SimplePPT: A Simple Principal Tree Algorithm

Qi Mao* Le Yang† Li Wang‡ Steve Goodison§ Yijun Sun¶

Abstract

Many scientific datasets are of high dimension, and the analysis usually requires visual manipulation by retaining the most important structures of data. Principal curve is a widely used approach for this purpose. However, many existing methods work only for data with structures that are not self-intersected, which is quite restrictive for real applications. To address this issue, we develop a new model, which captures the local information of the underlying graph structure based on reversed graph embedding. A generalization bound is derived that show that the model is consistent if the number of data points is sufficiently large. As a special case, a principal tree model is proposed and a new algorithm is developed that learns a tree structure automatically from data. The new algorithm is simple and parameter-free with guaranteed convergence. Experimental results on synthetic and breast cancer datasets show that the proposed method compares favorably with baselines and can discover a breast cancer progression path with multiple branches.

Keywords: principal graph, reversed graph embedding, principal curve, cancer progression path

1 Introduction

In many fields of science, one often encounters observations represented as high-dimensional vectors sampled from unknown distributions. It is sometimes difficult to directly analyze data in the original space, and is desirable to perform data dimensionality reduction or associate data with some structured objects. One example is the study of human cancer, which is a dynamic disease that develops over an extended time period through the accumulation of a series of genetic alterations. The delineation of this dynamic process would provide critical insights into molecular mechanisms underlying the disease process, and inform the development of diagnostics, prognostics and targeted therapeutics. The recently developed high-throughput genomics technology has made it possible to measure the expression levels of all genes in tissues from thousands of tumor samples simultaneously. However, the delineation of the cancer progression path embedded in a high-dimensional genomics space remains a challenging problem [25].

Principal component analysis (PCA) [15] is one of the most commonly used methods to visualize data in a low-dimensional space, but its linear assumption limits its general applications. Several nonlinear approaches based on the kernel trick have been proposed [23], but they remain sub-optimal for detecting complex structures. Alternatively, if data dimensionality is very high, manifold learning based on the local information of data can be effective. Examples include locally linear embedding (LLE) [22] and Laplacian eigenmaps [1]. However, these methods generally require to construct a carefully tuned neighborhood graph as their performance heavily depends on the quality of constructed graphs.

Another approach is principal curve, which was initially proposed as a nonlinear generalization of the first principal component line [14]. Informally, a principal curve is an infinitely differentiable curve with a finite length that passes through the middle of data. Several principal-curve approaches have been proposed, including those that minimize certain types of risk functions such as the quantization error [14, 17, 24, 21, 12] and the negative log-likelihood function [26, 3]. To overcome the over-fitting issue, regularization is generally required. Kégl et al. [17] bounded the total length of a principal curve, and proved that the principal curve with a bounded length always exists if the data distribution has a finite second moment. Similar results were obtained by bounding the turns of a principal curve [21]. More recently, the elastic maps approach [12] was proposed that regularizes the elastic energy of a membrane. An alternative definition of a principal curve based on a mixture model was considered in [14], where the model parameters are learned through maximum likelihood estimation and the regularization is achieved using the smoothness of coordinate functions. Genera-

*Bioinformatics Laboratory, The State University of New York at Buffalo, Buffalo, NY 14201, USA. maoq1984@gmail.com

†Department of Computer Science and Engineering, The State University of New York at Buffalo, Buffalo, NY 14201, USA. lyang25@buffalo.edu

‡The Institute for Computational and Experimental Research in Mathematics (ICERM), Brown University, Providence, RI 02912, USA. liwangucsd@gmail.com

§Department of Health Sciences Research, Mayo Clinic, Jacksonville, FL 32224, USA. goodisonsteve@gmail.com

¶Department of Microbiology and Immunology, The State University of New York at Buffalo, Buffalo, NY 14201, USA. yijunsun@buffalo.edu

tive topographic mapping (GTM) [3] was proposed to maximize the posterior probability of the data which is generated by a low-dimensional discrete grid mapped into the original space and corrupted by additive Gaussian noise. GTM provides a principled alternative to the self-organizing map (SOM) [18] for which it is impossible to define an optimality criterion [9].

Methods for learning a principal curve have been widely studied, but they are generally limited to learn a structure that does not intersect itself [14]. Only a few methods can handle complex principal objects. Kégl and Krzyzak [16] extended their polygonal line method [17] for skeletonization of handwritten digits. The principal manifold approach [11] extends the elastic maps approach [12] to learn a graph structure generated by graph grammar. A major drawback of the two methods is that they require either a set of predefined rules (specifically designed for handwritten digits [16]) or grammars with many parameters to be tuned, which makes their implementations complicated and their adaptations to new datasets difficult. More importantly, their convergences are not guaranteed. Recently, a subspace constrained mean shift (SCMS) method [19] was proposed that can obtain principal points for any given second-order differentiable density function, but it is still not trivial to obtain a predicted structure.

In this paper, we propose a new regularized principal graph model that addresses some of the aforementioned limitations. As a showcase, we develop a principal tree approach to learning a tree structure and principal points simultaneously. The main contributions of this paper are summarized as follows:

- By reversing the intuition of manifold learning, we define reversed graph embedding for the representation of a principal graph. The new representation can be interpreted as the length of a principal graph. We propose a new principal graph model by minimizing a relaxed quantization error with a boundedness constraint on the length of the principal graph. A generalization bound is also derived.

- To learn the graph structure from data, a principal tree model is presented. We then propose a simple algorithm to learn the principal points and the tree structure simultaneously. Theoretical and empirical convergence analyses are presented.

- Extensive experiments are conducted on a variety of synthetic datasets and a high-dimensional breast cancer gene expression dataset. Experimental results demonstrate that the proposed principal tree method performs better than baselines, and can recover the underlying structures of given datasets.

2 Regularized Principal Graph

We propose a new model for principal graph learning. Motivated by manifold learning, a new regularizer is presented for capturing the graph structure of a given dataset. The generalization bound is also derived.

2.1 Reversed Graph Embedding. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V} = \{V_1, \ldots, V_M\}$ is a set of vertices and $\mathcal{E}$ is a set of edges. Suppose that every vertex V_m corresponds to a point $\mathbf{z}_m \in \mathcal{Z} \subset \mathbb{R}^d$, which lies on a manifold with an intrinsic dimension d. Let $\mathcal{X} \subset \mathbb{R}^D$ be the input space and $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N \subset \mathcal{X}$ be a given dataset. We consider learning a function $f_\mathcal{G} \in \mathcal{F}$ and $f_\mathcal{G} : \mathcal{Z} \to \mathcal{X}$ over $\mathcal{G}$ that maps the intrinsic space $\mathcal{Z}$ to the input space $\mathcal{X}$.

Given a graph $\mathcal{G}$, denote as $w_{i,j}$ the weight of edge (V_i, V_j), where $w_{i,j}$ represents the similarity value (or connection indicator) between $\mathbf{z}_i$ and $\mathbf{z}_j$ in the intrinsic space $\mathcal{Z}$. Intuitively, if $\mathbf{z}_i$ and $\mathbf{z}_j$ are neighbors on $\mathcal{G}$ with a high degree of similarity, $f_\mathcal{G}(\mathbf{z}_i)$ and $f_\mathcal{G}(\mathbf{z}_j)$ are also close to one another. To capture this intuition, we consider the following optimization problem

$$\min_{f_\mathcal{G} \in \mathcal{F}} \min_{\mathbf{z}_1, \ldots, \mathbf{z}_M} \sum_{(V_i, V_j) \in \mathcal{E}} w_{i,j} ||f_\mathcal{G}(\mathbf{z}_i) - f_\mathcal{G}(\mathbf{z}_j)||^2. \tag{2.1}$$

The above formulation has several interesting properties. First, problem (2.1) is a reverse thinking of Laplacian eigenmap [1]. If vertices V_i and V_j are close on $\mathcal{G}$, which means that $\mathbf{z}_i$ and $\mathbf{z}_j$ has a high degree of similarity $w_{i,j}$, data points $f_\mathcal{G}(\mathbf{z}_i)$ and $f_\mathcal{G}(\mathbf{z}_j)$ in $\mathcal{X}$ are also close. On the contrary, in the Laplacian eigenmap, the similarity $v_{i,j}$ between $\mathbf{x}_i$ and $\mathbf{x}_j$ is computed in $\mathcal{X}$ to capture the local information of the manifold, while the distance between $\mathbf{z}_i$ and $\mathbf{z}_j$, $||\mathbf{z}_i - \mathbf{z}_j||$, are computed in $\mathcal{Z}$. Specifically, Laplacian eigenmap solves the following optimization problem,

$$\min_{\mathbf{z}_1, \ldots, \mathbf{z}_M} \sum_{i,j} v_{i,j} ||\mathbf{z}_i - \mathbf{z}_j||^2.$$

Weights $w_{i,j}$ and $v_{i,j}$ are computed in different dimensional spaces, so they represent distinct kinds of locality information. Based on the above discussion, we thus name the formulation specified in (2.1) as reversed graph embedding.

Second, the optimal function $f_\mathcal{G}^* \in \mathcal{F}$ obtained by solving (2.1) is related to harmonic or pluriharmonic functions. This can be further illustrated by the following observations. Let $\mathcal{N}_m$ be the neighbors of a point $\mathbf{z}_m, \forall m$. For any given $\mathbf{z}_m$, problem (2.1) can be rewritten as

$$\min_{f_\mathcal{G}(\mathbf{z}_m)} \sum_{j \in \mathcal{N}_m} w_{m,j} ||f_\mathcal{G}(\mathbf{z}_m) - f_\mathcal{G}(\mathbf{z}_j)||^2,$$

which has an analytic solution by fixing the rest of

variables $\{f_{\mathcal{G}}(\mathbf{z}_j)\}_{j\neq m}$:

$$f_{\mathcal{G}}(\mathbf{z}_m) = \frac{1}{\sum_{j\in\mathcal{N}_m} w_{m,j}} \sum_{j\in\mathcal{N}_m} w_{m,j} f_{\mathcal{G}}(\mathbf{z}_j). \tag{2.2}$$

If (2.2) holds for all m, function $f_{\mathcal{G}}$ is a harmoinc function on $\mathcal{G}$ since its value in each nonterminal vertex is the mean of the values in the closest neighbors of this vertex [12]. It is easier to incorporate any neighborhood structure existing in $\mathcal{G}$ into the proposed formulation than plurihamonic graphs defined in [12], since it imposes penalty only on a subset of k-stars as

$$||f_{\mathcal{G}}(\mathbf{z}_m) - \frac{1}{\sum_{j\in\mathcal{N}_m} w_{m,j}} \sum_{j\in\mathcal{N}_m} w_{m,j} f_{\mathcal{G}}(\mathbf{z}_j)||^2, \tag{2.3}$$

where $|\mathcal{N}_m| = k, \forall m$. The connection of $f_{\mathcal{G}}$ to harmonic or pluriharmonic functions enriches the learned function $f_{\mathcal{G}}$.

Third, reversed graph embedding facilitates the learning of a graph structure from data. The weight $w_{i,j}$ encodes the similarity or connection between V_i and V_j on $\mathcal{G}$. Taking the binary encoding as an example, $w_{i,j} = 1$ if $i \in \mathcal{N}_j$ and $j \in \mathcal{N}_i$, and 0 otherwise. In most cases, a dataset is given, but graph $\mathcal{G}$ is unknown. Hence, it is necessary to automatically learn $\mathcal{G}$ from data. The objective function of reversed graph embedding is linear with respect to the weights $\{w_{i,j}\}_{i.j=1}^M$. This linearity property benefits the learning of the graph structure. However, principal elastic map [12] is not suitable for the same purpose since the variables $\{w_{i,j}\}_{i.j=1}^M$ are coupled in the problem of minimizing (2.3).

2.2 Regularized Principal Graph. Given a graph $\mathcal{G}$ with edge weights $\{w_{i,j}\}_{i.j=1}^M$, points $\{\mathbf{z}_i\}_{i=1}^M$ and a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, we propose to learn a mapping function by minimizing the empirical quantization error with a constrained functional class:

$$\min_{f_{\mathcal{G}}\in\mathcal{F}_{\mathcal{G},\ell}} \frac{1}{N}\sum_{i=1}^N \min_{\mathbf{z}\in\mathcal{Z}} c(\mathbf{x}_i, f_{\mathcal{G}}(\mathbf{z})), \tag{2.4}$$

where $c(\mathbf{x}, \mathbf{x}') = ||\mathbf{x} - \mathbf{x}'||^2$ is the square loss function. The constrained functional class $\mathcal{F}_{\mathcal{G},\ell}$ is defined as

$$\mathcal{F}_{\mathcal{G},\ell} = \{f_{\mathcal{G}} \in \mathcal{F},\ \ell(\mathcal{G}) \leq \ell\}, \tag{2.5}$$

where the constraint is defined in terms of the objective function of the reversed graph embedding (2.1) as

$$\ell(\mathcal{G}) = \sum_{(V_i,V_j)\in\mathcal{E}} w_{i,j}||f_{\mathcal{G}}(\mathbf{z}_i) - f_{\mathcal{G}}(\mathbf{z}_j)||^2. \tag{2.6}$$

It is worth noting that the quantity $\ell(\mathcal{G})$ can be considered as the length of a principal graph. In the case where $\mathcal{G}$ is a linear chain structure, $\ell(\mathcal{G})$ is the same as the length of a polygonal line defined in [17]. However, the (2.4) is more flexible than principal curves since the graph structure allows self-intersection. For principal graph learning, elastic map [12] also defines a penalty based on a given graph. However, based on the discussion of the third property of reversed graph embedding, it is difficult to solve problem (2.4) with respect to both the function $f_{\mathcal{G}}$ and the graph weights $\{w\}_{i.j=1}^M$ within the elastic-maps framework. In contrast, the proposed constraint based on reversed graph embedding leads to a simple and efficient algorithm to learn a principal tree model with guaranteed convergence. This will be clarified in Section 3.

2.3 Uniform Convergence Bound. In this section, we determine a bound on the sample size sufficient to ensure that given a graph $\mathcal{G}$ and intrinsic points $\{\mathbf{z}_m\}_{m=1}^M$ we can find an $f_{\mathcal{G}} \in \mathcal{F}_{\mathcal{G},\ell}$ close to the best by solving problem (2.4).

Let $p(\mathbf{x})$ be an unknown probability distribution where $\mathbf{x} \in \mathcal{X}$. We find a function $f_{\mathcal{G}}$ by minimizing the expected quantization error

$$\mathrm{R}[f_{\mathcal{G}}] = \int_{\mathcal{X}} \min_{\mathbf{z}\in\mathcal{Z}} c(\mathbf{x}, f_{\mathcal{G}}(\mathbf{z}))\mathrm{d}p(\mathbf{x}). \tag{2.7}$$

Given a dataset $\mathcal{D}$, the expected quantization error is unknown, so we instead minimize its empirical quantization error estimated from $\mathcal{D}$ as

$$\mathrm{R}_{emp}[f_{\mathcal{G}}] = \frac{1}{N}\sum_{i=1}^N \min_{\mathbf{z}\in\mathcal{Z}} c(\mathbf{x}_i, f_{\mathcal{G}}(\mathbf{z})). \tag{2.8}$$

Following the work of [24], we define the function as

$$f_{\mathcal{G}}(\mathbf{z}) = \sum_{i=1}^M \beta_i \kappa(\mathbf{z}_i, \mathbf{z}), \mathbf{z}_i \in \mathcal{Z}, \beta_i \in \mathcal{X}\ , \tag{2.9}$$

where $\kappa : \mathcal{Z}\times\mathcal{Z} \to \mathbb{R}$ is a kernel function and $\{\beta_i\}_{i=1}^M$ are model parameters to be learned.

For any kernel function κ with a positive integral operator $(T_k f)(\mathbf{z}) = \int_{\mathcal{Z}} f(\mathbf{z}')\kappa(\mathbf{z}', \mathbf{z})\mathrm{d}(\mathbf{z}')$, one can write $\kappa(\mathbf{z}, \mathbf{z}') = \sum_i \lambda_i\phi_i(\mathbf{z})\phi_i(\mathbf{z}')$, where (λ_i, ϕ_i) is the eigensystem of the integral operator T_k. Let $\mathcal{F}^c_{\mathcal{G},\ell} = \{(\mathbf{x}, \mathbf{z}) \to c(\mathbf{x}, f_{\mathcal{G}}(\mathbf{z})) : f_{\mathcal{G}} \in \mathcal{F}_{\mathcal{G},\ell}\}$. We have the following rates of convergence for the optimal estimates. The proof of Theorem 2.1 is given in Appendix A.

THEOREM 2.1. *Let $\mathcal{F}^c_{\mathcal{G},\ell}$ be a class of continuous functions from $\mathcal{Z}$ to $\mathcal{X} \subseteq U_\tau$ and p be a distribution over $\mathbb{R}^D$. Let $f^*_{\mathcal{G},emp} = \arg\min_{f_{\mathcal{G}}\in\mathcal{F}^c_{\mathcal{G},\ell}} R_{emp}[f_{\mathcal{G}}]$ and $f^*_{\mathcal{G}} = \arg\min_{f_{\mathcal{G}}\in\mathcal{F}^c_{\mathcal{G},\ell}} \mathrm{R}[\mathrm{f}_{\mathcal{G}}]$. Suppose that $\mathcal{F}^c_{\mathcal{G},\ell}$ is compact. If N points are drawn i.i.d. from p, then for all $\eta > 0$ and $\varepsilon \in (0, \eta/2)$, we have*

$$\begin{aligned}&\mathbb{P}\{|\ \mathrm{R}[f^*_{\mathcal{G},emp}] - \mathrm{R}[f^*_{\mathcal{G}}]\ | > \eta\}\\ \leq& 2\left(\mathcal{N}\left(\frac{\varepsilon}{l_c}, \mathcal{F}_{\mathcal{G},\ell}, L_\infty(\ell_2^d)\right) + 1\right)\exp\left(-\frac{2N(\eta-\varepsilon/2)^2}{e_c}\right),\end{aligned}$$

where l_c is the Lipschitz constant of cost function $c(\cdot)$, $\mathcal{N}\left(\frac{\varepsilon}{2l_c}, \mathcal{F}_{\mathcal{G},\ell}, L_\infty(\ell_2^d)\right)$ is the $\frac{\varepsilon}{2l_c}$ covering number of $\mathcal{F}_{\mathcal{G},\ell}$ given a metric $L_\infty(\ell_2^d)$ defined as

$$L_\infty(\ell_2^d)(f_\mathcal{G}, f'_\mathcal{G}) = \sup_{\mathbf{z}\in R^d} ||f_\mathcal{G}(\mathbf{z}) - f'_\mathcal{G}(\mathbf{z})||_2.$$

Meanwhile, if $\lambda_j = \mathcal{O}(e^{-\alpha j^p})$ with $\alpha, p > 0$, we have $\log\mathcal{N}\left(\varepsilon, \mathcal{F}_{\mathcal{G},\ell}, L_\infty(\ell_2^d)\right) = \mathcal{O}\left(\log^{\frac{p+1}{p}}\left(\frac{1}{\varepsilon}\right)\right)$. If $\lambda_j = \mathcal{O}(j^{-(\alpha+1)})$ for some $\alpha > 0$, and for any $\delta \in (0, \alpha/2)$, we have $\log\mathcal{N}\left(\varepsilon, \mathcal{F}_{\mathcal{G},\ell}, L_\infty(\ell_2^d)\right) = \mathcal{O}(\varepsilon^{-2/\alpha+\delta})$.

As shown in Theorem 2.1, the covering number $\mathcal{N}(\varepsilon, \mathcal{F}_{\mathcal{G},\ell}, L_\infty(\ell_2^d))$ is independent of the sample size N. Therefore, the union bound in Theorem 2.1 vanishes as the number of samples tends to infinity. Hence, the proposed model is consistent.

3 A Principal Tree Learning Algorithm

Given a graph $\mathcal{G}$ and a set $\{\mathbf{z}_m\}_{m=1}^M$, constructing a principal graph model by minimizing the empirical quantization error can find an $f_\mathcal{G} \in \mathcal{F}^c_{\mathcal{G},\ell}$ close to the best possible if the number of data points is sufficiently large. However, the graph structure of $\mathcal{G}$ is generally unknown. In order to apply the theoretical results of Theorem 2.1 to principal graph learning, we propose to automatically learn $\mathcal{G}$ and $\{f_\mathcal{G}(\mathbf{z}_i)\}_{i=1}^M$ from data simultaneously. We below consider learning a tree structure as a showcase.

3.1 Principal Tree Model. According to Section 2.1, a number of variables need to be optimized, including the optimal function $f_\mathcal{G}$, a set of intrinsic points $\{\mathbf{z}_m\}_{m=1}^M$ and a graph $\mathcal{G}$. Instead of learning a function $f_\mathcal{G}$ and $\{\mathbf{z}_m\}_{m=1}^M$ separately, we can define $\tilde{\mathcal{Z}} = \{1, \ldots, M\}$ and $f_\mathcal{G} : m \to \mathbf{f}_{\mathcal{G}_m}$, where $\mathbf{f}_{\mathcal{G}_m} \in \mathcal{X}$, and learn a set of principal points $\{f_\mathcal{G}(\mathbf{z}_m)\}$ for every m. The canonical distortion error of a vector quantizer can be written as

$$\widehat{\mathrm{R}}[f_\mathcal{G}] = \int_{\mathbb{R}^D} \min_{m\in\{1,\ldots,M\}} ||\mathbf{x} - \mathbf{f}_{\mathcal{G}_m}||^2 \mathrm{d}p(\mathbf{x}). \tag{3.10}$$

To obtain M centroids $\{\mathbf{f}_{\mathcal{G}_m}\}_{m=1}^M$, minimizing the empirical distortion error is equivalent to solving the K-means problem. The hard partition obtained by K-means, however, is sensitive to noise, outliers, or some data points that cannot be thought of as belonging to a single cluster [10]. Soft partition methods such as Gaussian mixture modeling have also been used in modeling principal curves [3, 26]. However, the likelihood of a Gaussian mixture model tends to be infinite when a singleton is formed [26].

To alleviate the problems suffered by the aforementioned methods, we propose to minimize a relaxed empirical quantization error given by

$$\widehat{\mathrm{R}}_{emp}[f_\mathcal{G}] = \min_{r\in\mathcal{B}_r} \frac{1}{N}\sum_{i=1}^N\sum_{m=1}^M r_{i,m}||\mathbf{x}_i - \mathbf{f}_{\mathcal{G}_m}||^2 + \sigma\Omega(r)\ , \tag{3.11}$$

where $\mathcal{B}_r = \{r : \sum_{m=1}^M r_{i,m} = 1, r_{i,m} \geq 0, \forall i, \forall m\}$, $\Omega(r) = \frac{1}{N}\sum_{i=1}^N\sum_{m=1}^M r_{i,m}\log r_{i,m}$ 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, and is also used in fuzzy K-means for clustering problems [2].

By replacing $\mathrm{R}_{emp}[f_\mathcal{G}]$ in (2.4) with $\widehat{\mathrm{R}}_{emp}[f_\mathcal{G}]$ and considering $\mathcal{G}$ as a tree structure, the principal graph learning (2.4) can be reformulated as the following optimization problem

$$\min_{f_\mathcal{G}, b\in\mathcal{B}_b} \widehat{\mathrm{R}}_{emp}[f_\mathcal{G}], \quad \text{s.t.} \sum_{(V_i,V_j)\in\mathcal{E}} b_{i,j}||\mathbf{f}_{\mathcal{G}_i} - \mathbf{f}_{\mathcal{G}_j}||^2 \leq \ell\ ,$$

where the convex set of a relaxed minimum spanning tree is denoted by $\mathcal{B}_b = \{b : \sum_{(i,j)\in\mathcal{E}} b_{i,j} = |\mathcal{V}| - 1, \sum_{i\in\mathcal{S}, j\in\mathcal{S}} b_{i,j} \leq |\mathcal{S}| - 1, \forall\mathcal{S} \subseteq \mathcal{V}, b_{i,j} \geq 0, \forall(i,j)\}$ [7], and the bounded length of a principal tree is $\ell \in \mathbb{R}^+$. The graph structure can then be recovered by the set of edges $\{(V_i, V_j) : b_{i,j} \neq 0\}$.

Instead of directly imposing a length constraint in problem (3.12), it is equivalent to minimizing the relaxed empirical quantization error with a regularization in the objective function. Hence, problem (3.12) can be reformulated as follows

$$\min_{\mathbf{f}_\mathcal{G}, b\in\mathcal{B}_b} \widehat{\mathrm{R}}_{emp}[f_\mathcal{G}] + \frac{\lambda}{2}\sum_{(V_i,V_j)\in\mathcal{E}} b_{i,j}||\mathbf{f}_{\mathcal{G}_i} - \mathbf{f}_{\mathcal{G}_j}||^2\ , \tag{3.12}$$

where $\lambda > 0$ is a parameter. For an appropriately selected λ, (3.12) and (3.12) are equivalent [24].

Next, we propose a simple algorithm to solve problem (3.12) and then present a convergence analysis.

3.2 Alternate Convex Search. To solve problem (3.12), we employ the alternate convex search method, which is frequently used to solve biconvex optimization problems [13].

We first show that problem (3.12) is a biconvex optimization problem. Let $\mathcal{B}_{f_\mathcal{G}} = \{\mathbf{f}_{\mathcal{G}_m} \in \mathbb{R}^D, \forall m\}$. Together with $\mathcal{B}_b$ and $\mathcal{B}_r$, we have three sets of variables, which are all convex and can be decoupled. We combine $\mathcal{B}_r$ and $\mathcal{B}_b$ by Cartesian product as $\mathcal{B}_{r,b} = \mathcal{B}_r \times \mathcal{B}_b = \{(r, b) : r \in \mathcal{B}_r, b \in \mathcal{B}_b\}$, which is still a convex set [4]. The objective function is jointly convex with respect to $(r, b) \in \mathcal{B}_{r,b}$. By the definition presented in [13], problem (3.12) is a biconvex problem.

Alternate convex search is a minimization method to solve a biconvex problem where the variable set

Algorithm 1 Principal Tree Learning Algorithm

Input: Data $\mathbf{X}$, parameters λ and σ, M
Initialize $\mathbf{F}_{\mathcal{G}}$
repeat
  $d_{i,j} = ||\mathbf{f}_{\mathcal{G}_i} - \mathbf{f}_{\mathcal{G}_j}||^2, \forall i, \forall j$
  Apply Kruskal's algorithm to obtain $\mathbf{B}$
  $\mathbf{L} = diag(\mathbf{B1}) - \mathbf{B}$
  Compute $\mathbf{R}$ via (3.14)
  $\Lambda = diag(\mathbf{R}^T\mathbf{1})$
  $\mathbf{F}_{\mathcal{G}} = \mathbf{XR}(\lambda\mathbf{L} + \Lambda)^{-1}$
until Convergence

can be divided into disjoint blocks [13]. The blocks of variables defined by convex subproblems are solved cyclically by optimizing the variables of one block while fixing the variables of all other blocks. In this way, each convex subproblem can be solved efficiently by using a convex minimization method.

Denote $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$, $\mathbf{F}_{\mathcal{G}} = [\mathbf{f}_{\mathcal{G}_1}, \ldots, \mathbf{f}_{\mathcal{G}_M}] \in \mathbb{R}^{D\times M}$, $\mathbf{R} \in \mathbb{R}^{N\times M}$ with the (i,m)th entry as $r_{i,m}$, diagonal matrix $\Lambda = diag(\mathbf{R}^T\mathbf{1})$, $\mathbf{B} \in \mathbb{R}^{M\times M}$ with the (m,m')th entry as $b_{m,m'}$, the Laplacian matrix $\mathbf{L} = diag(\mathbf{B1}) - \mathbf{B} \in \mathbb{R}^{M\times M}$, and $\mathbf{1}$ is a column vector with all entries as one. The problem (3.12) can be solved as follows.

Fix $\{\mathbf{R}, \mathbf{B}\}$ and solve $\mathbf{F}_{\mathcal{G}}$: Given $\{\mathbf{R}, \mathbf{B}\}$, the optimization problem for solving $\mathbf{F}_{\mathcal{G}}$ can be reformulated as an unconstrained quadratic programming problem

$$\min_{\mathbf{F}_{\mathcal{G}}} -2tr(\mathbf{F}_{\mathcal{G}}^T\mathbf{XR}) + tr(\Lambda\mathbf{F}_{\mathcal{G}}^T\mathbf{F}_{\mathcal{G}}) + \lambda tr(\mathbf{F}_{\mathcal{G}}\mathbf{L}\mathbf{F}_{\mathcal{G}}^T) \ ,$$

which has an analytic solution given by

$$\mathbf{F}_{\mathcal{G}} = \mathbf{XR}(\lambda\mathbf{L} + \Lambda)^{-1}. \tag{3.13}$$

It is worth noting that the principal tree algorithm can automatically adjust the local information from data through $\mathbf{L}$ constructed in each iteration, while most existing methods assume that the structure is predefined, that is, $\mathbf{L}$ is fixed in advance. The merit of avoiding tuning the neighborhood graph comes from the automatically learned graph structure. This is a major difference between our method and graph Laplacian based methods, e.g., Laplacian eigenmap [1].

Fix $\mathbf{F}_{\mathcal{G}}$ and solve $\{\mathbf{R}, \mathbf{B}\}$: Given $\mathbf{F}_{\mathcal{G}}$, the jointly convex optimization over $\{\mathbf{R}, \mathbf{B}\}$ can be decoupled into two convex optimization problems with respect to $\mathbf{B}$ and $\mathbf{R}$, respectively. To obtain $\mathbf{R}$, we solve the following constrained optimization problem

$$\min_{\mathbf{R}\in\mathcal{B}_r} \sum_{i=1}^{N}\sum_{m=1}^{M} r_{i,m}\left(||\mathbf{x}_i - \mathbf{f}_{\mathcal{G}_m}||^2 + \sigma \log r_{i,m}\right).$$

By applying the Lagrangian duality theorem [4], we can readily obtain the analytic solution

$$r_{i,m} = \frac{\exp\left(-||\mathbf{x}_i - \mathbf{f}_{\mathcal{G}_m}||^2/\sigma\right)}{\sum_{m=1}^{M}\exp\left(-||\mathbf{x}_i - \mathbf{f}_{\mathcal{G}_m}||^2/\sigma\right)}, \forall i, \forall m. \tag{3.14}$$

To obtain $\mathbf{B}$, we solve an LP relaxation for minimum spanning tree given by

$$\min_{\mathbf{B}\in\mathcal{B}_b} \sum_{(V_i,V_j)\in\mathcal{E}} b_{i,j}d_{i,j} \ ,$$

where $d_{i,j} = ||\mathbf{f}_{\mathcal{G}_i} - \mathbf{f}_{\mathcal{G}_j}||^2$. It can be approximately solved by Kruskal's algorithm [7]. The proposed methods is named as SimplePPT, the pseudo-code of which is given in Algorithm 1.

3.3 Convergence Analysis. Let $g : \mathcal{B} \to R$ be the objective function of problem (3.12), that is,

$$g(\mathbf{F}_{\mathcal{G}}, \mathbf{R}, \mathbf{B}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{M} r_{i,m}(||\mathbf{x}_i - \mathbf{f}_{\mathcal{G}_m}||^2 + \sigma \log r_{i,m}) + \frac{\lambda}{2}\sum_{(V_i,V_j)\in\mathcal{E}} b_{i,j}||\mathbf{f}_{\mathcal{G}_i} - \mathbf{f}_{\mathcal{G}_j}||^2$$

where $\mathcal{B} = \mathcal{B}_{F_{\mathcal{G}}} \times \mathcal{B}_r \times \mathcal{B}_b$. The objective function g is bounded from below by $-\sigma \log M$ because of the use of the negative entropy. Moreover, the objective function is continuous and differentiable on $\mathcal{B}$.

Let $\mathbf{y}^{(t)}$ be the vectorized representation of variables $\{\mathbf{F}_{\mathcal{G}}{}^{(t)}, \mathbf{R}^{(t)}, \mathbf{B}^{(t)}\}$. The following proposition states that both the objective values and variables converge. The sketched proof is provided in Appendix B.

PROPOSITION 3.1. *The sequence $\{g(\mathbf{y}^{(t)})\}_{t\in\mathbb{N}}$ generated by Algorithm 1 converges monotonically, and the variable sequence $\{\mathbf{y}^{(t)}\}_{t\in\mathbb{N}}$ also converges, that is,* $\lim_{t\to\infty} ||\mathbf{y}^{(t+1)} - \mathbf{y}^{(t)}|| = 0$.

By Proposition 3.1, we define the stopping criterion of Algorithm 1, which is the relative increase of the objective values (or the relative difference in $\mathbf{y}$) in two consecutive iterations. The empirical convergence results are given in Section 4.2.

3.4 Time Complexity. The time complexity of Algorithm 1 is determined by three individual parts. The first part is the complexity of running Kruskal's algorithm to construct a minimum spanning tree. It requires $\mathcal{O}(M^2D)$ for computing a fully connected graph and $\mathcal{O}(M^2 \log M)$ for finding a spanning tree. The second part is dominated by computing the soft assignments of samples, which has a complexity of $\mathcal{O}(NMD)$. The third part is dominated by the inverse of a matrix of size $M \times M$ that takes $\mathcal{O}(M^3)$ operations and

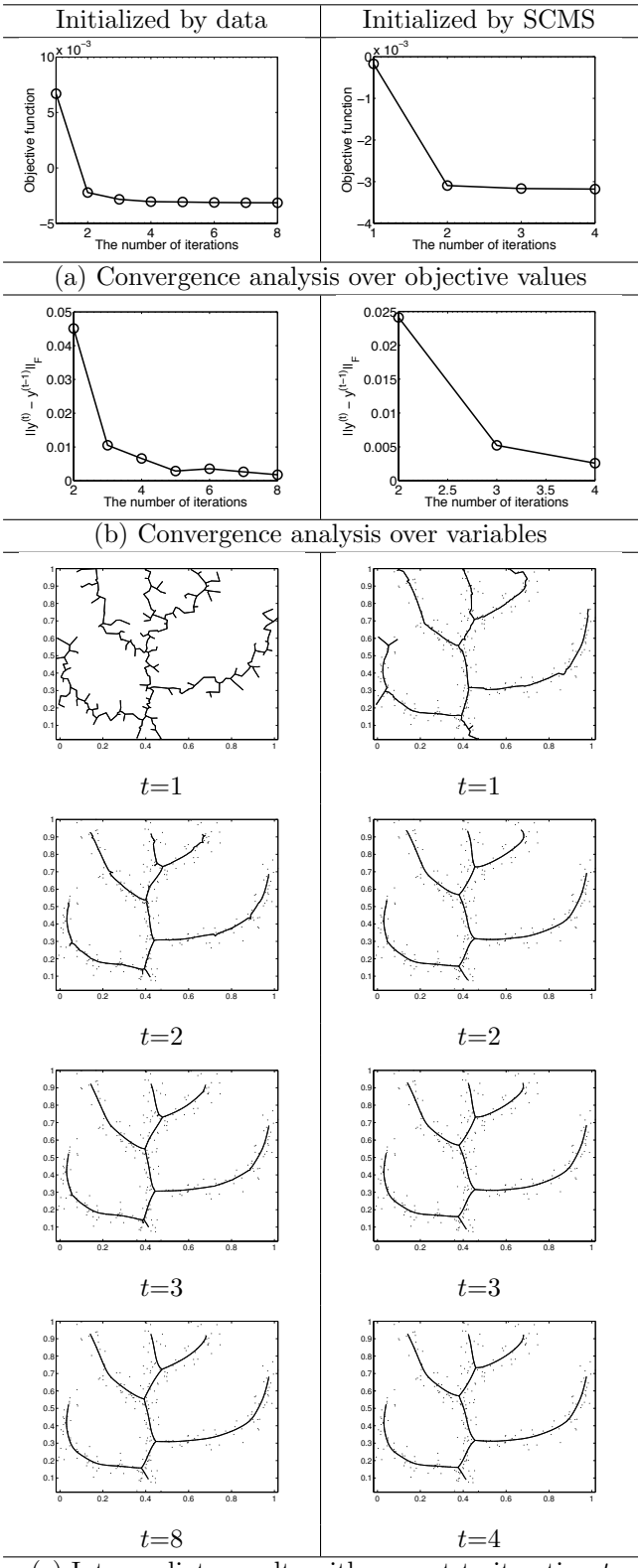
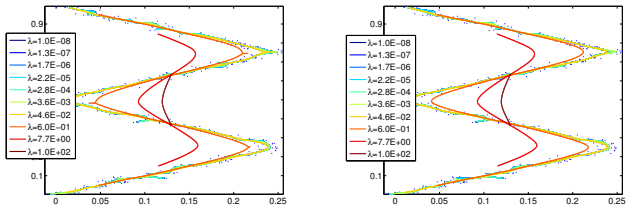


Figure 1: Convergence analyses and intermediate results of SimplePPT performed on the Tree dataset using two different initialization methods.



(a) Initialized by data (b) Initialized by SCMS

Figure 2: Results of sensitivity analysis of SimplePPT performed on the Zigzag dataset by using different λ. Two initialization strategies are used.

matrix multiplication that takes $DNM + DM^2$ operations. Therefore, the total complexity for each iteration is $\mathcal{O}(M^3 + DNM + M^2D)$. For the special case of $M = N$, the complexity becomes $\mathcal{O}(N^3 + DN^2)$. If the number of samples is large, a small number M is suggested for fast learning.

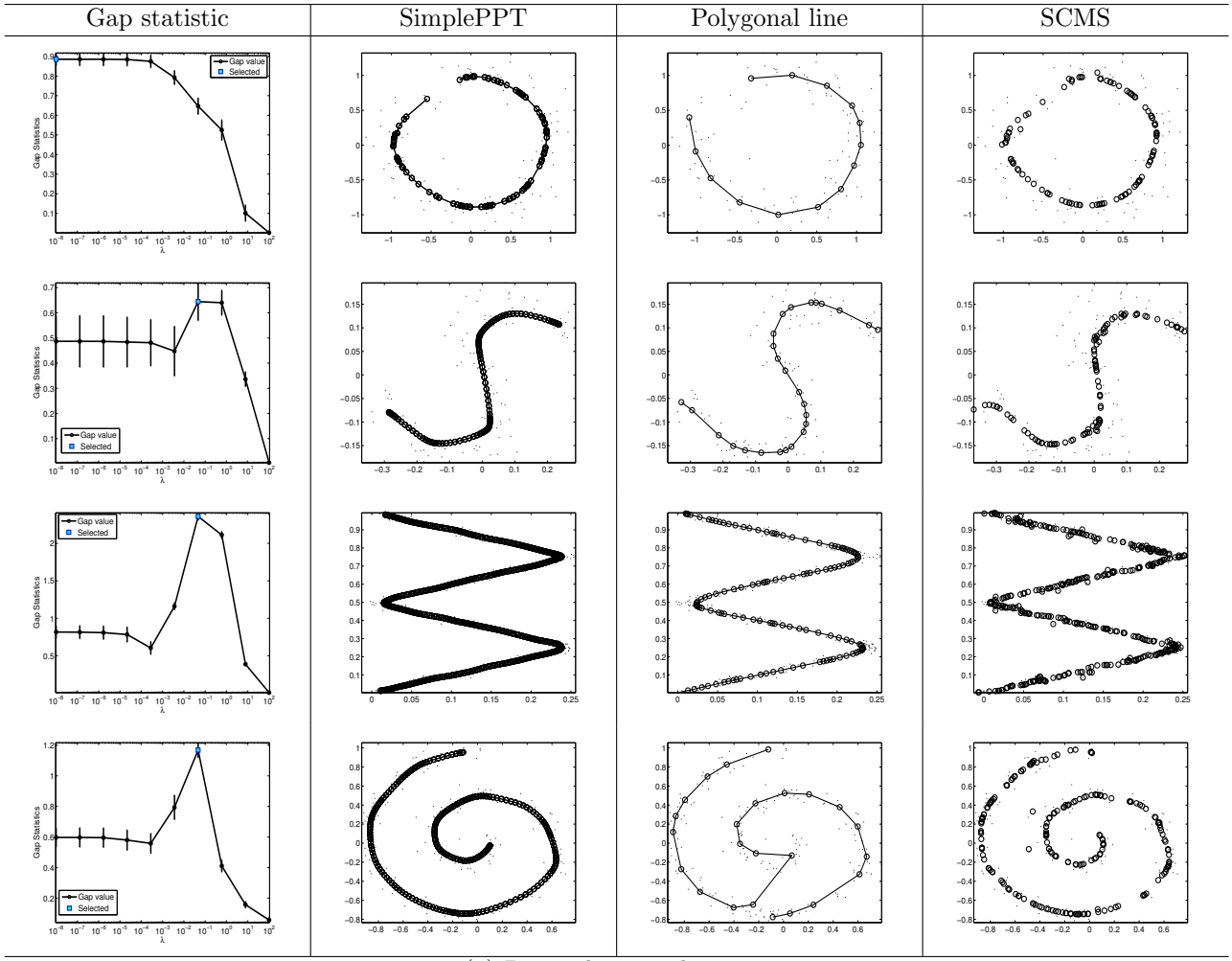
4 Experiments

We conduct an extensive experiment on both synthetic and real-world datasets to demonstrate the performance of the proposed method. We compare with two state-of-the-art baseline methods, namely the polygonal line method [17] and the SCMS method [19]. Since the goal of our method is to construct a principal tree or a principal curve from a given dataset, we do not apply our method to datasets with a general graph structure such as those with loops or disconnected components.

We first give a discussion on some implementation issues of the algorithm. Then, we present a convergence and parameter sensitivity analysis using synthetic data. Finally, we report the experimental results on various synthetic datasets and a breast cancer dataset.

4.1 Implementation Issues. The proposed algorithm have four parameters that need to be specified: the number of principal points M, the initialization matrix $\mathbf{F}_{\mathcal{G}}$, and two hyper-parameters σ and λ. In Appendix C, we establish the equivalence between a method minimizing the relaxed quantization error and the mean shift algorithm [6] (i.e., SCMS). This means that we can set $M = N$, and the initialization of principal points can be either the original data points or the principal points returned by SCMS. To estimate the parameter σ, we employ the leave-one-out-maximum likelihood criterion described in [19]. Alternatively, we can tune σ in SCMS and use the same σ in SimplePPT. We use the gap statistic method [27] to automatically tune parameter λ, which is detailed in Appendix D.

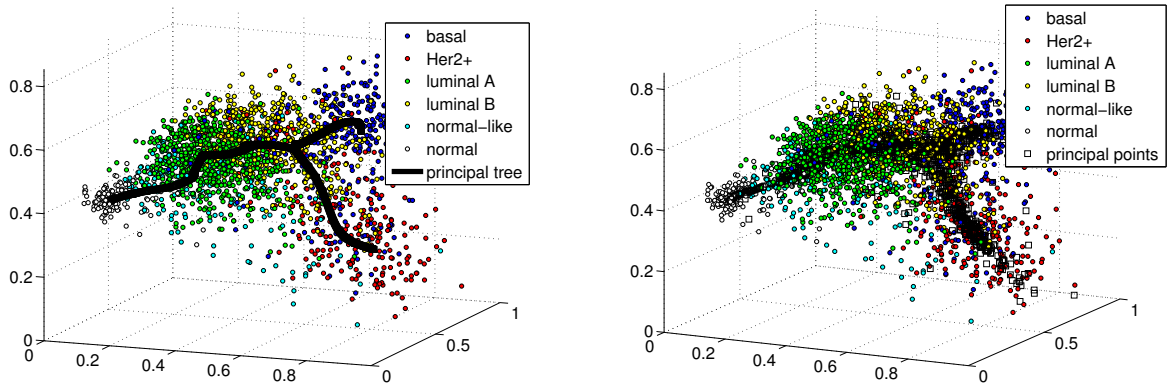
4.2 Convergence and Sensitivity Analysis. We perform a convergence analysis of Algorithm 1 using a synthetical tree dataset. Figures 1 shows the empirical



(a) Principle curve learning

(b) Principle tree learning

Figure 3: Results of three principal curve methods performed on six synthetic datasets. The first and second columns show the result of estimating the optimal λ by using gap statistics and the principal trees generated by SimplePPT using the optimal λ, respectively. The third and forth columns report the results generated by the polygonal line method and SCMS, respectively.



(a) SimplePPT (b) SCMS

Figure 4: Results of SimplePPT and SCMS applied to a breast cancer dataset.

convergence results obtained by using two initialization strategies discussed in Section 4.1 as well as the intermediate results. We observe that the proposed algorithm converges in less than 10 iterations, and the relative differences of variables between two consecutive iterations quickly converge to zero. This is consistent with the result of the theoretical analysis in Section 3.3. From Figure 1(c), we can see that when Algorithm 1 continues with more iterations, the tree structure becomes smoother. This empirically verifies the intuition of the reverse graph embedding.

We then perform a parameter sensitivity analysis by using the Zigzag data to demonstrate how the algorithm behaves with respect to different λ. Figure 2 shows the principal trees constructed by using ten different λ ranging from 10^{-8} to 10^{2}. It is clear that the larger λ is, the shorter the length of a principal tree is. Therefore, λ is an important parameter that controls the tradeoff between the curve fitting error and the length of a principal tree.

4.3 Synthetic Data. We evaluate the performance of SimplePPT by comparing with the polygonal line method [17] and SCMS [19] on six synthetic datasets. Among them, the first four datasets are also used in [17, 19]. The experiments are conducted in two settings. The first setting is to evaluate the three methods for principal curve learning, while the second setting is to construct tree structures of the datasets. In all experiments, we employ gap statistic to automatically tune the parameter λ. In the convergence analysis, we showed that the final results returned by using two initialization methods do not differ significantly, but using the result from SCMS as the initialization generally converges faster. Therefore, in the following experiments, we use the principal points returned by SCMS to initialize SimplePPT.

The first four rows of Figure 3 show the results from the principal curve learning setting. We have the following observations: 1) Gap statistic can effectively find parameter λ leading to reasonably good results on all four datasets. 2) The proposed method can obtain more smoothing curves than the other tested methods. 3) The polygonal line method fails on the Spiral data. We also see that SCMS cannot obtain a curve structure because many projected points do not have ordering information, and some points are scattered as shown in the Spiral data. This leaves a non-trivial problem to learn the underlying structure by using SCMS. Our proposed method does not have these problems. The last two rows of Figure 3 show the results obtained from the datasets containing tree structures. The polygonal line method fails on two datasets due to the principal curve assumption. The results are consistent with those obtained in the first setting.

The above results suggest that 1) our method can tune parameters automatically and obtain results at least as good as the baselines, and outperforms the polygonal line method that may fail on some datasets. 2) Our method can handle more complicated data structures than the polygonal line method, and provides detailed tree structures that cannot be trivially obtained by SCMS.

4.4 Breast Cancer Data. We finally apply our method to a breast cancer dataset to demonstrate its utility for solving real-world problems. The dataset is downloaded from [8] and contains the expression levels of over $25,000$ genes and the copy numbers of over $30,000$ genes from 144 normal and $1,989$ breast tumor samples. By using a non-linear regression method and clustering analysis, a total of $1,140$ genes were found to be associated with breast cancer progression [25]. We apply the principal tree methods to the data represented by the selected genes to recover the underlying data structure, which in this case represents the progression

path of breast cancer towards malignancy. For the purpose of visualization, the original data points and the learned principal points are projected onto a three-dimensional space spanning by the first three principal components of the data and are shown in Figure 4(a). For ease of discussion, each tumor sample is color-coded with its corresponding PAM50 subtype label, including normal-like, luminal A, luminal B, HER2+, and basal [20]. The learned data manifold suggests a linear bifurcating progression path for breast cancer progression, starting from the normal tissue samples, gradually transiting to luminal subtypes and finally forming a bifurcating structure leading to either HER2+ or basal subtypes. The latter two subtypes are known to be the most aggressive breast tumor types. This result is consistent with the result in a previous study [25]. In contrast, the principal points returned by SCMS does not have a clear progression structure (Figure 4(b)).

5 Conclusion

In this paper, we proposed a simple principal tree learning method, which can be used to obtain a set of principal points and a tree structure simultaneously. The experimental results demonstrated the effectiveness of the proposed method. Since our principal graph model are formulated from a general graph, the development of principal graph methods for other specific structure is also possible. In the future, we will explore principal graph learning on other graphs such as K-nearest neighbor graphs and apply it to other real-world datasets.

Acknowledgements

This work is supported in part by the National Science Foundation under grant number 1322212 and the SUNY Research Foundation.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2001.

[2] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[3] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: the generative topographic mapping. *Neural Comput*, 10(1):215–34, 1998.

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

[5] L. Cayton. Algorithms for manifold learning. Technical report, UCSD, 2005.

[6] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans Pattern Anal Mach Intell*, 17(8):790–9, 1995.

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

[8] C. Curtis, S. P. Shah, S. Chin, and et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–52, 2012.

[9] E. Erwin, K. Obermayer, and K. Schulten. Self-organizing maps: ordering, convergence properties and energy functions. *Biol Cybern*, 67:47–55, 1992.

[10] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta. A survey of kernel and spectral methods for clustering. *Pattern Recogn*, 41:176–90, 2008.

[11] A. Gorban, B. Kégl, D. Wunsch, and A. Zinovyev. Principal manifolds for data visualisation and dimension reduction, *Lecture Notes in Computational Science and Engineering*. Springer, 2007.

[12] A.N. Gorban and A. Y. Zinovyev. *Principal Graphs and Manifolds*, chapter 2, pages 28–59. IGI Global, Hershey, PA, USA, 2009.

[13] J. Gorski, F. Pfeuffer, and K. Klamroth. Biconvex sets and optimization with biconvex functions - a survey and extensions. *Math Method Oper Res*, 66:373–407, 2007.

[14] T. Hastie and W. Stuetzle. Principal curves. *JASA*, 84:502–16, 1989.

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

[16] B. Kégl and A. Kryzak. Piecewise linear skeletonization using principal curves. *IEEE Trans Pattern Anal Mach Intell*, 24(1):59–74, 2002.

[17] B. Kégl, A. Kryzak, T. Linder, and K. Zeger. Learning and design of principal curves. *IEEE Trans Pattern Anal Mach Intell*, 22(3):281–97, 2000.

[18] T. Kohonen. *Self-organizing Maps*. Springer, 1997.

[19] U. Ozertem and D. Erdogmus. Locally defined principal curves and surfaces. *JMLR*, 12:1249–86, 2011.

[20] J.S. Parker, M. Mullins, M.C. Cheang, and et. al. Supervised risk predictor of breast cancer based on intrinsic subtypes. *J Clin Oncol*, 27(8):1160-7, 2009.

[21] S. Sandilya and S. R. Kulkami. Principal curves with bounded turn. *IEEE Trans Inf Theory*, 48(10):2789–93, 2002.

[22] L. Saul and S. Roweis. Think globally, fit locally: unsupervised learning of low dimensional mainfolds. *JMLR*, 4:119–55, 2003.

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

[24] A. J. Smola, S. Mika, B. Schölkopf, and R. C. Williamson. Regularized principal manifolds. *JMLR*, 1:179–209, 2001.

[25] Y. Sun, J. Yao, N. Nowak, and S. Goodison. Cancer progression modeling using static sample data. *Genome Biol*, 15(8):440, 2014.

[26] R. Tibshirani. Principal curves revisited. *Statistics and Computation*, 2:183–90, 1992.

[27] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *J R Stat Soc*, 63(2):411–23, 2001.