcal{T}}\sum_{i,j}s_{i,j}c_{i,j}, \quad (27)$$

where the set of linear constraints over convex domain is given by $\mathcal{S}_\mathcal{T}=\{\mathbf{S}\ge 0\}\cap\mathcal{S}'$. Problem (27) can be solved by Kruskal's algorithm [39].

Let $\lambda=\frac{2}{\gamma+1}$. We can equivalently rewrite (26) as the following optimization problem

$$\min_{\mathbf{W},\mathbf{Z},\mathbf{S}}\sum_{i=1}^N||\mathbf{y}_i-\mathbf{W}\mathbf{z}_i||^2+\frac{\lambda}{2}\sum_{i,j}s_{i,j}||\mathbf{z}_i-\mathbf{z}_j||^2 \quad (28)$$
$$\text{s.t. } \mathbf{W}^T\mathbf{W}=\mathbf{I}_d,\ \ \mathbf{S}\in\mathcal{S},$$

where $\mathbf{W}=[\mathbf{w}_1,\ldots,\mathbf{w}_d]\in\mathbb{R}^{D\times d}$ is a matrix consisting of an orthogonal set of d linear basis vectors $\mathbf{w}_l\in\mathbb{R}^D,\forall l$, $\mathbf{Z}=[\mathbf{z}_1,\ldots,\mathbf{z}_N]^T\in\mathbb{R}^{N\times d}$ represents the projected data points in the low-dimensional space $\mathbb{R}^d$, $\mathbf{S}=[s_{i,j}]\in\mathbb{R}^{N\times N}$ is an adjacent matrix of a tree $\mathcal{T}=(\mathcal{V},\mathcal{E}_\mathcal{T})$ where $\mathcal{E}_\mathcal{T}=\{(i,j):s_{i,j}\neq 0\}$.

If $\lambda=0$, problem (28) is equivalent to the optimization problem of PCA, otherwise the data points $\mathbb{Y}$ are mapped into a low-dimensional space where $\{\mathbf{z}_i\}_{i=1}^N$ form a tree. Therefore, PCA is a special case of problem (28). Another important observation is the distances between any two latent points are computed in a low-dimensional space, i.e., $||\mathbf{z}_i-\mathbf{z}_j||^2$ since latent points $\{\mathbf{z}_i\}_{i=1}^N$ are in $\mathbb{R}^d$. As a result, problem (28) can effectively mitigate the curse of dimensionality. For ease of reference, we name the problem (28) as dimensionality reduction tree (DRTree).

Algorithm 2 Discriminative DRTree (DDRTree)

```
**Input:** Data matrix $\mathbf{Y}$, parameters $\lambda$, $\sigma$ and $\gamma$
Initialize $\mathbf{Z}$ by PCA
$K=N$, $\mathbf{C}=\mathbf{Z}$
**repeat**
    $c_{k,k'}=||\mathbf{c}_k-\mathbf{c}_{k'}||^2,\forall k,\forall k'$
    Obtain $\mathbf{S}$ by solving (27) via Kruskal's algorithm
    $\mathbf{L}=\mathrm{diag}(\mathbf{S}\mathbf{1})-\mathbf{S}$
    Compute $\mathbf{R}$ with each element as (31)
    $\Gamma=\mathrm{diag}(\mathbf{1}^T\mathbf{R})$
    $\mathbf{Q}=\frac{1}{1+\gamma}\left[\mathbf{I}_N+\mathbf{R}\left(\frac{1+\gamma}{\gamma}\left(\frac{\lambda}{\gamma}\mathbf{L}+\Gamma\right)-\mathbf{R}^T\mathbf{R}\right)^{-1}\mathbf{R}^T\right]$
    Perform eigendecomposition
    $\mathbf{Y}^T\mathbf{Q}\mathbf{Y}=\mathbf{U}\Lambda\mathbf{U}^T$
    and diag($\Lambda$) is sorted in a descending order.
    $\mathbf{W}=\mathbf{U}(:,1:d)$
    $\mathbf{Z}=\mathbf{Q}\mathbf{Y}\mathbf{W}$
    $\mathbf{C}=\left(\frac{\lambda}{\gamma}\mathbf{L}+\Gamma\right)^{-1}\mathbf{R}^T\mathbf{Z}$
**until** Convergence
```

4.3 Discriminative DRTree

DRTree projects data points in a high-dimensional space to latent points that directly form a tree structure in the low-dimensional space. However, the tree structure achieved might be at the risk of losing clustering information. In other words, some data points are supposed to form a cluster, but they are scattered to different branches of the tree, and distances between them on the intrinsic structure become large.

To incorporate the discriminative information, we introduce another set of latent points $\{\mathbf{c}_k\}_{k=1}^K$ as the centers of $\{\mathbf{z}_i\}_{i=1}^N$ where $\mathbf{c}_k\in\mathbb{R}^d$ so as to minimize the trade-off between the objective functions of K-means and DRtree. As a result, we formulate the following optimization problem

$$\begin{aligned}\min_{\mathbf{W},\mathbf{Z},\mathbf{S},\mathbf{C},\mathbb{P}}&\sum_{i=1}^N||\mathbf{y}_i-\mathbf{W}\mathbf{z}_i||^2+\frac{\lambda}{2}\sum_{k,k'}s_{k,k'}||\mathbf{c}_k-\mathbf{c}_{k'}||^2\\&+\gamma\sum_{k=1}^K\sum_{j\in\mathcal{P}_k}||\mathbf{z}_j-\mathbf{c}_k||^2 \quad (29)\\\text{s.t. }&\mathbf{W}^T\mathbf{W}=\mathbf{I}_d,\ \ \mathbf{S}\in\mathcal{S}_\mathcal{T},\end{aligned}$$

where the third term of the objective function is same as the objective function of K-means, $\mathbb{P}=\{\mathcal{P}_1,\ldots,\mathcal{P}_K\}$ is a partition of $\{1,\ldots,N\}$, $\mathbf{C}=[\mathbf{c}_1,\ldots,\mathbf{c}_K]^T\in\mathbb{R}^{K\times d}$ and $\gamma\ge 0$ is a trade-off parameter between the objective function of DRTree and empirical quantization error of latent points $\{\mathbf{z}_i\}_{i=1}^N$ and $\{\mathbf{c}_k\}_{k=1}^K$.

Unlike problem (28), problem (29) is now regularized on centers $\{\mathbf{c}_k\}_{k=1}^K$ instead of $\{\mathbf{z}_i\}_{i=1}^N$. However, problem (28) is a special case of problem (29) if $K=N$ and $\gamma\to\infty$ since the third term can be removed without changing the optimal solution of (29) due to $\mathbf{z}_i=\mathbf{y}_i,\forall i$ at optimum. In this case, problems (28) and (29) are equivalent. Except for the special case, problem (29) is able to achieve discriminative and compact feature representation for dimensionality reduction since clustering objective and DRTree are optimized in a unified framework.

The hard partition imposed by K-means, however, has several drawbacks. First, parameter K is data-dependent, so it is hard to set properly. Second, it is sensitive to noise, outliers, or some data points that cannot be thought of as belonging to a single cluster [40]. Soft partition methods such as Gaussian mixture modeling have also been used in modeling principal curves [41], [42]. However, the likelihood of a Gaussian mixture model tends to be infinite when a singleton is formed [42]. To alleviate the problems from which the aforementioned methods suffer, we propose to replace the hard partition K-means with a relaxed regularized empirical quantization error given by

$$\begin{aligned}\min_{\mathbf{W},\mathbf{Z},\mathbf{S},\mathbf{C},\mathbf{R}} & \sum_{i=1}^{N} ||\mathbf{y}_i - \mathbf{W}\mathbf{z}_i||^2 + \frac{\lambda}{2}\sum_{k,k'} s_{k,k'}||\mathbf{c}_k - \mathbf{c}_{k'}||^2 \\ & + \gamma\left[\sum_{k=1}^{K}\sum_{i=1}^{N} r_{i,k}||\mathbf{z}_i - \mathbf{c}_k||^2 + \sigma\Omega(\mathbf{R})\right] \quad (30) \\ \text{s.t. } & \mathbf{W}^T\mathbf{W} = \mathbf{I}_d, \ \ \mathbf{S} \in \mathcal{S}_{\mathcal{T}}, \ \ \sum_{k=1}^{K} r_{i,k} = 1, r_{i,k} \geq 0, \forall i, \forall k,\end{aligned}$$

where $\mathbf{R} \in \mathbb{R}^{N\times K}$ with the (i,k)th entry as $r_{i,k}$, $\Omega(\mathbf{R}) = \sum_{i=1}^{N}\sum_{k=1}^{K} r_{i,k}\log r_{i,k}$ is the negative entropy regularization, and $\sigma > 0$ is the regularization parameter. The negative entropy regularization transforms hard assignment used in K-means to soft assignment used in Gaussian mixture models [28], and is also used in other tasks [43].

The following proposition shows that problem (30) with respect to $\{\mathbf{C},\mathbf{R}\}$ by fixing the remaining variables is equivalent to the mean shift clustering method [44], which is able to determine the number of clusters automatically and initialize centers $\{\mathbf{c}_k\}_{k=1}^{K}$ by latent points $\{\mathbf{z}_i\}_{i=1}^{N}, \forall i$ if $K = N$. The following lemma shows that the optimal solution $\mathbf{R}$ has an analytical expression if $\mathbf{C}$ is given.

Lemma 1. *Given $\{\mathbf{W},\mathbf{Z},\mathbf{S},\mathbf{C}\}$, Problem (30) has the optimal solution $\mathbf{R}$ given by the following analytical form, $\forall k, \forall i$*

$$r_{i,k} = \exp\left(-||\mathbf{z}_i - \mathbf{c}_k||^2/\sigma\right)\Big/\sum_{k=1}^{K}\exp\left(-||\mathbf{z}_i - \mathbf{c}_k||^2/\sigma\right). \quad (31)$$

Proposition 3. *Given $\{\mathbf{W},\mathbf{Z},\mathbf{S}\}$, $\lambda = 0$ and assuming $K = N$, problem (30) with respect to $\{\mathbf{C},\mathbf{R}\}$ can be solved by a mean shift clustering method by initializing $\mathbf{C} = \mathbf{Z}$.*

The key difference between problem (30) and the traditional mean shift is that the latent points $\{\mathbf{z}_i\}_{i=1}^{N}$ in our model are variables and can be affected by dimensionality reduction and tree structure learning. We also build a connection between problem (30) and problem (29) as shown in the following proposition.

Proposition 4. *If $\sigma \to 0$, (30) is equivalent to (29).*

The above properties of problem (30) facilitates the setting of parameters in different contexts of applications. In the case of dimensionality reduction, discriminative information might be important for some applications such as clustering problems, and Proposition 3 provides a natural way to form a cluster without predefining the number of clusters. In the case of finding K clusters, we prefer problem (29) to (30) since (29) is formulated in terms of K clusters directly. According to Proposition 4, the purpose of clustering can also be achieved by solving problem (30) with a small σ.

Alternating structure optimization [45] is used to solve problem (30). We first partition variables into two groups $\{\mathbf{W},\mathbf{Z},\mathbf{C}\}$ and $\{\mathbf{S},\mathbf{R}\}$, and then solve each subproblem iteratively until the convergence is achieved.

Given $\{\mathbf{S},\mathbf{R}\}$, we can obtain an analytical solution by solving problem (30) with respect to $\{\mathbf{W},\mathbf{Z},\mathbf{C}\}$, which is discussed in Proposition 5. Before presenting Proposition 5, we first state a necessary condition of the proposition in Lemma 2 by proving the existence of the inverse matrix of $\frac{1+\gamma}{\gamma}(\frac{\lambda}{\gamma}\mathbf{L} + \Gamma) - \mathbf{R}^T\mathbf{R}$.

Lemma 2. *The inverse of matrix $\frac{1+\gamma}{\gamma}(\frac{\lambda}{\gamma}\mathbf{L} + \Gamma) - \mathbf{R}^T\mathbf{R}$ exists if $\sum_{i=1}^{N} r_{i,k} > 0, \forall k$, where $\Gamma = \mathrm{diag}(\mathbf{1}^T\mathbf{R})$ and Laplacian matrix over a tree encoded in $\mathbf{S}$ is $\mathbf{L} = \mathrm{diag}(\mathbf{S}\mathbf{1}) - \mathbf{S}$.*

The conditions $\sum_{i=1}^{N} r_{i,k} > 0, \forall k$, always hold in the case of the soft-assignment obtained by Lemma 1.

Proposition 5. *By fixing $\{\mathbf{S},\mathbf{R}\}$, problem (30) with respect to $\{\mathbf{W},\mathbf{Z},\mathbf{C}\}$ has the following analytical solution:*

$$\mathbf{W} = \mathbf{U}(:,1:d), \ \ \mathbf{Z} = \mathbf{Q}\mathbf{Y}\mathbf{W}, \ \ \mathbf{C} = \left(\frac{\lambda}{\gamma}\mathbf{L} + \Gamma\right)^{-1}\mathbf{R}^T\mathbf{Z} \quad (32)$$

where $\mathbf{Q} = \frac{1}{1+\gamma}\left[\mathbf{I}_N + \mathbf{R}\left(\frac{1+\gamma}{\gamma}\left(\frac{\lambda}{\gamma}\mathbf{L} + \Gamma\right) - \mathbf{R}^T\mathbf{R}\right)^{-1}\mathbf{R}^T\right]$, $\mathbf{U}$ and $\mathrm{diag}(\Lambda)$ are the eigenvectors and eigenvalues of matrix $\mathbf{Y}^T\mathbf{Q}\mathbf{Y}$ with $\mathrm{diag}(\Lambda)$ sorted in a descending order, respectively, $\Gamma = \mathrm{diag}(\mathbf{1}^T\mathbf{R})$ and the Laplacian matrix over a tree encoded in $\mathbf{S}$ is defined as $\mathbf{L} = \mathrm{diag}(\mathbf{S}\mathbf{1}) - \mathbf{S}$.

By fixing $\{\mathbf{W},\mathbf{Z},\mathbf{Y}\}$, problem (30) with respect to $\{\mathbf{S},\mathbf{R}\}$ is a jointly convex optimization problem with respect to $\mathbf{S}$ and $\mathbf{R}$. Importantly, the subproblems with respective to $\mathbf{S}$ and $\mathbf{R}$ can be solved independently. According to Lemma 1, the optimal $\mathbf{R}$ is given by equation (31). To obtain the optimal $\mathbf{S}$, the optimization problem (30) with respect to $\mathbf{S}$ is $\min_{\mathbf{S}\in\mathcal{S}_{\mathcal{T}}}\sum_{k,k'} s_{k,k'}||\mathbf{c}_k - \mathbf{c}_{k'}||^2$, which can be solved by Kruskal's method.

As discussed in Section 4.2, PCA is a special case of DRTree, so variable $\mathbf{Z}$ can be naturally initialized by PCA. By Proposition 3, we can set $K = N$ and initialize $\mathbf{Y} = \mathbf{Z}$. The pseudo-code of Discriminative DRTree is given in Algorithm 2, briefly named as *DDRTree*. The implementations of *DRTree* and *DDRTree* in both MATLAB and R can be freely available[1].

5 Connections to Existing Methods

We have developed several methods based on regularized empirical Bayesian inference so that both the global assumption of generative model and the local assumption of manifold learning are naturally incorporated into a unified model. In addition to the relationships of our method to MVU and various probabilistic models such as PPCA and GPLVM discussed in the related work, we further present a detailed discussion of other existing methods that are closely related to our proposed model.

[1] http://www.uta.edu/faculty/wangl3/

5.1 Connection to Reversed Graph Embedding

Reversed graph embedding [28] was proposed to learn a set of principal points in the original space. Given a dataset $\mathcal{D} = \{\mathbf{y}_i\}_{i=1}^N$, it formulates the following optimization problem to learn a set of latent variables $\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ in low-dimensional space with $\mathbf{z}_i \in \mathcal{Z}$ given by

$$\min_{\mathcal{G}\in\widehat{\mathcal{G}}_b} \min_{f_\mathcal{G}\in\mathcal{F}} \min_{\{\mathbf{z}_1,\ldots,\mathbf{z}_N\}} \sum_{i=1}^N ||\mathbf{y}_i - f_\mathcal{G}(\mathbf{z}_i)||^2 + \frac{\lambda}{2}\sum_{i,j} b_{i,j}||f_\mathcal{G}(\mathbf{z}_i) - f_\mathcal{G}(\mathbf{z}_j)||^2, \tag{33}$$

where $\lambda \geq 0$ is a parameter that controls the trade-off between the data reconstruction error and the objective function of reverse graph embedding, and $\widehat{\mathcal{G}}_b$ is a feasible set of graphs with the set $\mathcal{V}$ of vertices and a set $\mathcal{E}$ of edges specified by a set $\{b_{i,j}\}$ of edge weights. We consider learning a function $f_\mathcal{G} \in \mathcal{F}$ and $f_\mathcal{G} : \mathcal{Z} \to \mathcal{Y}$ over $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that maps the intrinsic space $\mathcal{Z}$ to the input space $\mathcal{Y}$.

For simplicity, the work [28] considers learning $f_\mathcal{G}(\mathbf{z}_i)$ as one single variable for variables $f_\mathcal{G}$ and $\mathbf{z}_i$. In contrast, we in this paper aim to learn a set of points and the projection matrix as two separate variables, so that we can control the reduced dimensionality of the intrinsic space where the graph structure may reside. Moreover, we provide a general similarity matrix learning framework (16) for principal graph learning and special tree structure learning formulations (26) and (29).

5.2 Connection to Maximum Entropy Unfolding

MEU [19] was proposed to directly model the density of observed data $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ by minimizing the KL divergence between a base density $m(\mathbf{Y})$ and the density $p(\mathbf{Y})$ given by $\min_{p(\mathbf{Y})} \int p(\mathbf{Y}) \log(p(\mathbf{Y})/m(\mathbf{Y}))\mathrm{d}\mathbf{Y}$, under the constraints on the expected squared pairwise distances $\phi_{i,j}$ of any two samples, $\mathbf{y}_i$ and $\mathbf{y}_j$. Let $m(\mathbf{Y})$ be a very broad, spherical Gaussian density with covariance $\lambda^{-1}\mathbf{I}_D$. The density function is then constructed as

$$p(\mathbf{Y}) \propto \exp\Big(-\frac{1}{2}\mathrm{Tr}(\lambda\mathbf{Y}\mathbf{Y}^T)\Big)\exp\Big(-\frac{1}{2}\sum_i\sum_{j\in\mathcal{N}_i} s_{i,j}\phi_{i,j}\Big),$$

even though the explicit form of these constraints is not given. The Laplacian matrix $\mathbf{L}$ defined over similarity $s_{i,j}$ is achieved by maximizing the logarithmic function of $p(\mathbf{Y})$. Finally, the embedding is obtained by applying KPCA on the kernel matrix $\mathbf{K} = (\mathbf{L} + \lambda\mathbf{I}_N)^{-1}$.

One of the key differences is that our framework directly models the posterior distribution $p(\mathbf{X}|\mathbf{Y})$ of latent data, while MEU models the density of observed data. As a result, MEU has to assume that the data features are i.i.d. given the model parameters. However, this assumption is hardly satisfied if feature correlation exists. In contrast, our model assumes that the reduced features in the latent space are i.i.d, which is more reasonable than that used in MEU since the latent space is generally assumed to be formed by a set of orthogonal bases, such as PCA and KPCA.

5.3 Connection to Structure Learning

SMCE [24] was proposed using ℓ_2 norm over the errors that measure the linear representation of every data point by using its neighborhood information. Similarly, ℓ_1 graph was learned for image analysis using ℓ_1 norm over the errors for enhancing the robustness of the learned graph [26]. These two methods [24], [26] learn a directed graph from data so that they might yield suboptimal results by heuristically transforming a directed graph to an undirected graph for clustering and dimensionality reduction.

Instead of learning directed graphs by using the above two methods, an integrated model for learning an undirected graph by imposing a sparsity penalty (i.e., ℓ_1 prior) on a symmetric similarity matrix and a positive semidefinite constraint on the Laplacian matrix was proposed [27] as,

$$\begin{aligned}\max_{\mathbf{Q}\succeq 0, \mathbf{S}, \sigma^2}\ & \log\det(\mathbf{Q}) - \frac{1}{D}\mathrm{Tr}(\mathbf{Q}\mathbf{Y}\mathbf{Y}^T) - \frac{\beta}{D}||\mathbf{S}||_1 \\ \text{s.t. } & \mathbf{Q} = \mathrm{diag}(\mathbf{S}\mathbf{1}) - \mathbf{S} + \mathbf{I}_N/\sigma^2 \\ & s_{i,i} = 0, s_{i,j} \geq 0, \forall\, i,j \\ & \sigma^2 > 0,\end{aligned} \tag{34}$$

where $\beta > 0$ is a regularization parameter. Another approach dimensionality reduction through regularization of the inverse covariance in the loglikelihood (DRILL) [19] was proposed by applying an ℓ_1 prior to the elements $\Lambda_{i,j}, \forall i,j$ of an inverse covariance Λ, i.e. $||\Lambda||_1 = \sum_{i,j}|\Lambda_{i,j}|$, given by,

$$\max_{\Lambda\succeq 0} \frac{D}{2}\log\det(\Lambda + \lambda\mathbf{I}_N) - \frac{1}{2}\mathrm{Tr}((\Lambda + \lambda\mathbf{I}_N)\mathbf{Y}\mathbf{Y}^T) - ||\Lambda||_1, \tag{35}$$

and an implied covariance matrix is $\mathbf{K} = (\Lambda + \lambda\mathbf{I}_N)^{-1}$.

By comparing (16) with (34) and (35), we can see that (16) is more similar to (34) since the sparsity is imposed on $\mathbf{S}$, not on Λ, which is analogous to the Laplacian matrix $\mathbf{L} = \mathrm{diag}(\mathbf{S}\mathbf{1}) - \mathbf{S}$. In fact, the sparsities of $\mathbf{S}$ and $\mathbf{L}$ are the same except the diagonal, but the properties of two matrices are very different. Our model demonstrates two key differences from the two methods. First, (16) has additional term for modeling the data generation process. If $\gamma = 0$ and the absolute difference is used (see Section 3.2), the primal problem of (16) is equivalent to (34). $\gamma > 0$ is useful to retain the structure of data after dimensionality reduction. Second, our structure learning model DRTree and DDRTree takes the spanning trees as the candidate structure, which is very different from ℓ_1 regularization over $\mathbf{S}$ so that our structure learning methods can transform original points into embedded points that form spanning trees in the latent space.

6 Experiments

We perform extensive experiments to verify our proposed models, SPL and DDRTree, separately, by comparing them to various existing dimensionality reduction methods on a variety of synthetic and real world datasets.

6.1 Structured Projection Learning

6.1.1 Parameter Sensitivity Analysis

We investigate the parameter sensitivity of the proposed SPL method by varying γ and C on DistortedSShape, a synthetic data of 100 data points, which has been used in [46]. For simplicity, we study the influence of parameters by varying one and fixing the other. The neighborhood size is set to 10 and the reduced dimension is 2. We vary $\gamma \in \{0, 10^{-3}, 10^3\}$ and $C \in \{10^{-2}, 10^3, \infty\}$.

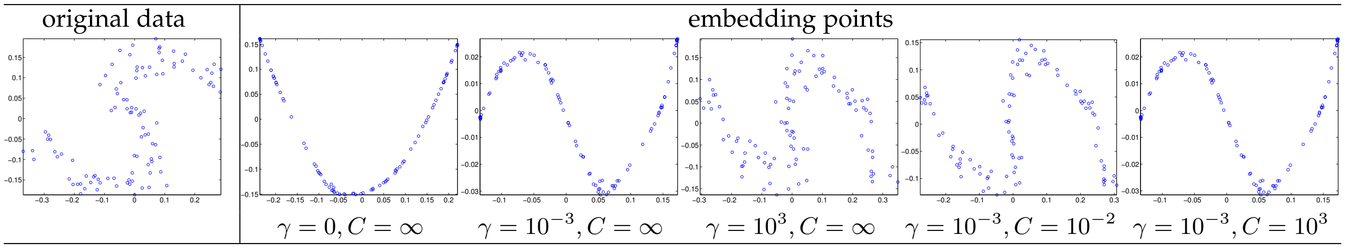


Fig. 1. The visualization results for parameter sensitivity analysis of SPL on DistortedSShape by either varying $\gamma \in \{0, 10^{-3}, 10^{3}\}$ with prefixed $C = \infty$, or varying $C \in \{10^{-2}, 10^{3}, \infty\}$ with $\gamma = 10^{-3}$ using the neighborhood size as 10 in a 2-D space. For the results of parameter sensitivity analysis on other parameters, please see them in the supplementary material.

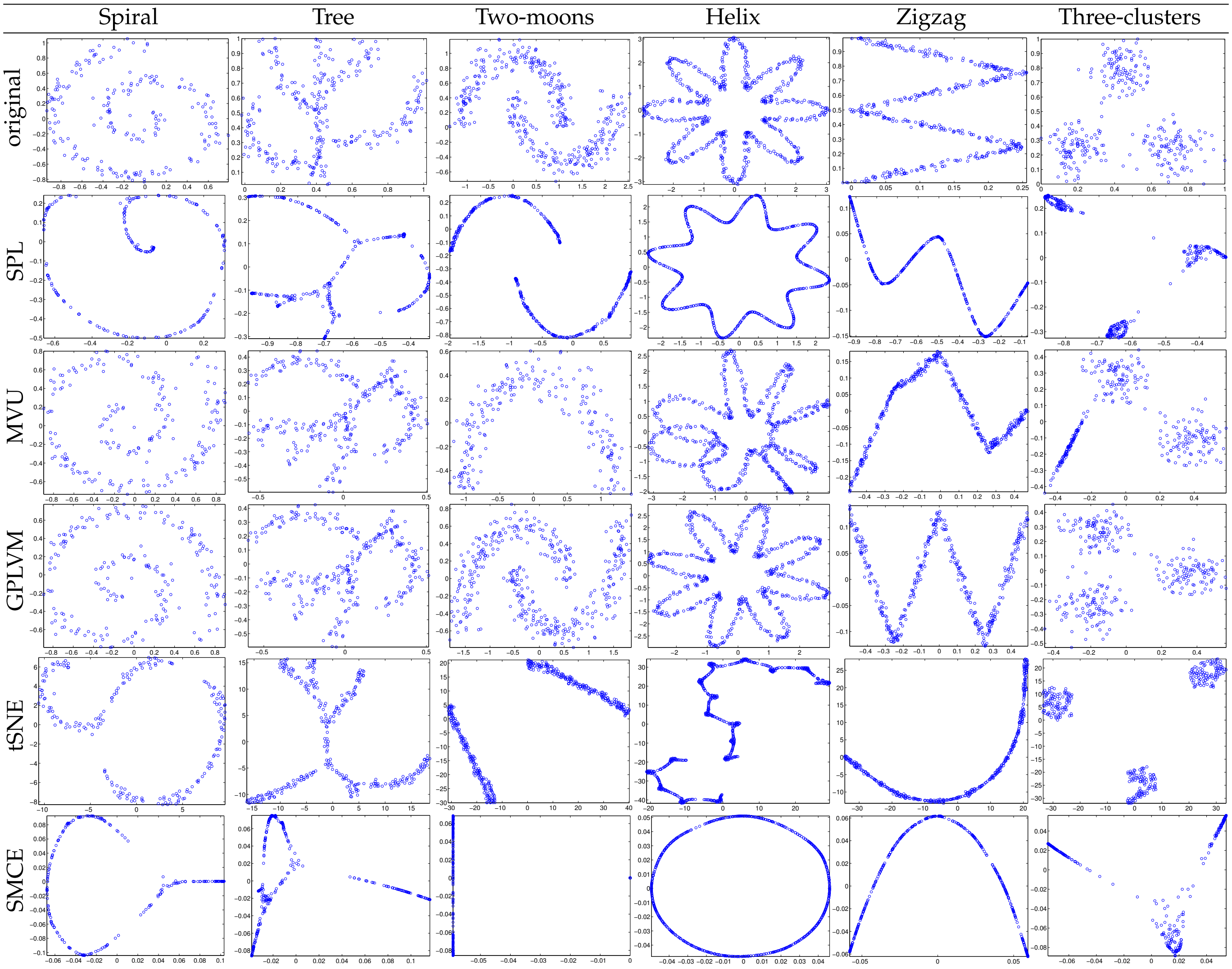


Fig. 2. The visualization results of embedding points using five methods on six synthetic data. Due to the space limit, the visualization results of the remaining four methods are reported in the supplementary materials.

Fig. 1 shows the original data and the resulting embeddings using SPL by varying γ and C. We made the following observations from Fig. 1: (i) SPL with a small γ can obtain embedding points with less noises (smooth S-shape skeleton); (ii) SPL with $\gamma = 0$ changes the spatial intrinsic structure from S-shape to V-shape structure, while the true structures can be retained by introducing the generative model using a small γ; (iii) a large C is preferred since it does not allow a large violation of pairwise distances from original data to the embedding data; (iv) SPL can achieve relatively good embeddings in a large interval of γ and C. All these observations are consistent with algorithmic analysis in Section 3.4.

6.1.2 *Synthetic Data*

We conduct experiments for data visualization on six synthetic datasets by comparing the proposed SPL method with various dimensionality reduction methods, including KPCA [8], MVU [12], MEU [19], LE [11], LLE [10], GPLVM [17], tSNE [20], SMCE [24]. Two datasets, *Spiral* and *Zigzag*[2], are used, which are also used in the principal curve learning [46]

[2] https://www.lri.fr/∼kegl/researchUdeM/research/pcurves /implementations/Samples/

and the underlying structures are spiral and zigzag curves, respectively. Dataset *2moons*[3] is used in manifold-based semi-supervised learning [22] for distinguishing two classes, so the true skeleton structure consists of two smooth curves. Following the work [4], we generate a 3-D dataset, *helix* by using drtoolbox, where the data point $\mathbf{y}_i$ is computed by $\mathbf{y}_i = [(2+\cos(8p_i))\cos(p_i), (2+\cos(8p_i))\sin(p_i), \sin(8p_i)]$ where p_i is a random number that is sampled from a uniform distribution with support $[0,1]$. Two datasets, *Tree* and *Three-clusters* are from [47], where the underlying structures are tree and three clusters with weak connections, respectively. Each data is projected to a 2-D space for data visualization by the above mentioned methods. Gaussian kernel is used for kernel-based methods or methods with prefixed similarity measurement. We set $C = 10^3$ and $\gamma = 10^{-3}$ for SPL for all synthetic datasets. The same neighborhood size $K = 10$ is used for all neighborhood based methods.

Fig. 2 shows the visualization results of embedding points in a 2-D space obtained by five methods on six synthetic datasets. More comparing results are presented in the supplementary materials due to the space limit. We made the following observations. Some methods, such as KPCA, MVU, LLE and GPLVM, cannot tolerate noise of data so that the learned embeddings do not show a smooth skeleton structure. Some methods like MEU, LE, tSNE and SMCE can learn a smooth skeleton structure from noise data, but most of them are not consistent with the underlying structures. For example, only tSNE and SPL can obtain two separate curves from two-moons data, but the curves learned by SPL are much smoother and visually are more similar to moon-shaped structures than tSNE. On helix and zigzag, most methods obtain the embeddings that are very different from the true skeleton structures. Only SPL can obtain smooth skeleton structures by retaining the inherent structures on all six datasets. These observations imply that SPL is able to learn a smooth skeleton structure, and simultaneously retain the underlying structure of data in a 2-D space.

6.1.3 Visualization of Two Real Datasets

A collection of 400 teapot images from [48] are used[4] for the purpose of visualization. These images are taken successively as a teapot was rotated 360°. Our goal is to unveil the circular structure in 3-D space that organizes the 400 images. Each image consists of 76×101 RGB pixels and is represented as a vector of size $23,028$. Thus, this data is high-dimensional. We run our algorithm with parameters $\gamma \in \{0, 10^{-3}\}$, $C = 10^3$ and $K = 5$. The visualization results are reported in Fig. 3. A two-dimensional representation of the same set of teapot images is given in [48], where MVU can also successfully arrange these images in a circle (see Figure 3 in [48]). This is the same as our result with $\gamma = 0$. However, when $\gamma = 10^{-3}$, the visualization of embeddings is slightly different, although the similar circular structure can be achieved.

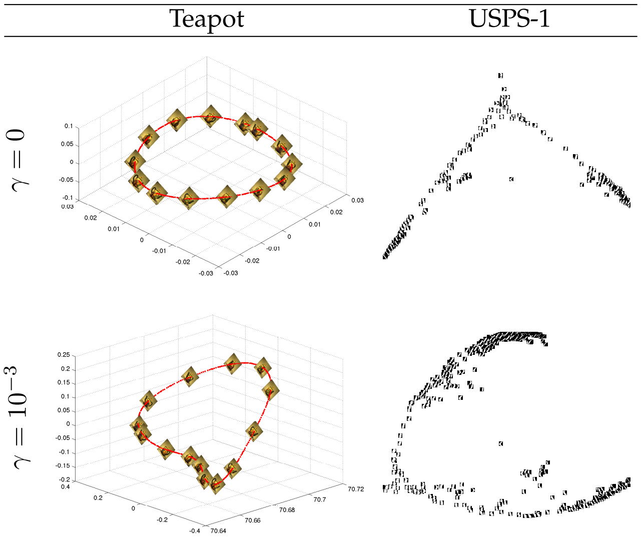


Fig. 3. Visualization results of SPL with parameters $\gamma \in \{0, 10^{-3}\}$ and $C = 10^3$ on Teapot and USPS with tag 1.

Another data is USPS handwritten digits[5], which contains handwritten digits from 0 to 9 with different written styles. Each one is a gray image of size 16×16. The vectorization of each image with label "1" is used to study the embeddings in a 2-D space since these images demonstrate very clear written styles. There are $1,100$ images in total. Following the same setting as did for teapot data, the visualization results of USPS data are shown in Fig. 3, where each image is shown in the position located by its associated embedded point in 2-D space. We observe that the images of "1"s are sorted in an order such that the angular degree of image "1"s is changed continuously, and the perpendicular ones are shown in the middle of the learned skeleton structure. The visualization difference between $\gamma = 0$ and $\gamma = 10^{-3}$ is that the latter demonstrates a smoother change than the former since the latter forms an arc while the former is a right angle.

6.1.4 Classification Performance of Embeddings

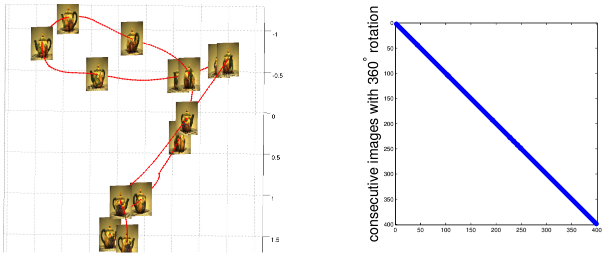
As shown in Table 1, ten datasets taken from the UCI and Statlib repositories are used to evaluate classification performance of embedded points learned by baseline methods same as those used in the experiments on synthetic data. The reduced dimensionality of data is shown in Table 1 by preserving 95% of energy of data. Following [12], we use the leave-one-out cross validation accuracy as the criterion for evaluating one-nearest neighbor classifier on the embeddings learned by these baseline methods. For methods that require K-nearest neighbor graph as the input, we tune $K \in \{5, 10, 15, 20, 30, 50\}$. We tune the parameter $\lambda \in \{0.01, 0.1, 1, 10\}$ for SMCE. Other parameters are set as the default values in the drtoolbox [6]. In addition, we tune $C = \{10, 10^3\}$ and $\gamma \in \{0, 10^{-3}\}$. The best results are reported for every baseline methods by tuning their own parameters.

Table 1 shows the leave-one-out cross validation accuracy of one-nearest neighbor classifier over the embeddings learned by nine methods on ten benchmark datasets. It is clear to see that SPL is competitive to tSNE in terms of

[3] manifold.cs.uchicago.edu/manifold_regularization/2moons.mat
[4] http://www.cc.gatech.edu/~lsong/data/teapotdata.zip
[5] http://www.cs.nyu.edu/~roweis/data/usps_all.mat
[6] https://lvdmaaten.github.io/drtoolbox/

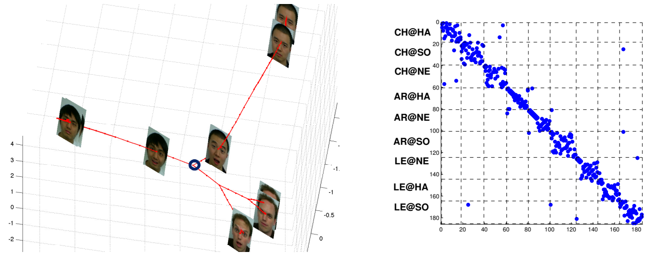
TABLE 1
The leave-one-out cross validation accuracy of one-nearest neighbor classifier over ten datasets. N is the number of data points. c is the true number of clusters. D is the original dimensionality and d is the reduced dimensionality. The best results are in bold.

	Iris	CMU-PIE	COIL20	Isolet	Pendigits	Satimage	USPS	Vehicle	Segment	Letter
(N, c)	(150, 3)	(3329, 68)	(1440, 20)	(3119, 2)	(3498, 10)	(4435, 6)	(2007, 10)	(846, 4)	(231,7)	(5000, 26)
(D, d)	(4, 2)	(1024, 39)	(2014,84)	(617,165)	(16, 9)	(36, 6)	(256, 32)	(18, 6)	(19,7)	(16,12)
LLE [10]	0.9467	0.9655	0.9965	0.9298	0.9760	0.8570	0.9013	0.6537	0.9623	0.8960
LE [11]	0.9133	0.6248	0.9833	0.9368	0.9714	0.8586	0.9023	0.5981	0.9398	0.7436
MVU [12]	0.6533	0.4662	0.7660	0.8035	0.9737	0.8607	0.7693	0.5579	0.9342	0.6042
KPCA [8]	0.9000	0.2701	0.5583	0.7086	0.9883	0.8462	0.3303	0.5615	0.9550	0.8488
GPLVM [17]	0.9333	0.9787	**1.0000**	0.8410	0.9866	0.8884	0.5944	0.5898	0.9688	0.8974
MEU [19]	0.8867	0.9507	**1.0000**	0.9349	0.9840	0.8652	0.9312	0.6407	0.9537	0.1244
SMCE [24]	0.9400	0.9612	**1.0000**	0.9314	0.9806	0.8848	0.9307	0.6832	0.9398	0.8790
tSNE [20]	**0.9600**	0.9751	**1.0000**	0.9468	0.9909	**0.9037**	0.9292	0.6738	0.9671	**0.9134**
SPL	**0.9600**	**0.9805**	**1.0000**	**0.9497**	**0.9943**	0.8938	**0.9317**	**0.6915**	**0.9680**	0.9054



(a) curve (b) adjacency matrix

Fig. 4. Experimental results of DDRTree applied to Teapot images. (a) principal curve generated by DDRTree. Each dot represents one teapot image. Images following the principal curve are plotted at intervals of 30 for visualization. (b) The adjacency matrix of the curve follows the ordering of the 400 consecutive teapot images with $360°$ rotation.



(a) tree (b) adjacency matrix

Fig. 5. Experimental results of our DDRTree method performed on facial expression images. (a) A hierarchical tree generated by DDRTree. Each dot represents one face image. Images of three types of facial expressions from three subjects are plotted for visualization. The black circle is the root of the hierarchical structure; (b) The adjacency matrix of the tree on nine blocks indicates that each block corresponds to one facial expression of one subject.

classification accuracy, and demonstrates much better than the others. As shown in [20], tSNE helps to achieve good classification performance by learning a new embedding of original data. The learning criterion of tSNE is better suitable for clustering/classification, but it is not appropriate for learning skeleton structures in a latent space as observed in Section 6.1.2 and Section 6.1.3. These results imply that SPL is not only suitable for learning skeleton structures in latent spaces from high-dimensional data, but also it can achieve competitive or better classification performance on the learned embedding points by comparing with various existing methods.

6.2 Learning Tree Structures From Real Datasets

We investigate the ability of our proposed DDRTree method to automatically discover tree structures from three real-world datasets. The latent tree structures of these three datasets include principal curves, hierarchical tree structures, and a cancer progression path.

6.2.1 Principal Curve

The same teapot data in Section 6.1.3 is used to justify our tree structure learning methods. Similar to [49], the data in each dimension is normalized to have zero mean and unit standard deviation.

We run our proposed DDRTree method using the kernel matrix as the input. We set $\lambda = 0.1 \times N$ and $d = 36$ that keeps 95% of total energy. The experimental results of DDRTree are shown in Figure 4. The principal curve (Figure 4(a)) is shown in terms of the first 3 columns of the learned projection matrix $\mathbf{W}$ as the coordinates where each dot represents one image. The sampled images at intervals of 30 are plotted for the purpose of visualization. Figure 4(b) shows the linear chain dependency among teapot images following the consecutive rotation process. We can see that the curve generated by our method is consistent with the rotating process of the 400 consecutive teapot images.

A similar result is also recovered by the clustering via Hilbert Schmidt independence criterion (CLUHSIC) [49], which assumes that the label kernel matrix is a ring structure [49]. However, there are three main differences. First, we learn a projection space where images are arranged in the form of a principal curve, while CLUHSIC applies KPCA to transform original data to an orthogonal space where clustering is performed. Second, the principal curve generated by our DDRTree method is much smoother than that obtained by CLUHSIC (see Figure 4 in [49]). Third, our method learns the adjacency matrix from the given dataset, but CLUHSIC requires a label matrix as *a prior*. We attempted to run MVU by keeping 95% energy, i.e., $d = 36$. However, storage allocation fails due to the large memory requirement of solving a semidefinite programming problem in MVU. Hence, MVU cannot be used to learn a relatively large dataset. However, our method does not have this issue.

6.2.2 Hierarchical Tree

Facial expression data[7] is used for hierarchical clustering, which takes into account both the identities of individuals and the emotion being expressed [49]. This data contains 185 face images (308×217 RGB pixels) with three types of facial expressions (NE: neutral, HA: happy, SO: shock) taken from three subjects (CH, AR, LE) in an alternating order, with

[7] http://www.cc.gatech.edu/~lsong/data/facedata.zip

around 20 repetitions each. Eyes of these facial images have been aligned, and the average pixel intensities have been adjusted. As with the teapot data, each image is represented as a vector, and is normalized in each dimension to have zero mean and unit standard deviation.

DDRTree is applied to this dataset with $\lambda = 0.1 \times N$ and $d = 185$. The experimental results are shown in Figure 5. We can clearly see that three subjects are connected through different branches of a tree. If we take the black circle in Figure 5(a) as the root of the hierarchy, the tree forms a two-level hierarchical structure. As shown in Figure 5(b), all three facial expressions from three subjects are also clearly separated. A similar two-level hierarchy is also recovered by CLUHSIC (Figure 3(b) in [49]). However, the advantages of using DDRTree discussed above for teapot images are also applied here. In addition, we can observe more detailed information from the tree structure. For example, LE@SO is the junction to other two subjects, i.e., AR@SO and CH@SO, which can be observed from the 9th row of the adjacency matrix 5(b). This observation suggests that the shock is the most similar facial expression among three subjects. However, CLUHSIC is not able to obtain this information.

6.2.3 Cancer Progression Path

We are particularly interested in studying human cancer, a dynamic disease that develops over an extended time period. Once initiated from a normal cell, the advance to malignancy can to some extent be considered a Darwinian process - a multistep evolutionary process - that responds to selective pressure [50]. The disease progresses through a series of clonal expansions that result in tumor persistence and growth, and ultimately the ability to invade surrounding tissues and metastasize to distant organs. As shown in Figure 6(a), the evolution trajectories inherent to cancer progression are complex and branching [50]. Due to the obvious necessity for timely treatment, it is not typically feasible to collect time series data to study human cancer progression [28]. However, as massive molecular profile data from excised tumor tissues (static samples) accumulates, it becomes possible to design integrative computation analyses that can approximate disease progression and provide insights into the molecular mechanisms of cancer. We have previously shown that it is indeed possible to derive evolutionary trajectories from static molecular data, and that breast cancer progression can be represented by a high-dimensional manifold with multiple branches [51].

We interrogate a large-scale, publicly available breast cancer dataset [52] for cancer progression modeling. The dataset contains the expression levels of over $25,000$ gene transcripts obtained from 144 normal breast tissue samples and $1,989$ tumor tissue samples. By using a nonlinear regression method, a total of 359 genes were identified that may play a role in cancer development [51]. In the analysis, we set $\lambda = 5 \times N$ and $d = 80$ that retains 90% of energy.

Figure 6(b) shows the learned latent structures and latent points in a reduced dimensional space. Each tumor sample is colored with its corresponding PAM50 subtype label, a molecular approximation that uses a 50-gene signature to group breast tumors into five subtypes including normal-like, luminal A, luminal B, HER2+ and basal [53]. Basal and HER2+ subtypes are known to be the most aggressive breast tumor types. The learned graph structure in the low-dimensional space suggests a linear bifurcating progression path, starting from the normal tissue samples, and diverging to either luminal A or basal subtypes. The linear trajectory through luminal A continues to luminal B and to the HER2+ subtype. Significant side-branches are evident for both luminal A and luminal B subtypes, suggesting that these subtypes can be further delineated. The revealed data structure is consistent with the proposed branching architecture of cancer progression shown in Figure 6.

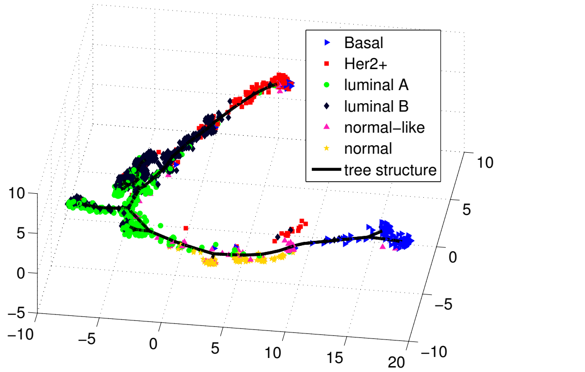


Fig. 6. Graph structure learned by DDRTree on breast cancer dataset with $d = 80$ and visualized in three-dimensional space spanned by the first three components of the learned projection matrix.

7 Conclusion

In this paper, we proposed a general probabilistic framework for dimensionality reduction, which not only takes the noise of data into account, but also utilizes the neighborhood graph as the locality information. Based on this framework, we presented a model that can learn a smooth skeleton of embedding points from high-dimensional, noisy data. In order to learn an explicit graph structure, we developed another new dimensionality reduction method that learns a latent tree structure and low-dimensional feature representation simultaneously. We extended the proposed method for clustering problems by imposing the constraint that data points belonging to the same cluster are likely to be close along the learned tree structure. The experimental results demonstrated the effectiveness of the proposed methods for recovering inherent structures from real-world datasets. Dimensionality reduction via learning a graph is formulated from a general graph, so the development of new dimensionality reduction methods for various specific structure is also possible.

References

[1] C. Ding and X. He, "K-means clustering via principal component analysis," in *ICML*, 2004.

[2] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[3] J. B. Tenenbaum, V. deSilva, and J. C. Landford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[4] L. Van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," *Tilburg University Technical Report, TiCC-TR 2009-005*, 2009.

[5] J. A. Lee and M. Verleysen, "Quality assessment of dimensionality reduction: Rank-based criteria," *Neurocomputing*, vol. 72, no. 7, pp. 1431–1443, 2009.

[6] J. T. Jolliffe, *Principal Component Analysis*. Springer-Verlag, Berlin, 1986.

[7] C. J. C. Burges, "Dimension reduction: a guided tour," *FTML*, vol. 2, no. 4, pp. 275–365, 2009.

[8] B. Schölkopf, A. Smola, and K. Muller, "Kernel principal component analysis," *Advances in Kernel Methods - Support Vector Learning*, pp. 327–352, 1999.

[9] L. Cayton, "Algorithms for manifold learning," UCSD, Tech. Rep., 2005.

[10] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *JMLR*, vol. 4, pp. 119–155, 2003.

[11] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering." in *NIPS*, vol. 14, 2001, pp. 585–591.

[12] K. Weinberger, F. Sha, and L. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *ICML*, 2004, p. 106.

[13] S. Lafon and A. B. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization," *IEEE T-PAMI*, vol. 28, no. 9, pp. 1393–1403, 2006.

[14] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *PNAS*, vol. 100, no. 10, pp. 5591–5596, 2003.

[15] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM Journal on Scientific Computing*, vol. 26, no. 4, pp. 313–338, 2005.

[16] M. Tipping and C. Bishop, "Probabilistic principal component analysis," *JRSS: Series B*, vol. 61, no. 3, pp. 611–622, 1999.

[17] N. D. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *JMLR*, vol. 6, pp. 1783–1816, 2005.

[18] M. K. Titsias and N. D. Lawrence, "Bayesian gaussian process latent variable model," in *AISTATS*, 2010, pp. 844–851.

[19] N. D. Lawrence, "A unifying probabilistic perspective for spectral dimensionality reduction: Insights and new models," *JMLR*, vol. 13, no. 1, pp. 1609–1638, 2012.

[20] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *JMLR*, vol. 9, no. 2579-2605, p. 85, 2008.

[21] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *NIPS*, 2002, pp. 833–840.

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

[23] M. Maier and U. Luxburg, "Influence of graph construction on graph-based clustering measures," in *NIPS*, 2009.

[24] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *NIPS*, 2011, pp. 55–63.

[25] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in *Database Theory-ICDT'99*, 1999.

[26] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang, "Learning with ℓ_1-graph for image analysis," *IEEE TIP*, vol. 19, no. 4, pp. 858–866, 2010.

[27] B. Lake and J. Tenenbaum, "Discovering structure by learning sparse graph," in *Proceedings of the 33rd Annual Cognitive Science Conference*. Citeseer, 2010.

[28] Q. Mao, L. Yang, L. Wang, S. Goodison, and Y. Sun, "SimplePPT: A simple principal tree algorithm," in *SDM*, 2015.

[29] Q. Mao, L. Wang, S. Goodison, and Y. Sun, "Dimensionality reduction via graph structure learning," in *ACM SIGKDD*, 2015, pp. 765–774.

[30] B. Scholkopf and A. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[31] L. Xiao, J. Sun, and S. Boyd, "A duality view of spectral methods for dimensionality reduction," in *ICML*, 2006, pp. 1041–1048.

[32] A. K. Gupta and D. K. Nagar, *Matrix variate distributions*. CRC Press, 1999, vol. 104.

[33] J. Zhu, N. Chen, and E. Xing, "Bayesian inference with posterior regularization and applications to infinite latent svms," *JMLR*, vol. 15, no. 1, pp. 1799–1847, 2014.

[34] M. Dudik, S. Phillips, and R. Schapire, "Maximum entropy density estimation with generalized regularization and an application to species distribution modeling," *JMLR*, vol. 8, pp. 1217–1260, 2007.

[35] L. Wang, Q. Mao, and I. W. Tsang, "Latent smooth skeleton embedding." in *AAAI*, 2017, pp. 2703–2709.

[36] X. Qiu, Q. Mao, Y. Tang, L. Wang, R. Chawla, H. Pliner, and C. Trapnell, "Reversed graph embedding resolves complex single-cell developmental trajectories," *bioRxiv*, p. 110668, 2017.

[37] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," lecture notes of EE392o, Stanford University, Autumn Quarter 2003-2004.

[38] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *COLT*, 2003, pp. 144–158.

[39] M. Cheung, "Minimum-cost spanning trees," http://people.orie.cornell.edu/dpw/orie6300/fall2008/Recitations/rec09.pdf.

[40] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recogn*, vol. 41, pp. 176–190, 2008.

[41] C. M. Bishop, M. Svensén, and C. K. I. Williams, "GTM: the generative topographic mapping," *Neural Comput*, vol. 10, no. 1, pp. 215–234, 1998.

[42] R. Tibshirani, "Principal curves revisited," *Stat Comput*, vol. 2, pp. 183–190, 1992.

[43] Q. Mao, I. Tsang, S. Gao, and L. Wang, "Generalized multiple kernel learning with data-dependent priors," *IEEE T-NNLS*, vol. 29, no. 6, pp. 1134–1148, 2015.

[44] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE T-PAMI*, vol. 17, no. 8, pp. 790–799, 1995.

[45] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *JMLR*, vol. 6, pp. 1817–1853, 2005.

[46] B. Kégl, A. Kryzak, T. Linder, and K. Zeger, "Learning and design of principal curves," *IEEE T-PAMI*, vol. 22, no. 3, pp. 281–297, 2000.

[47] J. Yao, Q. Mao, S. Goodison, V. Mai, and Y. Sun, "Feature selection for unsupervised learning through local learning," *Pattern Recognition Letters*, vol. 53, pp. 100–107, 2015.

[48] K. Q. Weinberger and L. K. Saul, "An introduction to nonlinear dimensionality reduction by maximum variance unfolding," in *AAAI*, 2006.

[49] L. Song, A. Smola, A. Gretton, and K. Borgwardt, "A dependence maximization view of clustering," in *ICML*, 2007.

[50] M. Greaves and C. C. Maley, "Clonal evolution in cancer," *Nature*, vol. 481, no. 7381, pp. 306–313, 2012.

[51] Y. Sun, J. Yao, N. Nowak, and S. Goodison, "Cancer progression modeling using static sample data," *Genome Biol*, vol. 15, no. 8, p. 440, 2014.

[52] C. Curtis, S. P. Shah, S. Chin *et al.*, "The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups," *Nature*, vol. 486, no. 7403, pp. 346–352, 2012.

[53] J. Parker, M. Mullins, M. Cheang *et al.*, "Supervised risk predictor of breast cancer based on intrinsic subtypes," *J Clin Oncol*, vol. 27, no. 8, pp. 1160–1167, 2009.

Li Wang is currently an assistant professor with Department of Mathematics, University of Texas at Arlington, Texas, USA. She worked as a research assistant professor with Department of Mathematics, Statistics, and Computer Science at University of Illinois at Chicago, Chicago, USA from 2015 t0 2017. She worked as the Postdoctoral Fellow at University of Victoria, BC, Canada in 2015 and Brown University, USA, in 2014. She received her Ph.D. degree in Department of Mathematics at University of California, San Diego, USA, in 2014. She received the master degree in Computational Mathematics from Xi'an Jiaotong University, Shaanxi, China, in 2009 and the Bachelor degree in Information and Computing Science from China University of Mining and Technology, Jiangsu, China in 2006. Her research interests include large scale optimization, polynomial optimization and machine learning.

Qi Mao is currently a senior researcher at HERE North America LLC, Chicago, USA. He received his PhD in 2014 with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. He received the Master degree with Computer Science from Nanjing University, Nanjing, in 2009 and the Bachelor degree with Computer Science from Anhui University, Hefei, in 2005. His research interests include machine learning, data mining, and large-scale data analysis.