A Unified Model for Probabilistic Principal Surfaces

Kui-yu Chang and Joydeep Ghosh, *Member*, *IEEE*

Abstract—Principal curves and surfaces are nonlinear generalizations of principal components and subspaces, respectively. They can provide insightful summary of high-dimensional data not typically attainable by classical linear methods. Solutions to several problems, such as proof of existence and convergence, faced by the original principal curve formulation have been proposed in the past few years. Nevertheless, these solutions are not generally extensible to principal surfaces, the mere computation of which presents a formidable obstacle. Consequently, relatively few studies of principal surfaces are available. Recently, we proposed the probabilistic principal surface (PPS) to address a number of issues associated with current principal surface algorithms. PPS uses a manifold oriented covariance noise model, based on the generative topographical mapping (GTM), which can be viewed as a parametric formulation of Kohonen's self-organizing map. Building on the PPS, we introduce a unified covariance model that implements PPS $(0 < \alpha < 1)$, GTM $(\alpha = 1)$, and the manifold-aligned GTM $(\alpha > 1)$ by varying the clamping parameter α. Then, we comprehensively evaluate the empirical performance (reconstruction error) of PPS, GTM, and the manifold-aligned GTM on three popular benchmark data sets. It is shown in two different comparisons that the PPS outperforms the GTM under identical parameter settings. Convergence of the PPS is found to be identical to that of the GTM and the computational overhead incurred by the PPS decreases to 40 percent or less for more complex manifolds. These results show that the generalized PPS provides a flexible and effective way of obtaining principal surfaces.

Index Terms—Principal curve, principal surface, probabilistic, dimensionality reduction, nonlinear manifold, generative topographic mapping.

1 Introduction

In many real world applications, it is often desirable to reduce the dimensionality of the original feature space for the problem at hand to alleviate the "curse-of-dimensionality" [1] and to obtain better generalization. It may also help in addressing practical issues, such as limited computational power and memory, or for data visualization needs. Dimensionality reduction via feature selection/extraction can be supervised or unsupervised. Supervised methods such as linear discriminant analysis [2] utilize additional information like class labels. On the other hand, unsupervised dimensionality reduction, which is the main focus of this paper, relies entirely on the input features.

Linear dimensionality reduction techniques such as principal component analysis (PCA) [3], factor analysis [3], independent component analysis [4], and projection pursuit [5] have been very well-studied in the past. Linear techniques are attractive for their simplicity and amenability to analysis, but may be inadequate for modelling highly nonlinear data. On the other end of the spectrum, researchers have proposed nonlinear methods such as generalized linear models [6], autoassociative neural networks [7], self-organizing maps [8], and principal surfaces [9], [10] for dimensionality reduction. Recently, mixture models [11], which probabilistically blend a number of overlapping linear models, have become popular as a compromise between linear and nonlinear methods [12], [13], [14]. The mixture approach enjoys some of the simplicity and analyzability of linear models while remaining robust enough to model nonlinear data (provided there are enough models of sufficient complexity to fit each localized data region well). Fig. 1 shows an example of each of the aforementioned dimensionality reduction methods when applied to artificially generated data.

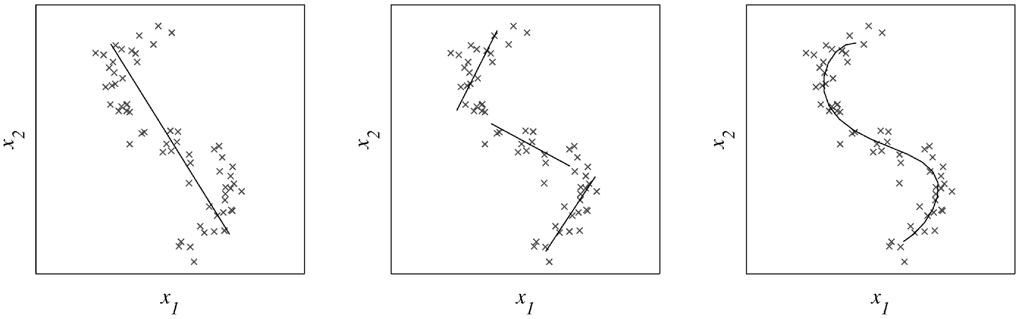
Among the nonlinear dimensionality reduction methods, principal surfaces[1] are the most attractive because they formalize the notion of a low-dimensional manifold passing through the "middle" of a data set, thereby generalizing principal components to the nonlinear domain. However, the original principal surface formulation [9] is not without its problems, stated as follows:

1. Existence cannot be guaranteed for arbitrary distributions.
2. Theoretical analysis is not as straightforward as with parametric models due to its nonparametric formulation.
3. It is inefficient for large sample size as all data points are needed to define a principal curve in practice.
4. It is biased at points of large curvature.
5. Convergence of the corresponding estimation algorithm cannot be guaranteed.

- *K.-y. Chang is with Interwoven, Inc., 1195 W. Freemont Ave., Sunnyvale, CA 94087. E-mail: kuiyu@lans.ece.utexas.edu.*
- *J. Ghosh is with the Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78712. E-mail: ghosh@ece.utexas.edu.*

Manuscript received 18 Nov. 1999; revised 19 July 2000; accepted 28 Aug. 2000.
Recommended for acceptance by I. Sethi.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110967.

[1] In this paper, the terms "surface" and "manifold" are used interchangeably to refer to manifolds of dimensionality 2 or greater.



(a) (b) (c)

Fig. 1. (a) Principal component analysis (PCA) showing the first principal axis, (b) mixture of three localized principal axes, (c) a principal curve.

Recently, we have proposed two improved estimation algorithms—probabilistic principal curve (PPC) and probabilistic principal surface (PPS), to address problems 2-4 for the case of principal curves and surfaces, respectively[2] [15]. The PPC estimates a principal curve with a cubic-spline smoothed mixture of oriented Gaussians, whereas the PPS incorporates oriented Gaussians into the generative topographical mapping (GTM) [16] framework to approximate a principal surface. The GTM itself is a parametric model of Kohonen's self-organizing map (SOM) [8]. Indeed, the SOM itself has been suggested [17], [18] to serve as a discrete approximation to principal surfaces. However, there are some significant differences between SOMs and principal surfaces that can lead to a poor solution in practice. For example, when the number of SOM nodes approaches the number of samples, the SOM degenerates into a highly irregular space-filling manifold [8]. On the contrary, the principal surface becomes increasingly smooth as the number of nodes increases! The GTM is better suited for estimating principal surfaces because its smoothness is largely determined by a mapping complexity parameter[3] and not by the number of nodes.

This paper introduces a generalized model that includes PPS and GTM as special cases. In Section 2, we formally state the problem of dimensionality reduction and review the current literature on principal curves, taking note of the advantages and disadvantages of each method. Section 3 summarizes and critiques existing algorithms for approximating principal surfaces. Section 4 describes our unified PPS model. Experimental results and commentaries on benchmark data sets are given in Section 5, where the sensitivities of the orientation parameter, manifold size, and mapping complexity with respect to reconstruction error are also analyzed. Section 6 discusses the experimental results. Finally, Section 7 concludes with a description of applications and directions for future research.

2 Principal Curves

2.1 Dimensionality Reduction

The problem of dimensionality reduction can be summarized as follows: Given N sample vectors $\{\mathbf{y}_n\}_{n=1}^N \subseteq \mathbb{R}^D$ drawn from the random vector $\vec{Y}$, find mappings $\mathcal{G} : \mathbb{R}^D \rightarrow \mathbb{R}^Q$ and[4] $\mathcal{F} :\rightarrow \mathbb{R}^D$ such that $\forall n = 1, \ldots, N$,

$$\mathcal{G}(\mathbf{y}_n) = \mathbf{x}_n, \tag{1}$$

$$\mathcal{F}(\mathbf{x}_n) = \hat{\mathbf{y}}_n \simeq \mathbf{y}_n, \tag{2}$$

where $\{\mathbf{x}_n\}_{n=1}^N \subseteq \mathbb{R}^Q$ denotes the corresponding set of reduced sample vectors drawn from the random vector $\vec{X}$ and D, Q denote the dimensionality of the original *data* and reduced *latent* spaces, respectively. The latent dimensionality Q is usually limited to 2 or 3 for visualization, otherwise, $Q \ll D$. The mappings $\mathcal{G}$ and $\mathcal{F}$ may be derived by optimizing one of several possible criteria such as maximum-likelihood or minimum mean square error (MSE). In PCA, for instance, both $\mathcal{G}$ and $\mathcal{F}$ are linear and the empirical reconstruction MSE is minimized. The forward mapping $\mathcal{G}$ for PCA can be computed via eigendecomposition of the sample covariance matrix and the derivation of $\mathcal{G}$ automatically leads to the corresponding reverse mapping $\mathcal{F}$. Similarly, latent variable models, such as factor analysis and independent component analysis, first compute $\mathcal{F}$, from which $\mathcal{G}$ can be obtained trivially using pseudoinverses. However, since an inverse mapping may not be easy to find for nonlinear transformations, usually $\mathcal{F}$ is first derived and $\mathcal{G}$ is then approximated by some projection operator.

2.2 Hastie and Stuetzle's Principal Curve

The principal curve was first defined by Hastie and Stuetzle [9] as a smooth (C^∞) unit-speed 1D manifold in $\mathbb{R}^D$ satisfying the *self-consistency* condition

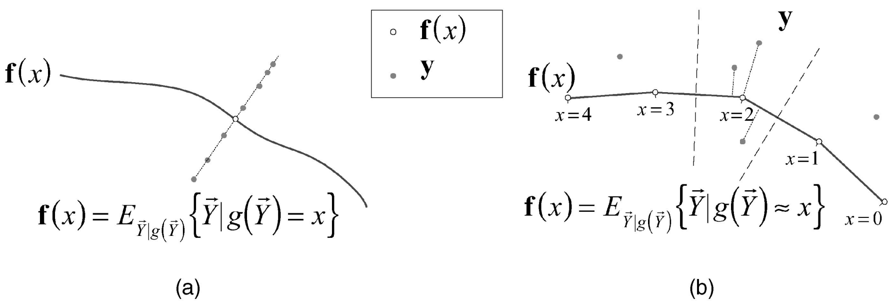
$$\mathbf{f}(x) = \mathrm{E}_{\vec{Y}|g(\vec{Y})}\left\{\vec{Y}|g\left(\vec{Y}\right) = x\right\}, \qquad \forall x \in \Lambda \subseteq \mathbb{R}, \tag{3}$$

where E is the conditional average operator and $g(\mathbf{y})$ is the projection operator given by

[2] Note that the principal curve is simply a 1D principal surface, but it is often singled out for investigation due to its simplicity. Consequently, algorithms derived for principal surfaces can be trivially applied to principal curves, but the reverse is not true in general.

[3] The size, shape, and type of latent basis functions also play a significant role in determining the complexity of the GTM. In this paper, we consider isotropic Gaussian latent basis functions uniformly laid out in latent space, with the width equal to twice the distance between adjacent bases.

[4] Some approaches, such as Sammon's projection, do not explicitly define the reverse mapping.


$\mathbf{f}(x)$
$\mathbf{f}(x) = E_{\vec{Y}|g(\vec{Y})}\left\{\vec{Y}|g(\vec{Y}) = x\right\}$
○ $\mathbf{f}(x)$
● $\mathbf{y}$
$\mathbf{y}$
$\mathbf{f}(x)$
$x=4$
$x=3$
$x=2$
$x=1$
$x=0$
$\mathbf{f}(x) = E_{\vec{Y}|g(\vec{Y})}\left\{\vec{Y}|g(\vec{Y}) \approx x\right\}$

(a)

(b)

Fig. 2. (a) Ideally, with infinite data, every point $\mathbf{f}(x)$ on the HSPC is defined as the average of all data points $\mathbf{y}$ projecting *exactly* onto $\mathbf{f}(x)$. (b) Practically, with limited data, $\mathbf{f}(x)$ is defined by piecewise linear segments. A point $\mathbf{f}(x)$ on the curve is the weighted average of all points $\mathbf{y}$ projecting within a neighborhood of $\mathbf{f}(x)$. For example, $\mathbf{f}(2)$ is the weighted average of all points $\mathbf{y}$ falling within the indicated neighborhood. Note that the number of nodes M in the HSPC equals N, which is the number of data points.

$$g\left(\vec{Y}\right) = \sup_{\lambda\in\Lambda}\left\{\lambda : \left\|\vec{Y} - \mathbf{f}(\lambda)\right\| = \inf_{\mu\in\Lambda}\left\|\vec{Y} - \mathbf{f}(\mu)\right\|\right\}. \tag{4}$$

The latent (coordinate) variable x is usually parameterized by the arc length along $\mathbf{f}(x)$, starting from either end. The *inf* operator finds the point or points on the curve $\mathbf{f}$ that are closest to $\mathbf{y}$; the *sup* operator simply picks the largest coordinate among these points. Note that, although (4) approximates the forward mapping $\mathcal{G}$ as a projection operator, the reverse mapping $\mathcal{F}$ is nonparametric as described by (3), thereby opening up various possibilities for estimating $\mathbf{f}(x)$ as long as the *consistency* property (3) is satisfied. Further, the reverse mapping $\mathcal{F}$ depends on the unknown latent variable x, suggesting that an iterative scheme is needed to compute $\mathbf{f}(x)$. For example, the original principal curve algorithm (denoted HSPC), updates $\mathbf{f}(x)$ by evaluating (3) and (4) in an iterative manner.

In practice, the distribution of $\vec{Y}$ is unknown and the conditional expectation operator is replaced by spline smoothers [19] or locally weighted linear regression [20]. Fig. 2 compares an ideal HSPC with its practical version.

There are several theoretical and practical concerns with the HSPC definition and they are summarized as follows:

1. Existence. HSPCs are not guaranteed to exist for arbitrary distributions, although existence can be shown for ellipsoidal or spherically symmetric densities in $\mathbb{R}^D$, and uniform densities within a square or annuli in $\mathbb{R}^2$ [21]. Note that, in most cases, the principal curves are not unique!
2. Nonparametric. The HSPC is nonparametric, making theoretical analysis complicated and involved, despite its generality.
3. Inefficient. In general,[5] all available (N) data points are needed to faithfully estimate the HSPC, thereby making computations for large N inefficient.
4. Biased. The HSPC is biased at locations of large curvature. Two opposing forces contribute to the overall bias—the model and estimation bias. At these locations, model bias causes the principal curve of data sampled from a function $\mathbf{f}$ with additive isotropic Gaussian noise $\vec{\varepsilon}$,
$$\vec{Y} = \mathbf{f}(x) + \vec{\varepsilon}, \tag{5}$$
to lie at the exterior of the generating curve. The model bias is a direct consequence of the projection operator (4) used in the forward mapping (because more points "project" onto the curve from the outside than from the inside, causing the principal curve to shift outward) and is illustrated in Fig. 3a. On the other hand, the estimation bias results in a "flattening" effect when a high degree of smoothing (large span) is applied, as shown in Fig. 3b. Ideally, it is desired that the model and estimation bias cancel off each other, unfortunately, the estimation bias is predominant in practice.
5. Convergence. Algorithmic convergence of the HSPC to a local minima solution is not assured.

2.3 Alternative Approaches to the Principal Curve

The bias problem (problem 4) was first addressed by Banfield and Raftery's (BR) algorithm [22], which follows the HSPC definition, but estimates the error residuals instead of the actual curve during computation, thereby reducing the estimation bias. However, the BR algorithm introduces numerical instabilities which may lead to a smooth but incorrect principal curve in practice. Chang and Ghosh [23] showed that a more representative principal curve is obtained by first computing a HSPC and then applying the BR algorithm to remove any existing estimation bias.

Tibshirani [24] provided a probabilistic definition of the principal curve (denoted TPC) using a cubic-spline smoothed mixture of Gaussians. Although this parametric formulation does not suffer from the model bias, it is still affected by the estimation bias due to the use of the smoother. Further, the generalized expectation maximization (EM) algorithm [25] ensures that the TPC will always converge to a local minima [26], thereby solving problems 2, 4, and 5. Unfortunately, this definition does not uphold the self-consistency property (3) and, since penalized-likelihood is used as the optimization criterion, the TPC in general will not be optimized in terms of the MSE.[6]

Chang and Ghosh [15] proposed a modified version of TPC known as the probabilistic principal curve (PPC). The

[5] While it is possible to subsample the data in order to obtain a more efficient principal curve representation, the details are beyond the scope of our discussion.

[6] One situation in which the maximum-likelihood solution of the TPC also optimizes the MSE (i.e., yields the minimum MSE solution) is when the TPC is a straight line with nodes corresponding to the projections of the data points onto the line.

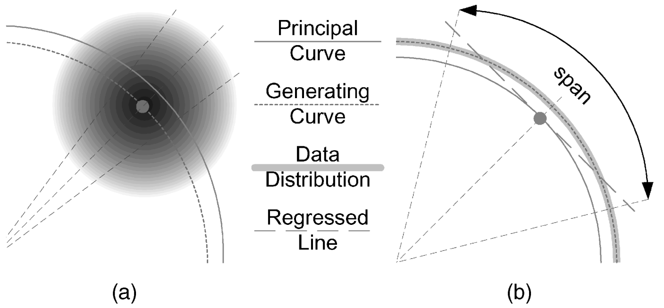


Fig. 3. Both types of bias arise at locations of large curvature. (a) Model bias occurs if the noise is Gaussian distributed about the generating curve. The larger data mass "outside" the curve results in a principal curve lying "outside" of the generating curve. (b) Estimation bias occurs for any local average smoothers with a large span. The large span causes the regression estimate (denoted as a solid dot) to lie at the interior of the generating curve.

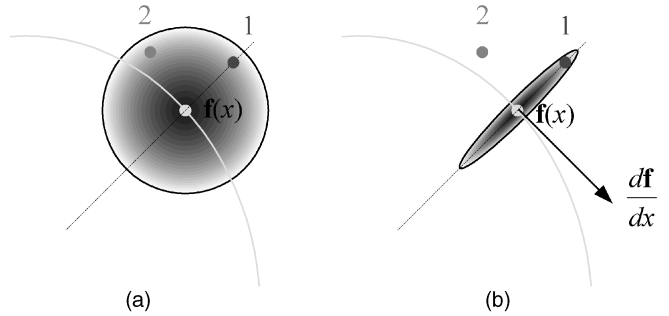


Fig. 4. (a) Under the spherical Gaussian noise model of TPC, points 1 and 2 exert equal influences on the node $\mathbf{f}(x)$, whereas point 1 in (b) is probabilistically closer to $\mathbf{f}(x)$ than point 2 due to the oriented noise covariance of the PPC. The implicit effect is that each node of the PPC will tend to move (during iteration of the EM algorithm) in a way such that the probabilistic (projection) distance is locally minimized.

PPC approximates the self-consistency condition by using oriented Gaussians in the mixture model whose variance along the tangential direction is attenuated by a factor $\alpha < 1$. Experiments show that PPC typically converges twice as fast while attaining much lower MSE compared to TPC. Moreover, the self-consistency condition is achieved in the limit $\alpha \rightarrow 0$. A shortcoming of PPC is that convergence is no longer guaranteed as the modification makes some simplifying assumptions that deviate from the generalized EM framework. Fig. 4 illustrates the PPC advantage with a 2D example.

A solution to problems 1 and 3 was recently proposed by Kégl et al. [27], [28], who define the PC (denoted KPC) as a finite-length curve that minimizes the MSE over all curves of equal or shorter length. Like Tibshirani's formulation, the self-consistency condition is foregone. In place of it is the minimum MSE condition, which ensures that the KPC retains the essence of PCA. The authors show that a KPC always exists for any data distribution with finite second moments, and an asymptotic convergence rate is provided. However, convergence of the corresponding polygonal line algorithm [29], which constructs an approximate but efficient KPC, cannot be guaranteed.

Delicado formulated the PC (denoted DPC) [30], [31] as a curve passing through *principal oriented points*, which are fixed points of some function from $\mathbb{R}^D$ to itself. The DPC always exists for data distribution with finite second moments. A key property of DPC is that it maximizes the "total" variance of data projected along the curve, thereby generalizing the variance-maximization property of the first principal component. The HSPC does not possess this property; for example, only the first principal component of a multivariate Gaussian distribution satisfies the definition of a DPC, whereas any principal component of this distribution qualifies as a HSPC.

A summary of the problems addressed by the various principal curve formulations is given in Table 1. The summary illustrates that, over the years, most problems associated with principal curves have been successfully addressed by redefining the PC. However, *none* of the newer definitions are easily extensible to principal manifolds ($Q > 1$). A simple formulation free of the aforementioned problems is clearly desirable for principal manifolds.

3 Principal Surfaces

In theory, principal surfaces can be defined analogously to principal curves. Now, $\mathbf{x}$ is a coordinate on the $Q \geq 2$ dimensional principal manifold, hence shown in bold. Then, the HSPC can be extended to the HS principal surface (HSPS) as follows:

$$\mathbf{f}(\mathbf{x}) = \mathrm{E}_{\vec{Y}|\mathbf{g}(\vec{Y})}\left\{\vec{Y}|\mathbf{g}\left(\vec{Y}\right) = \mathbf{x}\right\}, \qquad \forall \mathbf{x} \in \boldsymbol{\Lambda} \subseteq \mathbb{R}^{Q}, \tag{6}$$

$$\mathbf{g}\left(\vec{Y}\right) = \sup_{\boldsymbol{\lambda}\in\Lambda}\left\{\boldsymbol{\lambda} : \left\|\vec{Y} - \mathbf{f}(\boldsymbol{\lambda})\right\| = \inf_{\boldsymbol{\mu}\in\Lambda}\left\|\vec{Y} - \mathbf{f}(\boldsymbol{\mu})\right\|\right\}. \tag{7}$$

However, *finding* it is another matter which can become impractical for large Q and, therefore, for the most part, only 2D principal surfaces have been studied, if any. In fact, the HSPC has been extended to the $Q = 2$ case [32], but is nontrivial for $Q > 2$ as it involves Q dimensional estimates of the latent coordinates $\mathbf{x}$ and smoothing operations for the expectation step (6). As a solution to this problem, LeBlanc and Tibshirani [10] proposed adaptive principal surface (APS), which is a general adaptive parametric principal surface approximation for arbitrary Q, using multivariate adaptive regression splines (MARS) [33]. The additive error model used by MARS ensures that APSs are unbiased, and the algorithm is guaranteed to converge. However, it does not satisfy the self-consistency condition and also inherits the disadvantages of MARS, as listed below:

1. It involves complicated procedures for the forward selection and pruning of latent bases.
2. The latent basis functions are not intuitively[7] visualizable in the sense that they do not map out a regular topological grid in latent space.

In the remainder of this section, we critically review alternative principal surface approximation algorithms from the neural network community. Each algorithm is evaluated with respect to the five problems listed in Table 1. We show that, while each of these algorithms has something to offer, they often fall short in other areas. One notable exception is the probabilistic principal surface

[7] An elaborate process known as ANOVA decomposition can be used to interpret a MARS model [33].

TABLE 1
Problems Addressed by Various Principal Curve (PC) Formulations

Definitions	Original		Alternatives			
	HSPC	BR	TPC	PPC	DPC	KPC
self-consistent?	✓	✓		✓[3]		
existence?					✓	✓
parametric?			✓	✓		
efficient?					✓	✓
unbiased?		✓[1]	✓[2]	✓[2]		✓[1]
convergence?			✓			

Key: HSPC (Hastie and Stutzle), BR (Banfield and Raftery), TPC (Tibshirani), PPC (Chang and Ghosh), DPC (Delicado), KPC (Kégl et al.).
[1] *No estimation bias, but model bias still exists.*
[2] *No model bias, but estimation bias still exists.*
[3] *In the limit* $\alpha \to 0$.

(PPS), which appears to hold the best promise as a suitable principal surface approximator.

3.1 Self-Organizing Maps

Kohonen's self-organizing map (SOM) [8] is a nonparametric latent variable model with a topological constraint. During training, the reverse and forward mappings of a SOM[8] are defined, respectively, as

$$\mathbf{f}^{(k+1)}(\mathbf{x}) = \mathrm{E}_{\vec{Y}|\mathbf{g}(\vec{Y})}\left\{\vec{Y}|\mathbf{g}\left(\vec{Y}\right) \in \mathcal{N}(\mathbf{x},k)\right\}, \quad \forall \mathbf{x} \in \{\mathbf{x}_m\}_{m=1}^{M}, \tag{8}$$

$$\mathbf{g}\left(\vec{Y}\right) = \arg \min_{\boldsymbol{\mu} \in \{\mathbf{x}_m\}_{m=1}^{M}} \left\|\vec{Y} - \mathbf{f}^{(k)}(\boldsymbol{\mu})\right\|, \tag{9}$$

where $\mathcal{N}(\mathbf{x},k)$ is the set of nodes lying within a shrinking neighborhood of $\mathbf{x}$ in $\mathrm{I\!R}^Q$ at iteration k, with the neighborhood determined with respect to a chosen latent topology in $\mathrm{I\!R}^Q$. Common topologies include lines ($Q = 1$) and square or hexagonal grids ($Q = 2$). With enough training, the neighborhood of $\mathbf{x}$ will eventually contain just itself, i.e., $\lim_{k\to\infty} \mathcal{N}(\mathbf{x},k) = \mathbf{x}$ and the network is said to have converged. Therefore, at convergence, the topological constraints disappear and the reverse and forward mappings can be expressed, respectively, as follows:

$$\mathbf{f}(\mathbf{x}) = \mathrm{E}_{\vec{Y}|\mathbf{g}(\vec{Y})}\left\{\vec{Y}|\mathbf{g}\left(\vec{Y}\right) = \mathbf{x}\right\}, \quad \forall \mathbf{x} \in \{\mathbf{x}_m\}_{m=1}^{M}, \tag{10}$$

$$\mathbf{g}\left(\vec{Y}\right) = \arg \min_{\boldsymbol{\mu} \in \{\mathbf{x}_m\}_{m=1}^{M}} \left\|\vec{Y} - \mathbf{f}(\boldsymbol{\mu})\right\|, \tag{11}$$

which turns out to be equivalent to the equations for k-means clustering [34]. It can be inferred from (10) and (11) that a **converged** SOM is similar to a discretized version of (6) and (7). For this reason, the SOM can serve as an approximation to the principal surface [17], [18]. The SOM is computationally efficient since it uses only M ($\ll N$) nodes to model the latent manifold. It is also unbiased due to the spherical distance measure (11). However, the main problem with this approximation lies in its reverse mapping (10), which computes the average point-to-node distances (11) instead of the more general point-to-manifold projection distances (7), thereby failing the self-consistency requirement of principal surfaces (6). Analysis of the SOM is involved, partly due to its nonparametric nature, and so far results have shown that the training process does not actually minimize any objective function [35]. Moreover, convergence of the training algorithm has been shown only for the 1D case [36].

3.2 Generative Topographical Mapping

The generative topographical mapping (GTM) [37] is a principled and parametric alternative to the SOM with some nice properties. Like the SOM, it is comprised of M nodes $\{\mathbf{x}_m\}_{m=1}^{M}$ arranged typically on a uniform grid in latent space $\mathrm{I\!R}^Q$. However, unlike the SOM, whose topological constraints gradually disappear with time, the GTM topology is consistently enforced via the reverse-mapping $\mathcal{F}$, which has a generalized linear form

$$\mathbf{f}(\mathbf{x};\mathbf{W}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x}), \qquad \forall \mathbf{x} \in \{\mathbf{x}_m\}_{m=1}^{M},$$

where $\mathbf{W}$ is a $D \times L$ real matrix and

$$\phi(\mathbf{x}) = [\,\phi_1(\mathbf{x}) \quad \cdots \quad \phi_L(\mathbf{x})\,]^T, \qquad \forall \mathbf{x} \in \{\mathbf{x}_m\}_{m=1}^{M},$$

is the vector containing L latent basis functions $\phi_l(\mathbf{x}) : \mathrm{I\!R}^Q \to \mathrm{I\!R}$, $l = 1, \ldots, L$. The basis functions $\phi_l(\mathbf{x})$ are usually chosen to be isotropic Gaussians, the number of which largely determines the mapping complexity of $\mathcal{F}$. In practice, the Lth basis serves as a bias term, i.e., $\phi_L(\mathbf{x}) = 1$ $\forall \mathbf{x}$. Fig. 5 shows an example of a 1D GTM in 3D data space.

The M latent nodes of a GTM are assumed to be uniformly and discretely distributed in latent space with probability density function

$$p_{\vec{X}}(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} \delta(\|\mathbf{x} - \mathbf{x}_m\|), \tag{12}$$

[8] The batch mode SOM algorithm is considered here.

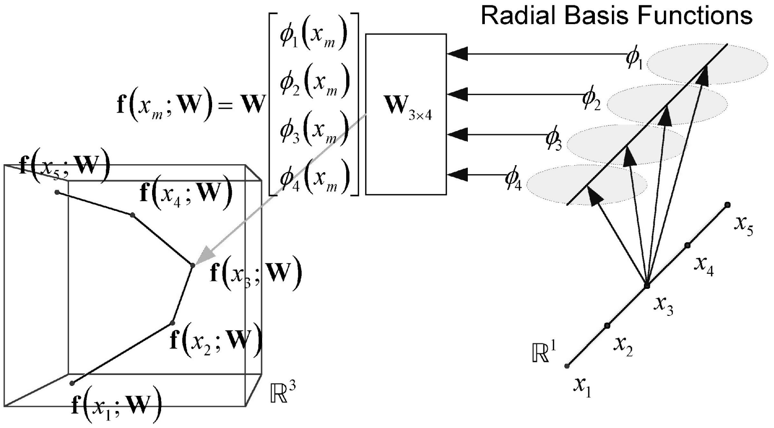

$\mathbf{f}(x_m;\mathbf{W}) = \mathbf{W}\begin{bmatrix}\phi_1(x_m)\\ \phi_2(x_m)\\ \phi_3(x_m)\\ \phi_4(x_m)\end{bmatrix}$
$\mathbf{W}_{3\times4}$
Radial Basis Functions
ϕ_1
ϕ_2
ϕ_3
ϕ_4
$\mathbf{f}(x_5;\mathbf{W})$
$\mathbf{f}(x_4;\mathbf{W})$
$\mathbf{f}(x_3;\mathbf{W})$
$\mathbf{f}(x_2;\mathbf{W})$
$\mathbf{f}(x_1;\mathbf{W})$
$\mathbb{R}^3$
$\mathbb{R}^1$
x_1
x_2
x_3
x_4
x_5

Fig. 5. A GTM example with $D = 3$, $Q = 1$, $L = 4$, and $\mathbf{W}$ a 3×4 matrix. In this example, a radial basis function network with four hidden units maps input latent node x_m to the corresponding output node $\mathbf{f}(x_m; \mathbf{W}) = \mathbf{W}\phi(x_m)$.

where δ is the Dirac delta function. In the data space, the conditional probability distribution of $\mathbf{y}$ given any $\mathbf{x} \in \{\mathbf{x}_m\}_{m=1}^{M}$ is modeled as an isotropic Gaussian with center $\mathbf{f}(\mathbf{x}_m; \mathbf{W})$ and global variance $1/\beta$,

$$p_{\vec{Y}|\vec{X}}(\mathbf{y}|\mathbf{x}_m) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left[-\frac{\beta}{2}\|\mathbf{f}(\mathbf{x}_m; \mathbf{W}) - \mathbf{y}\|^2\right]. \quad (13)$$

Combining (12) and (13) yields a constrained mixture of Gaussian distribution for the output data $\mathbf{y}$,

$$p_{\vec{Y}}(\mathbf{y}) = \frac{1}{M}\sum_{m=1}^{M} p_{\vec{Y}|\vec{X}}(\mathbf{y}|\mathbf{x}_m). \quad (14)$$

From (12), (13), and (14), the conditional probability distribution $p_{\vec{X}|\vec{Y}}(\mathbf{x}|\mathbf{y})$ can be easily computed using Bayes rule. In practice, the mean and/or mode of the conditional distribution $p_{\vec{X}|\vec{Y}}(\mathbf{x}|\mathbf{y})$ is used to find an approximated value for $\mathbf{x}$, i.e., the forward mapping $\mathcal{G}$ is approximated as the mean of $p_{\vec{X}|\vec{Y}}(\mathbf{x}|\mathbf{y})$,

$$\mathbf{g}(\mathbf{y}_n) = \mathrm{E}_{\vec{X}|\vec{Y}}\left\{\vec{X}|\vec{Y} = \mathbf{y}_n\right\}. \quad (15)$$

Equations (12), (13), and (14), are also used in the computation of the mixture-likelihood, which is then maximized with respect to $\mathbf{W}$ and β using the expectation maximization (EM) algorithm [38].

The GTM is efficient and unbiased. In addition, it enjoys the following advantages over the SOM:

1. simple parametric formulation,
2. fewer tunable parameters,
3. guaranteed convergence of the EM algorithm for all Q,
4. consistent mapping functions,
5. smoothness largely determined by the number of latent basis functions L, assuming uniformly distributed isotropic latent bases of constant widths.

The last two properties are especially important within the context of approximating principal surfaces because the GTM can have $M = N$ nodes while retaining smoothness, unlike the SOM. However, the GTM still lacks the self-consistency property of principal surfaces. This major shortcoming motivated our development of the probabilistic principal surfaces, described in the following section.

3.3 Probabilistic Principal Surfaces

In [15], we proposed probabilistic principal surfaces (PPS), which approximate principal surfaces with a modified GTM model. The motivation behind this modification lies in the desire to approximate the self-consistency property of principal surfaces, as shown in Fig. 4. Specifically, the spherical covariance $1/\beta$ in (13) is modified to:

$$\Sigma_{\text{old}} = \frac{\alpha}{\beta}\mathbf{I}_D + \frac{\gamma}{\beta}\sum_{d=Q+1}^{D} \mathbf{e}_d(\mathbf{x})\mathbf{e}_d^T(\mathbf{x}), \quad (16)$$

where $\mathbf{I}_D$ is the $D \times D$ identity matrix and $\{\mathbf{e}_d(\mathbf{x})\}_{d=Q+1}^{D}$ is the set of $D - Q$ unit vectors orthogonal to the manifold spanned by the Q tangential manifold gradient vectors $d\mathbf{f}(\mathbf{x})/dx_1, \ldots, d\mathbf{f}(\mathbf{x})/dx_Q$. Constants α and γ determine the amount of clamping in the tangential manifold direction and amplification in the orthogonal direction, respectively.

The PPS inherits all of GTM's nice properties. In addition, it typically converges faster than the GTM and provides a significantly lower MSE at a similar manifold smoothness level [15]. One disadvantage of PPS is that the modification in (16) results in a nonlinear-likelihood objective function, thus requiring an approximation to be used in the EM algorithm. Unfortunately, because of this approximation, convergence is no longer guaranteed. Nevertheless, this does not appear to be a significant problem as no convergence problems have been observed so far in practice.

3.4 Autoassociative Neural Networks

Autoassociative neural networks (AANNs) were popular in the 1990s as an effective compression tool [7], [39]. Linear AANNs have been shown to extract a linear combination of the principal components [40]. Early researchers proposed 2-layer nonlinear networks comprised of D inputs, a Q-node sigmoidal hidden layer, and a D-node linear output layer. The network output is trained using the back-propagation algorithm to mimic the input vector $\mathbf{y}$. Once trained, the hidden layer nodes will produce a reduced representation ($\mathbf{x}$) corresponding to each input $\mathbf{y}$. Kramer [41] showed that a 2-layer AANN is incapable of modeling the nonlinear relationship among the input and latent

TABLE 2
Problems Addressed By Various Approaches to Principal Surfaces

Algorithm	HSPS	APS	SOM	GTM	PPS
self-consistent?	✓				✓[2]
existence guaranteed?					
parametric?		✓		✓	✓
efficient?			✓	✓	✓
unbiased?		✓	✓	✓	✓
convergence guaranteed?		✓	[1]	✓	✓[3]

Key: HSPS (Hastie and Stuetzle's Principal Surface), APS (Adaptive Principal Surface), SOM (Self-Organizing Map), GTM (Generative Topographic Mapping), and PPS (Probabilistic Principal Surface).

[1] *Only for the case* $Q = 1$.

[2] *In the limit* $\alpha \to 0$.

[3] *Using the generalized EM algorithm described in Appendix C.*

variables, but a 4-layer network overcomes this limitation. Other notable studies include [42], [43], [44].

The AANN is not guaranteed to be self-consistent as it only minimizes the MSE. There have been no studies investigating the bias problem, if any, for AANNs. Malthouse [45] has shown that the AANN is inherently suboptimal in the projections of ambiguity points, defined as data points equidistant to more than one point on a principal manifold. A small change in the region about the ambiguity point may result in a discontinuous jump in the latent variable $\mathbf{x}$ and, since AANNs are unable to model discontinuous jumps, they are forced to interpolate the latent range spanning the discontinuous $\mathbf{x}$ whenever there is a discontinuity. This important observation puts to rest any further attempts at approximating principal manifolds with AANNs.

3.5 Summary

Table 2 summarizes the problems associated with the various approaches to computing principal surfaces. AANNs are not considered here for reasons described previously. Note that the bias criteria here is evaluated with the assumption that the underlying data distribution is generated from a finite number of fixed centers with additive isotropic Gaussian noise. From the table, it can be seen that the PPS exhibits the best prospects as a principal manifold approximator. To the best of our knowledge, existence has not been proven for any of the approaches listed here.

4 PPS with a Unified Covariance Model

4.1 Definition and Interpretation

We propose a unified oriented covariance model for the PPS at each node $\mathbf{x} \in \{\mathbf{x}_m\}_{m=1}^{M}$ that can be expressed as follows:

$$\Sigma(\mathbf{x}) = \frac{\alpha}{\beta}\sum_{q=1}^{Q} \mathbf{e}_q(\mathbf{x})\mathbf{e}_q^T(\mathbf{x}) + \frac{(D-\alpha Q)}{\beta(D-Q)} \sum_{d=Q+1}^{D} \mathbf{e}_d(\mathbf{x})\mathbf{e}_d^T(\mathbf{x}), \qquad 0 < \alpha < D/Q, \tag{17}$$

where

$\{\mathbf{e}_q(\mathbf{x})\}_{q=1}^{Q}$:
set of orthonormal vectors tangential to the manifold at $\mathbf{x}$,

$\{\mathbf{e}_d(\mathbf{x})\}_{d=Q+1}^{D}$:
set of orthonormal vectors orthogonal to the manifold at $\mathbf{x}$.

Note that the complete set of orthonormal vectors $\{\mathbf{e}_d(\mathbf{x})\}_{d=1}^{D}$ spans $\mathbb{R}^D$. The unified PPS model (17) is a more general version of (16) as it reduces PPS to GTM for $\alpha = 1$ and to the manifold-aligned GTM [46] for $\alpha > 1$, i.e.,

$$\Sigma(\mathbf{x}) = \begin{cases} \perp \text{ to manifold} & 0 < \alpha < 1 & \text{PPS} \\ \mathbf{I}_D \text{ or spherical} & \alpha = 1 & \text{GTM} \\ // \text{ to manifold} & 1 < \alpha < D/Q & \text{manifold-aligned GTM.} \end{cases}$$

As $\alpha \to 0$, the support of each node becomes increasingly concentrated on the orthogonal hyperplane at each node, effectively approximating the self-consistency condition of principal surfaces [24]. Note that the total energy or variance (sum of its eigenvalues) of $\Sigma(\mathbf{x})$ remains constant at D/β over the valid range of α, ensuring that the noise level of PPS remains unchanged regardless of its orientation. This property will prove useful later on when we compare the empirical performances of PPS and GTM. Fig. 6 shows the unit Mahalanobis distance loci of $\Sigma(\mathbf{x})$ for various values of α.

4.2 EM Algorithm

The EM algorithm, which is guaranteed to converge to a local minima [25], can be used to estimate the parameters of the PPS. The complete log-likelihood for the PPS, assuming equal and constant prior probabilities, is

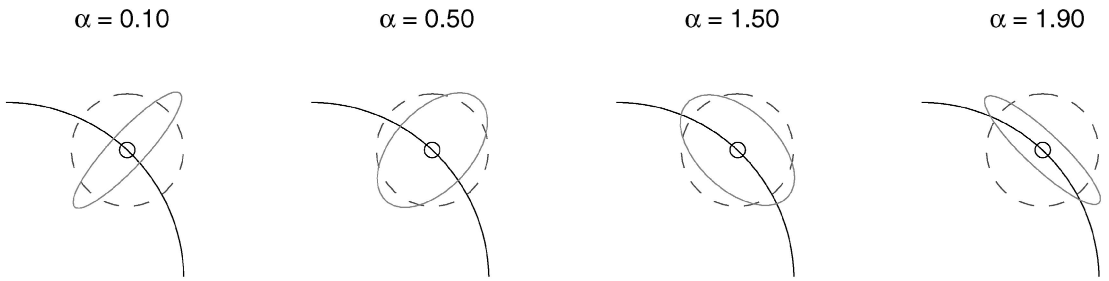


Fig. 6. Unoriented covariances $\alpha = 1$ (dashed line) and oriented covariances (solid line) for $\alpha = 0.10, 0.50, 1.50, 1.90$. The valid range for α is $0 < \alpha < 2$ for $D = 2, Q = 1$ in this example.

$$\mathcal{L}_c = \sum_{n=1}^{N}\sum_{m=1}^{M} z_{mn} \ln\left[p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)\frac{1}{M}\right], \tag{18}$$

where the binary variable z_{mn} indicates whether component m is responsible for generating point $\mathbf{y}_n$, i.e.,

$$z_{mn} = \begin{cases} 1 & \text{if component } m \text{ generated point } \mathbf{y}_n \\ 0 & \text{otherwise.} \end{cases}$$

Since z_{mn} is unknown or "missing," the complete log-likelihood (18) cannot be evaluated. Therefore, in the E-step of the EM algorithm, the expectation of $\mathcal{L}_c$ is computed instead:

$$\langle \mathcal{L}_c \rangle = \sum_{n=1}^{N}\sum_{m=1}^{M} r_{mn}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}, \alpha_{\text{old}}) \ln\left[p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)\frac{1}{M}\right], \tag{19}$$

where the responsibility parameter,

$$\begin{aligned} r_{mn}^{\text{old}} &= P_{\vec{X}|\vec{Y}}(\mathbf{x}_m|\mathbf{y}_n) \\ &= \frac{p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)P_{\vec{X}}(\mathbf{x}_m)}{\sum_{h=1}^{M} p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_h)P_{\vec{X}}(\mathbf{x}_h)} \\ &= \frac{p_{\vec{Y}|\vec{X}}(\mathbf{y_n}|\mathbf{x_m})}{\sum_{h=1}^{M} p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_h)}, \end{aligned} \tag{20}$$

is computed by substituting the "old" parameter values $\mathbf{W}_{\text{old}}, \beta_{\text{old}}, \alpha_{\text{old}}$ into the conditional probabilities $p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)$. In the M-step, the expected log-likelihood function (19) is maximized with respect to $\mathbf{W}$, β, and α, thereby giving the corresponding new iterated values. A regularizing term with an isotropic Gaussian prior on the weights is usually added to (19). Details of the derivations and update equations can be found in Appendix C. For simplicity, in this paper, we assume a constant clamping factor α and approximate the M-step with the original GTM M-step update equations [46], [47]. We have observed no convergence problems over hundreds of trials.

4.3 Computational Issues

The PPS incurs two additional computations over the GTM: 1) computation of the $D \times Q$ tangential matrix $\mathbf{E}_{//}(\mathbf{x})$, which is formed by concatenating the tangential manifold vectors $\{\mathbf{e}_q(\mathbf{x})\}_{q=1}^{Q}$, i.e., $\mathbf{E}_{//}(\mathbf{x}) = \left[\mathbf{e}_1(\mathbf{x}) \ \cdots \ \mathbf{e}_Q(\mathbf{x})\right]_{D\times Q}$, and 2) evaluation of the full Gaussian class-conditional probabilities $p_{\vec{Y}|\vec{X}}(\mathbf{y}|\mathbf{x}_m)$. The set of Q tangential vectors $\{\mathbf{e}_q(\mathbf{x})\}_{q=1}^{Q}$ can be estimated from the partial derivatives of the latent basis activations at $\mathbf{x}$:

$$\mathbf{e}'_q(\mathbf{x}) = \mathbf{W}\frac{\partial\phi(\mathbf{x})}{\partial x_q}, \tag{21}$$

where the constant latent basis derivative $\partial\phi(\mathbf{x})/\partial x_q$ needs to be evaluated only once. However, it is important to note that, since neither the row space of $\mathbf{W}$ nor the set $\left\{\partial\phi(\mathbf{x})/\partial x_q\right\}_{q=1}^{Q}$ is orthogonal in general, the resulting $\{\mathbf{e}'_q(\mathbf{x})\}_{q=1}^{Q}$ will not be orthonormal and, thus, must be made so via the Gram-Schmidt procedure [8] in order to satisfy the conditions of (17).

The matrix $\mathbf{E}_{//}(\mathbf{x})$ is updated once per EM training epoch, which requires $\mathcal{O}(LQD)$ operations for the matrix multiplication and $\mathcal{O}(Q^2D)$ operations for orthonormalization. At first glance, it would seem that the Gram-Schmidt procedure is also needed to compute the corresponding set of orthogonal manifold vectors $\{\mathbf{e}_d(\mathbf{x})\}_{d=Q+1}^{D}$ since (17) involves both the orthogonal and tangential sets of vectors. Fortunately, this is not necessary, as shown in Proposition 1 which simplifies (17) to a form containing just the tangential manifold vectors.

Proposition 1. *Equation (17) can be expressed in terms of just the set of tangential manifold vectors* $\left\{\mathbf{e}_q(\mathbf{x})\right\}_{q=1}^{Q}$ *as follows:*

$$\Sigma(\mathbf{x}) = B\mathbf{I}_D + (S - B)\mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x}), \tag{22}$$

with the constants

$$\begin{aligned} B &= \frac{D - \alpha Q}{\beta(D-Q)}, \\ S &= \frac{\alpha}{\beta}. \end{aligned}$$

Proof. See Appendix A. □

It turns out that (22) also greatly simplifies the corresponding expressions for the determinant and inverse, as shown in Proposition 2.

Proposition 2. *The determinant and inverse of* $\Sigma(\mathbf{x})$ *in (22) can be expressed, respectively, as*

$$|\Sigma| = S^Q B^{D-Q}, \qquad \forall \mathbf{x} \in \{\mathbf{x}_m\}_{m=1}^{M}, \tag{23}$$

$$\Sigma^{-1}(\mathbf{x}) = \frac{1}{B}\mathbf{I}_D - \frac{(S-B)}{BS}\mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x}). \tag{24}$$

The determinant in (23) is constant and, therefore, needs to be evaluated only once. For a given $\mathbf{E}_{//}(\mathbf{x})$, *the computational complexity of the inverse covariance* $\Sigma^{-1}(\mathbf{x})$ *is now* $\mathcal{O}(QD^2)$ *instead of the typical* $\mathcal{O}(D^3)$ *operations required of matrix inversion.*

Proof. See Appendix B. □

With the simplified PPS formulations (23) and (24), evaluation of the conditional probabilities $p_{\vec{Y}|\vec{X}}(\mathbf{y}|\mathbf{x})$ now requires $\mathcal{O}(QD^2)$ operations, which is an order higher (assuming small Q) than the $\mathcal{O}(D)$ complexity of the GTM. However, as shown later in Section 5.5, this computation overhead becomes less of an issue for more complex mappings.

4.4 Performance Evaluation

4.4.1 Roughness

In general, the evaluation of a nonlinear transformation is very subjective. For instance, the MSE cannot be considered alone because any transformation that computes a manifold interpolating all data points will have zero MSE. Therefore, a secondary measurement indicative of the overall smoothness is needed to determine the generality of a manifold. The secondary measurement we used in this paper is the *roughness* of a manifold which, for a 1D manifold, is defined

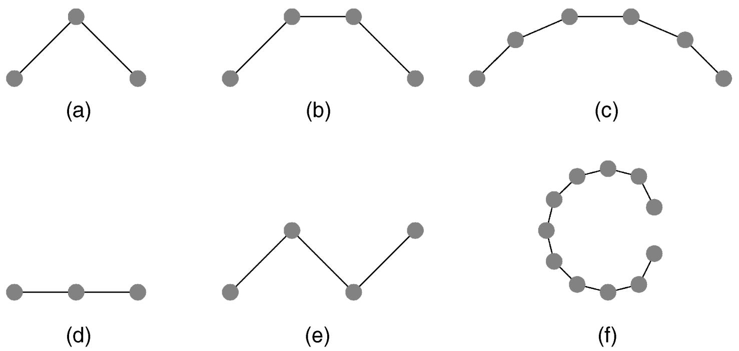


Fig. 7. (a) R = 90 (M = 3), (b) R = 90 (M = 4), and (c) R = 90 (M = 6) are examples of curves with different number of nodes sharing the same roughness value. (d) R = 0 (M =3), (e) R = 180 (M = 4), (f) R = 292.6 (M = 11) are other example curves with their corresponding roughness.

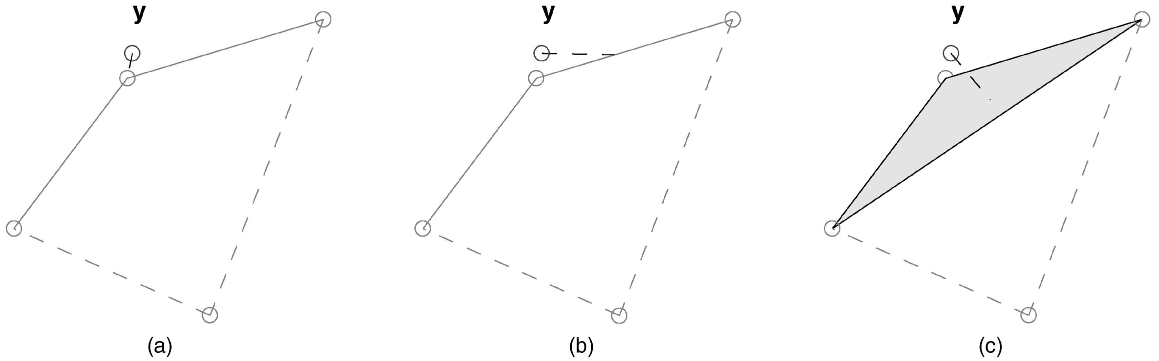


Fig. 8. Various approximations for projecting a point $\mathbf{y}$ onto a 4-node manifold patch in data space. (a) $MSE_{nn} = 0.4900$: projection onto nearest-manifold node. (b) $MSE_{grid} = 0.1700$: projection onto nearest-manifold grid. (c) $MSE_{\Delta} = 0.0800$: projection onto nearest-triangular patch.

as the cumulative angular variation (in degrees) between successive manifold segments:

$$R = \sum_{m=1}^{M-1} \cos^{-1}\left[\frac{\mathbf{f}'(x_m)^T\mathbf{f}'(x_{m+1})}{\|\mathbf{f}'(x_m)\| \;\; \|\mathbf{f}'(x_{m+1})\|}\right], \qquad (25)$$

where $\mathbf{f}'(x_m)/\|\mathbf{f}'(x_m)\|$ denotes the unit gradient directional vector at node x_m. Some sample curves and their corresponding R values are shown in Figs. 7a, 7b, 7c, 7d, 7e, and 7f. In general, R is independent of the number of nodes in the manifold since it simply measures the total angular variation. This is illustrated by the sample curves in Figs. 7a, 7b, and 7c. However, as the number of nodes in the manifold increases, more leeway is allowed for roughness, i.e., R is upper bounded by the number of segments,

$$0^\circ \leq R \leq (M-2)180^\circ, \qquad M \geq 2,$$

with equality holding for straight lines. For high-dimensional ($Q > 1$) manifolds, (25) is averaged over all 1D submanifolds (grid lines).

4.4.2 Projection onto a Manifold

The distance between any data point $\mathbf{y}$ and a 1D manifold can be easily obtained by linearly projecting $\mathbf{y}$ onto all $M-1$ segments on the manifold and taking the minimum of the $M-1$ distances. However, for 2D manifolds consisting of square patches each defined by four manifold nodes, this distance will have to be approximated. The simplest approximation is the nearest-neighbor distance MSE_{nn} as used in the SOM, which finds the minimal squared distance to all possible manifold nodes. A more accurate approximation, used in [15], is the minimum grid projection MSE_{grid}, which finds the shortest projection distance to a manifold grid. The best[9] approximation is the nearest triangulation MSE_{Δ}, which finds the nearest-projection distance to the two possible triangulations, i.e., $MSE_{\Delta} = \min(MSE_{\Delta 1}, MSE_{\Delta 2})$. Fig. 8 shows an example of the three types of approximated MSE used for projecting a point onto a 2D manifold patch. In this paper, we will evaluate all three approximations MSE_{nn}, MSE_{grid}, and MSE_{Δ}.

5 Experiments

In this section, we evaluate PPS, GTM, and the manifold-aligned GTM in terms of reconstruction error (MSE_{Δ}, MSE_{grid}, MSE_{nn}) and roughness R. Under our formulation, an objective comparison between the PPS, GTM, and manifold-aligned GTM can be made as each of them differs from the other only in the parameter α. In addition, the variance or energy of the PPS noise model remains constant over the valid range of α, further ensuring a fair comparison. Convergence properties are also investigated.

5.1 Data Set Description and Experiment Setup

Three popular UCI machine learning data sets [48], iris, glass, and diabetes, with characteristics described in Table 3

[9] Strictly speaking, the "best" distance (maximum-likelihood) under the generative framework is actually measured as that between the data point $\mathbf{y}$ and $\langle \mathbf{x}|\mathbf{y} \rangle$, the mean of its induced distribution $p_{\vec{X}|\vec{Y}}(\mathbf{x}|\mathbf{y})$ on the manifold. Moreover, as this distance may not be the shortest in the Euclidean sense, we instead compute the distance using linear projection onto the manifold made up of the nodes. In this paper, the generative model is simply a means of obtaining the node locations in data space.

TABLE 3
Characteristics of the Three UCI Machine Learning Data Sets

Dataset	D	N_{train}	N_{test}
iris	4	75	75
glass	9	107	107
diabetes	8	384	384

were considered. The goal is to study the reconstruction MSE when the input data is represented by 1D and 2D PPSs. To facilitate fair comparison across different data sets, each data set was first normalized to zero mean and unit *covariance* by sphering (whitening) [11]. The MSE and roughness R of each PPS is evaluated while varying the manifold size (M), manifold complexity (L), and clamping parameter α. The orientation parameter was varied over the set of 13 values as shown below:

$$\alpha \in \left\{0.1,\, 0.2,\, \ldots,\, 1.0,\, 0.75 + 0.25\frac{D}{Q}, 0.5 + 0.5\frac{D}{Q}, 0.25 + 0.75\frac{D}{Q}\right\}.$$

The manifold size M was varied from $0.1N$ to N in $0.1N$ increments, rounded to the nearest integer, where N refers to the number of training samples. A range of $L \in \{4, 9, 16\}$ latent basis functions was used.

At each (M, L, α) setting, a total of 25 simulation runs was repeated on randomly permutated 50/50 train/test partitions of the data, and averaged to yield the MSE estimate. Each run was allowed a maximum of 200 epochs with early stopping triggered whenever the change in training MSE ($Q = 1$) or MSE_{Δ} ($Q = 2$) went below $TOL = 0.1$ percent across consecutive 5-epoch windows. Overtraining is not an issue as long as the number of latent basis functions (L) is kept low. So, unlike generalization situations, some form of crossvalidation is not needed here. At the end of each 5-epoch training window, the test MSE was evaluated. For 1D manifolds, both the MSE_{nn} and MSE (projection) were evaluated, whereas all three MSE approximations MSE_{nn}, MSE_{grid}, and MSE_{Δ} were evaluated for 2D manifolds.

The following common settings were used for all experiments: regularization parameter $\lambda = 0.01$, isotropic Gaussian latent basis functions uniformly distributed within the range $[-1, 1]$ of each latent dimension, with widths of the basis functions set to twice the distance between two adjacent centers, manifolds initialized to the first principal axis ($Q = 1$) or plane ($Q = 2$). The publicly available GTM MATLAB toolbox [47] was modified to accommodate PPS. Note that the current GTM implementation restricts 2D manifolds to span a square patch in latent space, i.e., both M and L must be squares of integers.

5.2 1D PPS

A 1D PPS of relatively low mapping complexity ($L = 4$) was first computed for each of the three data sets over a range of M, α values. Figs. 9a, 9b, 9c, 9d, 9e, and 9f show plots of the roughness measure R and test MSE, respectively, versus M and α for a 1D PPS computed on the three data sets. Global minimum values are marked by a circle ($\bigcirc$) on the plots, with numerical values indicated above each plot. Interestingly, R actually decreases with increasing number of nodes and varies tremendously for $\alpha \gg 1$ across all three data sets. By specifically capturing noise in the tangential direction, the manifold-aligned GTM ($\alpha > 1$) enjoys a considerable advantage over the other two models; it yielded the smoothest manifold (lowest R), most noticeable in Fig. 9a, over all three data sets.

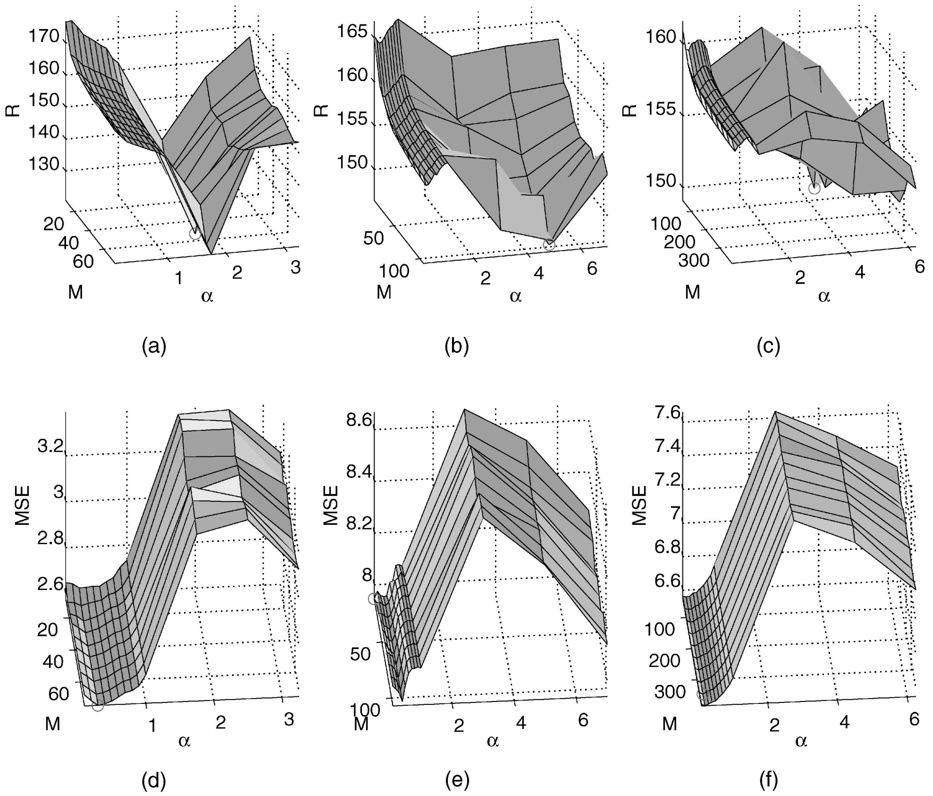
Due to the remarkable similarity between the MSE and MSE_{nn} results, which only differ slightly in magnitude, we shall comment only on the MSE performances and draw the same conclusions regarding the corresponding MSE_{nn} performances. As expected, the MSE did not vary much across the range of M for both the PPS and GTM ($\alpha \leq 1$), except for very small M. However, the manifold-aligned GTM ($\alpha > 1$) exhibited relatively higher variation across M, as seen in Figs. 9d, 9e, and 9f, indicating it to be a relatively unstable model. For all three data sets, MSE increases with α until a point where it starts to decrease slightly. This decrease always occurs way beyond $\alpha = 1$ (GTM), but the decrease is not substantial enough to undertake the low level of the PPS ($\alpha < 1$). *More importantly, the PPS was able to consistently attain the lowest MSE at any given M and L, exhibiting its superiority.* For all three data sets, the lowest MSE was achieved by the PPS with a relatively tight clamping factor ($\alpha = 0.1 \sim 0.3$) and a large number of nodes M.

The glass data set displayed a deviation from the other data sets in that its minimum MSE corresponds to the smallest value of $M = 11$ and $\alpha = 0.10$ (PPS), as indicated ($\bigcirc$) in Fig. 9e. Similarly, the lowest MSE for the GTM ($\alpha = 1$) is at $M = 11$ (not shown). In either case, there is a hefty price to pay, in the form of a large roughness R, as shown in the corresponding plot of the roughness in Fig. 9b. On the other hand, the lowest MSE_{nn} occurs at $M = 107$ and $\alpha = 0.40$, with a reasonable $R = 155.82$ (not shown). This example clearly demonstrates that the roughness R, in addition to MSE, must be evaluated when tuning the parameters of a PPS. In general, a compromise must first be made between the number of nodes M and the maximum tolerable roughness R, after which a suitable α can be determined experimentally.

In order to assess the relative MSE performances of the GTM and PPS at a given roughness level, five levels of roughness R were considered, ranging from 10 to 50 percent of the full range of roughness R of a GTM. At each roughness level, the GTM reconstruction MSE was noted, and compared to the *best* PPS at the same linearly interpolated roughness level. Fig. 10 shows the percentage decrease in MSE of the best PPS at various roughness level for each of the three data sets. From the figure, it can be seen that the best PPS always achieve an improvement (up to 5 percent) over the GTM at the same roughness level R.

5.3 2D PPS

In this section, the performances of 2D PPS is evaluated against that of the 2D GTM and manifold-aligned GTM. A 2D PPS is computed for each of the three data sets while varying $M, \alpha,$ and L. Figs. 11a, 11b, 11c, 11d, 11e, and 11f show plots of the roughness R versus M, α for each of the three data



(a) (b) (c)

(d) (e) (f)

Fig. 9. (1D PPS) Roughness (R) versus number of nodes (M) and clamping factor (α) for (a) iris1 (L = 4) Min R = 120.8212 @ (M, α) = (53, 1.75), (b) glass1 (L = 4) Min R = 146.4620 @ (M, α) = (97, 5.00), and (c) diab1 (L = 4) Min R = 149.1144 @ (M, α) = (39, 4.50). MSE versus M, α for (d) iris1 (L = 4) Min MSE = 2.5786 with R = 158.5607 @ (M, α) = (75, 0.30), (e) glass1 (L = 4) Min MSE = 7.9465 with R = 165.2362 @ (M, α) = (11, 0.10), and (f) diab1 (L = 4) Min MSE = 6.5509 with R = 156.8888 @ (M, α) = (346, 0.20).

sets at two complexity levels. The global minimum is indicated by a circle (◯). From the figures, it can be observed that the roughness surface R is rather smooth for all cases, except for the smaller iris data set at a lower complexity $(L = 4)$ in Fig. 11a. The manifold-aligned GTM $(\alpha > 1)$ is again the overall smoothness champion in all cases. Ironically, the roughness R exhibited less variation with respect to α for higher mapping complexities. In the case of the iris data, the high roughness of a lower complexity mapping may actually indicate that the data is locally clustered about each class, thereby requiring a more complex mapping. In fact, the less complex $(L = 4)$ PPS $(\alpha < 1)$ experienced a sharp increase in roughness R with decreasing α, as shown in Fig. 11a. Likewise, the same phenomena was also observed for the glass and diabetes data sets (not plotted here), which seems to defy the notion that a smaller L (less complex) should yield a smoother manifold.

One possible explanation is given as follows: As $\alpha \to 0$, the output data space nodes effectively become decoupled, i.e., each output node estimates the noise immediately within its projection vicinity, quite independent of its neighboring node. On the contrary, the nodes are closely coupled in latent space by virtue of having too few number of bases (small L). It is this nonlinear disparity of scales between the input and output spaces that leads to a poor mapping $\mathbf{W}$, invariably resulting in a rougher manifold. A PPS with sufficient number of latent bases (larger L), where each can become responsible for a fewer number of latent nodes, allows the mapping $\mathbf{W}$ to better handle the

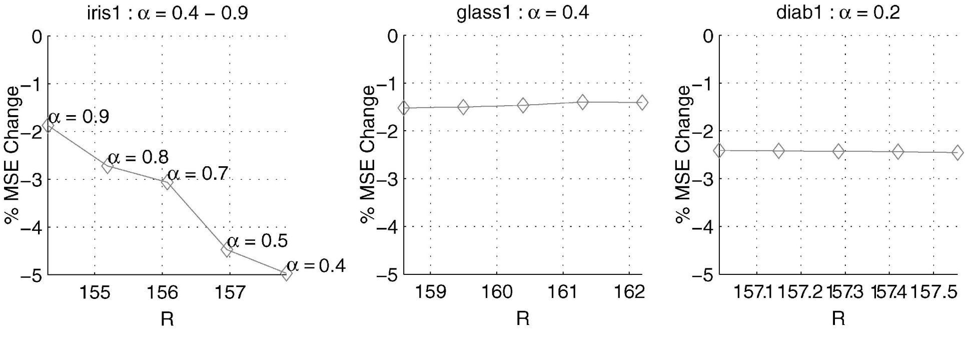


Fig. 10. (1D PPS) Percentage change in MSE of the best PPS over the GTM at each roughness level R for the iris, glass, and diabetes data set.

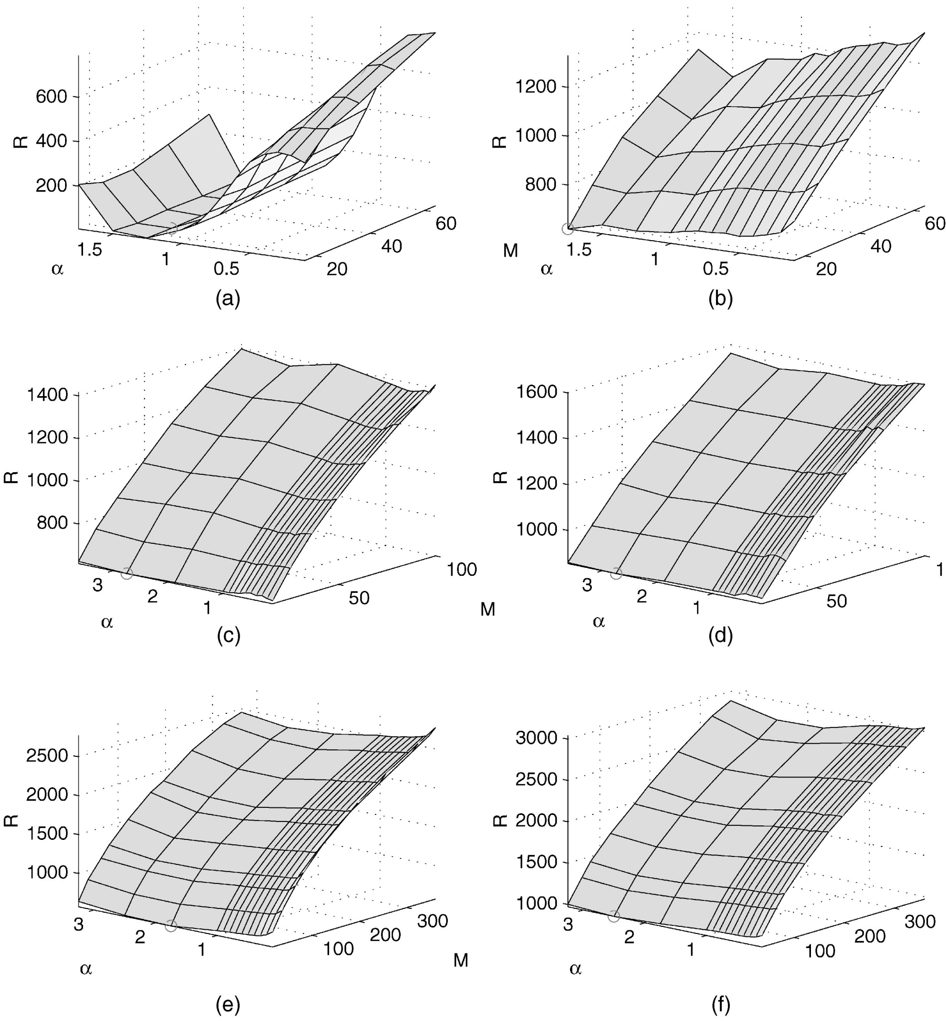


Fig. 11. (2D PPS) Roughness (R) versus number of nodes (M) and clamping factor (α) for (a) iris2 $(L = 4)$ Min R = 7.4033 @ (M, α) = (25, 1.25), (b) iris2 $(L = 9)$ Min R = 619.7569 @ (M, α) = (16, 1.75), (c) glass2 $(L = 9)$ Min R = 614.2889 @ (M,α) = (16, 2.75), (d) glass2 $(L = 16)$ Min R = 855.4906 @ (M,α) = (25, 2.75), (e) diab2 $(L = 9)$ Min R = 570.7659 @ (M,α) = (16, 1.75), and (f) diab2 $(L = 16)$ Min R = 971.1465 @ (M,α) = (36, 2.50).

decoupling of the output nodes at small α, as shown in Figs. 11b, 11c, 11d, 11e, and 11f.

Figs. 12a, 12b, 12c, 12d, 12e, and 12f show plots of the test MSE_Δ versus M, α for the three data sets. As with the 1D PPS case, the plots for the other two reconstruction error estimates MSE_{grid} and MSE_{nn} closely resemble those of the MSE_Δ and we shall only comment on the MSE_Δ. As shown, the MSE_Δ error surface of the manifold-aligned GTM (for $\alpha > 1$) becomes more varied with respect to M with increasing mapping complexity (larger L), again confirming that the manifold-aligned GTM should not be used for large L. Another effect of increasing complexity is that the MSE_Δ surface becomes convex with respect to α, with the optimal values of α in the range 0.2-0.4, as shown in Figs. 12b, 12d, 12e, and 12f.

The percentage improvement in MSE_Δ of the best PPS over the GTM is plotted in Fig. 13 for five levels of roughness levels $Rfac$, which denotes 10-50 percent of the range of R of a GTM. In general, the absolute percentage reduction decreases only slightly at higher roughness levels, with the general trend across L closely preserved. A significant reduction of 7-10 percent in MSE_Δ can be expected from low complexity $(L = 4)$ PPSs at the same roughness level as a GTM. However, caution should be exercised in using a low complexity PPS, which can be significantly rougher than the GTM with the same number of nodes. The amount of MSE_Δ reduction did not vary much for PPSs of sufficient complexity $(L = 9, 16)$, though $L = 16$ appears to be the "sweet spot" for the glass data set and $L = 4$ yields the best improvement for the diabetes data set.

5.4 Best Results

While the previously plotted error surfaces show that a PPS will always achieve a lower MSE than a GTM at any given M and L, it would be interesting to see how well the best PPS measures up to the best GTM over all possible configurations of M. In other words, we want to find out if it is possible to have the best GTM (with M_1 nodes) perform better than the best PPS (with $M_2 \neq M_1$ nodes). Table 4 summarizes, for each data set, the MSE_Δ (equals MSE for 1D PPS) and roughness R of the best PPS and best GTM at a given manifold dimensionality Q and complexity L. The manifold-aligned GTM was not included since it performed worse than the GTM with respect to reconstruction error.

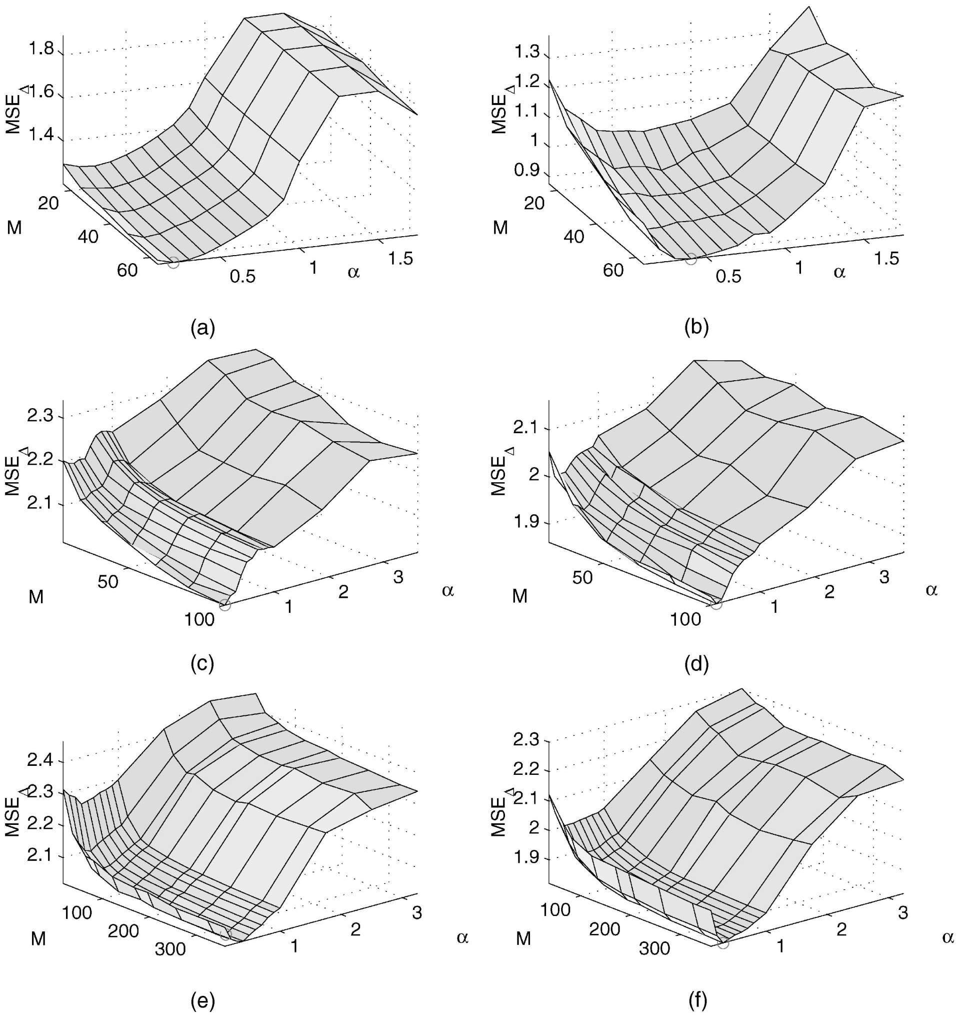


Fig. 12. (2D PPS) Test MSE_Δ versus number of nodes (M) and clamping factor (α) for (a) iris2 $(L = 4)$ Min $\mathrm{MSE}_\Delta = 1.2013$ with R = 770.5084 @ (M, α) = (64, 0.20), (b) iris2 $(L = 9)$ Min $\mathrm{MSE}_\Delta = .08757$ with R = 1265.6403 @ (M, α) = (64, 0.40), (c) glass2 $(L = 9)$ Min $\mathrm{MSE}_\Delta = 2.0156$ with R = 1416.5809 @ (M, α) = (100, 0.10), (d) glass2 $(L = 16)$ Min $\mathrm{MSE}_\Delta = 1.8617$ with R = 1601.1549 @ (M, α) = (100, 0.20), (e) diab2 $(L = 9)$ Min $\mathrm{MSE}_\Delta = 2.0187$ with R = 2476.3647 @ (M, α) = (324, 0.40), and (f) diab2 $(L = 16)$ Min $\mathrm{MSE}_\Delta = 1.8202$ with R = 2999.4415 @ (M, α) = (361, 0.30).

It can be seen that, in every case, the best PPS achieved a lower MSE_Δ than the best GTM. The 1D PPS showed modest reduction in MSE_Δ ranging from 1.5 percent to 4.6 percent, with roughness level maintaining at -1.0 percent to 4.4 percent. At a higher latent mapping complexity $(L = 9, 16)$, the best 2D PPS betters the GTM for the glass and diabetes data sets in terms of reconstruction error by 3.3 percent to 8.8 percent while retaining a lower or comparable roughness level (-49.8 percent to 2.7 percent). Notably different is the 2D PPS $(L = 9)$ result on the iris data set, which shows a larger 40.2 percent increase in R than the corresponding 8.8 percent decrease in MSE. This is due to the significantly larger number of nodes $(M = 64)$ used by the 2D PPS compared to that $(M = 36)$ used by the 2D GTM. For comparison, the best PPS using $M = 36$ nodes yielded a $MSE_\Delta = 0.9221$ and $R = 966.08$, which translates to a more reasonable change of -4.0 percent and 7.0 percent, respectively. From these results, we see that, in general, it is important to use a sufficient number of latent basis functions in order to realize the benefits of the PPS.

5.5 Convergence

Figs. 14a, 14b, 14c, 14d, 14e, 14f, 14g, 14h, and 14i show plots of the averaged (over 25 trials) 2D PPS training MSE_Δ versus epoch for all three data sets. A plot was obtained for three selected values of M, corresponding to low, medium, and high node densities, respectively. The marked location on each curve indicates "convergence," where the average MSE_Δ has reduced to within 1×10^{-5} of its final value. In general, the manifold-aligned GTM $(\alpha > 1)$ tends to vary erratically during training and failed to converge within 200 epochs in most of the cases. The situation gets better as α approaches unity from above, e.g., the curves corresponding to $\alpha = 1.3$ in Figs. 14a, 14b, and 14c exhibited good convergence behavior. Similarly, at the other end of the valid range of α, poor convergence behavior was observed for $\alpha = 0.1$ or less. In all other cases, the PPS required approximately the same number of epochs as the GTM for convergence.

Note that each PPS training epoch involves additional computations over the GTM. Fig. 15 plots the slow-down factor per epoch of the PPS with respect to the GTM for

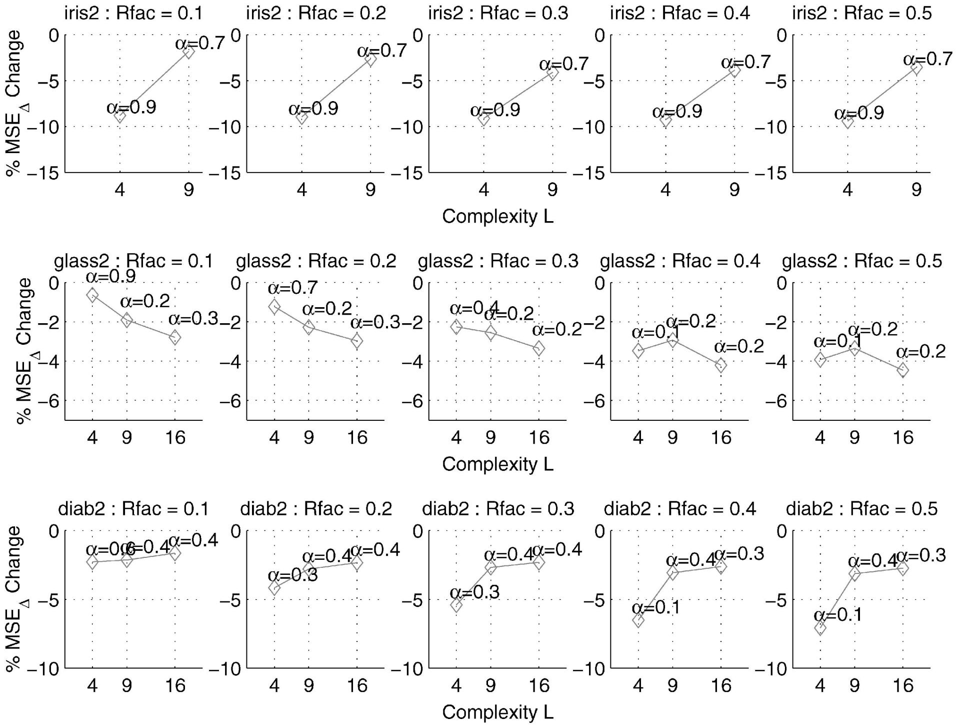


Fig. 13. (2D PPS) Percentage change in MSE_Δ of the best (corresponding α shown) PPS versus complexity L at each roughness factor level $Rfac$ for the iris, glass, and diabetes data set.

various values of M, L. The number of floating-point operations was used as a yardstick, taking into account the different complexities of addition and multiplication operations. For example, a value of 1.2 on the vertical axis would indicate that the PPS required 20 percent more floating point operations than the GTM for one epoch. As expected, the ratio remains fairly constant over M, except for very small values thereof. For small L, the PPS overhead is significant, requiring as much as three times as many operations as the GTM, as shown in Fig. 15d. However, the PPS overhead diminishes with respect to the core computations for increasingly complex (larger L) mappings, as shown by the relatively low 30-40 percent overhead incurred by the $L = 16$ plots.

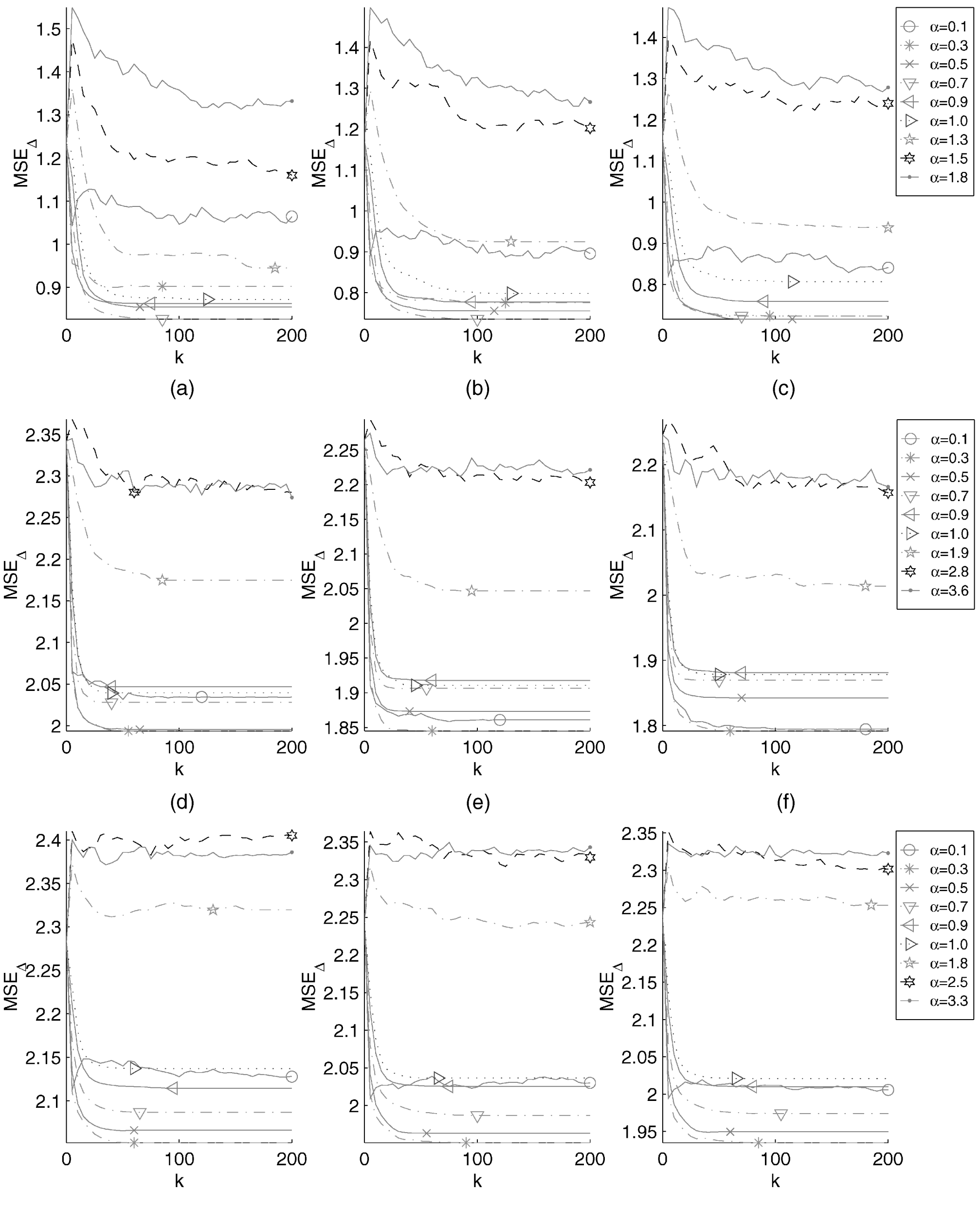
6 Discussion

From the performances of PPS on the three benchmark data sets, the following points can be made:

1. The PPS attains a lower MSE_Δ over the other two models for all M, L.

TABLE 4
Comparing the Best GTM Results Against the Best PPS Results

Dataset	Q	L	GTM ($\alpha = 1$)			PPS				% Change	
			M	MSE_Δ	R	M	α	MSE_Δ	R	MSE_Δ	R
iris	1	4	75	2.7020	151.95	75	0.3	2.5786	158.56	-4.6	4.4
	2	4	49	1.6046	142.98	64	0.2	1.2013	770.51	-25.1	438.9
		9	36	0.9601	902.65	64	0.4	0.8757	1265.64	-8.8	40.2
glass	1	4	11	8.0681	166.93	11	0.1	7.9465	165.24	-1.5	-1.0
	2	4	100	2.3520	711.04	100	0.1	2.1861	1008.24	-7.1	41.8
		9	100	2.1178	1379.96	100	0.1	2.0156	1416.58	-4.8	2.7
		16	100	1.9634	1562.07	49	0.2	1.8617	784.56	-5.2	-49.8
diabetes	1	4	384	6.7100	156.53	346	0.2	6.5509	156.89	-2.4	0.2
	2	4	361	2.3882	1461.69	361	0.1	2.1825	2231.95	-8.6	52.7
		9	361	2.0918	2528.16	324	0.4	2.0187	2476.36	-3.5	-2.0
		16	361	1.8822	2934.03	361	0.3	1.8202	2999.44	-3.3	2.2



(a) (b) (c)

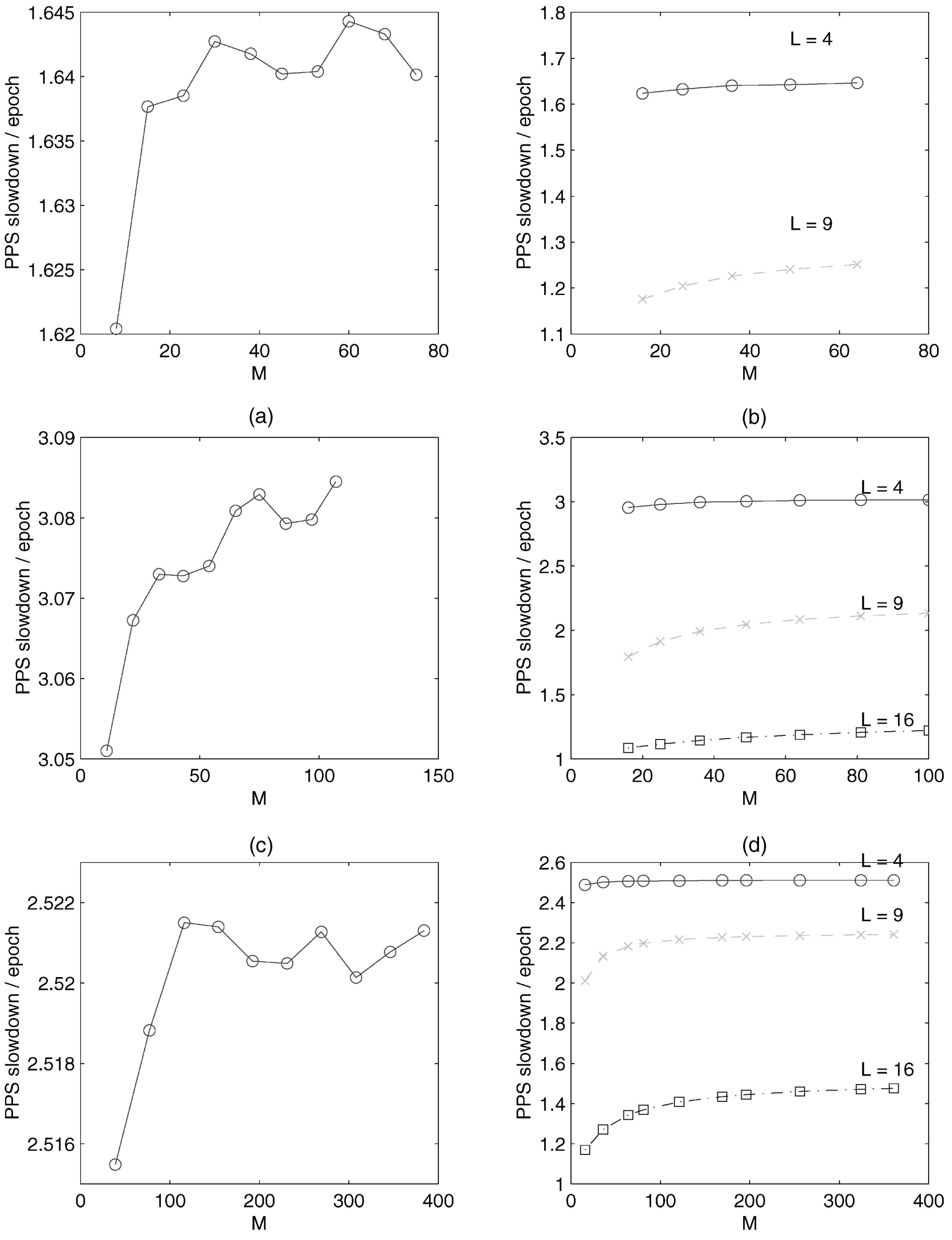
(d) (e) (f)

(g) (h) (i)

Fig. 14. (2D PPS). Averaged training MSE_Δ versus number of epochs k for (a) iris2: L = 9, M = 16, (b) iris2: L = 9, M = 36, (c) iris2: L = 9, M = 64, (d) glass2: L = 9, M =16, (e) glass2: L = 9, M =49, (f) glass2: L = 9, M =100, (g) diab2: L = 9, M = 36, (h) diab2: L = 9, M = 169, and (i) diab2: L = 9, M = 361. Marked locations on curve indicate the epoch at which all 25 runs of the experiment have converged, where applicable.

2. At the same roughness level R, the PPS gives a lower reconstruction error compared to the GTM and manifold-aligned GTM. The improvement gets better with increasing M.
3. The best PPS performs better than the best GTM, indicating that the superiority of the PPS is not local.
4. The optimal clamping factor α is very data-dependent and must be evaluated across trials. In general, extreme values are not recommended, whereas a value less than 0.5 should yield a PPS with satisfactory MSE and roughness R. Alternatively, the more involved generalized EM algorithm may be used to derive a flexible value for α.
5. The PPS not only reduces the MSE_Δ, but also the MSE_{grid} and MSE_{nn}, making it beneficial even in applications that only consider the data-to-node distances. Consequently, the MSE_{nn} and MSE_{grid} may be used to approximate MSE_Δ when searching for the best α parameter.
6. The computational overhead of the PPS reduces to only 30-40 percent over the GTM for larger L. Further, notable improvements in reconstruction error were also observed for larger L. The combination of these two observations suggests that the best result can be expected when a relatively complex PPS is employed.

In all experiments, it was noted that the unified model performed poorly at either extreme of the valid range of α values. As $\alpha \to 0$, the PPS models only the noise in the orthogonal manifold direction, with each node effectively



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 15. Per epoch slow-down factor of the PPS over the GTM for various values of M and L. Note that, as the mapping gets increasingly complex (larger L), the extra PPS computational overhead diminishes to just 40 percent or less. (a) Iris 1D PPS (L = 4), (b) iris 2D PPS, (c) glass 1D PPS (L = 4), (d) glass 2D PPS, (e) diab 1D PPS (L = 4), and (f) diab 2D PPS.

decoupled from its neighbor. On the contrary, at values of $\alpha \to D/Q$, which correspond to the manifold-aligned GTM model, only noise along the manifold is modeled. Thus, the value of α determines the type of noise model employed by the PPS. The idea of decomposing the noise into tangential and orthogonal components is not new. Banfield and Raftery [22] used a linear combination of variances along and about the manifold (in addition to a residual variance term) to cluster ice-floe outlines. However, as mentioned before, their results were limited to principal curves and cannot be extended to principal surfaces ($Q = 2$) and manifolds ($Q > 2$). The contribution of our work lies in its extensibility to principal manifolds of arbitrary dimensions.

7 Conclusion

We reviewed various formulations for principal curves and surfaces and critiqued each with respect to problems related to self-consistency, existence, parametricity, efficiency, bias, and convergence. We then proposed a unified PPS model that overcomes all of these problems, except for existence, which has not been proven for the general case of principal manifolds ($Q > 1$). With the unified PPS model, we were able to compare the 1D and 2D versions of the PPS to the corresponding GTM and manifold-aligned GTM in a fair manner over a range of parameters. The PPS was found to outperform the GTM and manifold-aligned GTM by a significant margin under two different comparisons using the reconstruction error criterion. The empirical convergence characteristic of the PPS was also studied and found to be comparable to the GTM in terms of the number of required training epochs. It was also shown that the computational overhead incurred by the PPS becomes less of an issue with increasing mapping complexity.

The PPS appears to be a promising approximation algorithm for principal manifolds. In addition, the proposed generalized EM algorithm (Appendix C) for PPS includes update equations for automatically determining the clamping factor α, making the PPS even more attractive. Further, it is possible to assign a different α to each manifold node in our model so as to give localized noise estimates that best fit the region, be it orthogonal or tangential. Although only empirical results on 1D and 2D manifolds were shown, the model can be easily extended to 3D or higher-dimensional manifolds. In fact, to date, a 3D spherical PPS has been applied to visualization of high-dimensional data and also for the pose-estimation and classification of aircraft and vehicle images [49], [50].

Appendix A
Proof of Proposition 1

Proof. Observe that, since $\{\mathbf{e}_d(\mathbf{x})\}_{d=1}^{D}$ is orthonormal and complete in $\mathbb{R}^D$, the following holds:

$$\mathbf{I}_D = \mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x}) + \mathbf{E}_{\perp}(\mathbf{x})\mathbf{E}_{\perp}^T(\mathbf{x}), \tag{26}$$

where $\mathbf{I}_D \in \mathbb{R}^{D\times D}$ is the identity matrix and $\mathbf{E}_{\perp}(\mathbf{x}) \in \mathbb{R}^{D\times(D-Q)}$ is formed by concatenating the orthogonal manifold vectors $\{\mathbf{e}_d(\mathbf{x})\}_{d=Q+1}^{D}$,

$$\mathbf{E}_{\perp}(\mathbf{x}) = [\,\mathbf{e}_{Q+1}(\mathbf{x}) \quad \cdots \quad \mathbf{e}_D(\mathbf{x})\,]_{D\times(D-Q)}.$$

Expressing (17) in matrix notation and substituting into it (26), we obtain the desired expression,

$$\begin{aligned}\Sigma(\mathbf{x}) &= S\mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x}) + B\mathbf{E}_{\perp}(\mathbf{x})\mathbf{E}_{\perp}^T(\mathbf{x})\\ &= S\mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^{\mathbf{T}}(\mathbf{x}) + \mathbf{B}\Big(\mathbf{I_D} - \mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^{\mathbf{T}}(\mathbf{x})\Big)\\ &= B\mathbf{I}_D + (S-B)\mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x}).\end{aligned}$$

□

Appendix B
Proof of Proposition 2

Before proving Proposition 2, it is necessary to first introduce and prove the following lemma.

Lemma 3. *Let* $\mathbf{A} \in \mathbb{R}^{D\times Q}$ *with* $Q < D$ *orthonormal columns (i.e.,* $\operatorname{rank}(\mathbf{A}) = Q$ *and* $\mathbf{A}^T\mathbf{A} = \mathbf{I}_Q$*), then the following holds for any positive real number* k*:*

1. $\left|\mathbf{A}^T\mathbf{A} + k\mathbf{I}_Q\right| = (k+1)^Q$.
2. $\left|\mathbf{A}\mathbf{A}^T + k\mathbf{I}_D\right| = k^{D-Q}(k+1)^Q$.
3. $(\mathbf{A}\mathbf{A}^T + k\mathbf{I}_D)^{-1} = \frac{1}{k}\mathbf{I}_D - \frac{1}{k(k+1)}\mathbf{A}\mathbf{A}^T$.

Proof.

1. Observe that, since $\mathbf{A}^T\mathbf{A} = \mathbf{I}_Q$,
$$\left|\mathbf{A}^T\mathbf{A}+k\mathbf{I}_Q\right| = \left|(k+1)\mathbf{I}_Q\right| = (k+1)^Q.$$
2. By taking the determinants on both sides of the following equality [51],
$$\begin{aligned}&\begin{bmatrix}\mathbf{A}\mathbf{A}^T + k\mathbf{I}_D & -\mathbf{A}\\ 0 & k\mathbf{I}_Q\end{bmatrix}\begin{bmatrix}\mathbf{I}_D & 0\\ \mathbf{A}^T & \mathbf{I}_Q\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{I}_D & 0\\ \mathbf{A}^T & \mathbf{I}_Q\end{bmatrix}\begin{bmatrix}k\mathbf{I}_D & -\mathbf{A}\\ 0 & \mathbf{A}^T\mathbf{A} + k\mathbf{I}_Q\end{bmatrix},\end{aligned}$$
we obtain the desired result
$$\begin{aligned}k^Q\left|\mathbf{A}\mathbf{A}^T + k\mathbf{I}_D\right| &= k^D\left|\mathbf{A}^T\mathbf{A} + k\mathbf{I}_Q\right|\\ \left|\mathbf{A}\mathbf{A}^T + k\mathbf{I}_D\right| &= k^{D-Q}(k+1)^Q.\end{aligned}$$
3. The proof follows directly from premultiplying both sides of the expression by $(\mathbf{A}\mathbf{A}^T + k\mathbf{I}_D)$. □

We now proceed to prove Proposition 2.

Proof. Applying the second result of Lemma 3 to the determinant of (22), we obtain

$$\begin{aligned}|\Sigma(\mathbf{x})| &= \left|B\mathbf{I}_D + (S-B)\mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x})\right|\\ &= (S-B)^D\left|\frac{B}{S-B}\mathbf{I}_D + \mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x})\right|\\ &= (S-B)^D\left(\frac{B}{S-B}\right)^{D-Q}\left(\frac{B}{S-B}+1\right)^Q\\ &= S^QB^{D-Q}.\end{aligned}$$

Note that the determinant remains constant for all $\mathbf{x} \in \{\mathbf{x}_m\}_{m=1}^{M}$. Similarly, applying the third result of Lemma 3, the inverse is

$$\begin{aligned}\Sigma^{-1}(\mathbf{x}) &= \Big(B\mathbf{I}_D + (S-B)\mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x})\Big)^{-1}\\ &= \frac{1}{S-B}\left(\frac{B}{S-B}\mathbf{I}_D + \mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x})\right)^{-1}\\ &= \frac{1}{S-B}\left[\frac{S-B}{B}\mathbf{I}_D - \frac{(S-B)^2}{BS}\mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x})\right]\\ &= \frac{1}{B}\mathbf{I}_D - \frac{(S-B)}{BS}\mathbf{E}_{//}(\mathbf{x})\mathbf{E}_{//}^T(\mathbf{x}).\end{aligned}$$

□

Appendix C
Derivation of the Generalized EM (GEM) Algorithm

For improved readability, the dependence on the discrete latent variable $\mathbf{x}_m$ is henceforth replaced by its subscript m, e.g., Σ_m refers to $\Sigma(\mathbf{x}_m)$. First, express the tangential manifold matrix in terms of $\mathbf{W}$,

$$\mathbf{E}_{//m} = \mathbf{W}\Big[\tfrac{\partial \boldsymbol{\phi}(\mathbf{x}_m)}{\partial x_1} \quad \cdots \quad \tfrac{\partial \boldsymbol{\phi}(\mathbf{x}_m)}{\partial x_Q}\Big] = \mathbf{W}\Theta_m, \tag{27}$$

according to (21). Note that a key assumption used in the ensuing derivations is that $\mathbf{W}\Theta_m$ is orthonormal, which is not true in general, as mentioned in the beginning of Section 4.3. In practice, $\mathbf{E}_{//m}$ is orthonormalized at each iteration. Next, the inverse PPS covariance (24) is written in terms of $\mathbf{W}$ using (27),

$$\begin{aligned}\Sigma_m^{-1} &= \frac{\mathbf{I}_D}{B} - \frac{S-B}{BS}\mathbf{E}_{//m}\mathbf{E}_{//m}^T\\ &= \frac{\mathbf{I}_D}{B} - \frac{S-B}{BS}\mathbf{W}\Theta_m\Theta_m^T\mathbf{W}^T\\ &= \frac{\mathbf{I}_D}{B} - \frac{S-B}{BS}\mathbf{W}\Psi_m\mathbf{W}^T,\end{aligned}$$

where we have defined

$$\Psi_m = \Theta_m\Theta_m^T.$$

Consider the conditional probability term in (19) and discarding constant terms,

$$\begin{aligned}&\ln p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)\\ &\quad= -\frac{1}{2}\ln\left|\Sigma_m\right| - \frac{1}{2}[\mathbf{y}_n - \mathbf{f}(\mathbf{x}_m;\mathbf{W})]^T\Sigma_m^{-1}[\mathbf{y}_n - \mathbf{f}(\mathbf{x}_m;\mathbf{W})]\\ &\quad= -\frac{1}{2}\ln(S^QB^{D-Q}) - \frac{\|\mathbf{y}_n - \mathbf{W}\phi_m\|^2}{2B} + \frac{S-B}{2BS}\\ &\qquad(\mathbf{y}_n - \mathbf{W}\phi_m)^T\mathbf{W}\Psi_m\mathbf{W}^T(\mathbf{y}_n - \mathbf{W}\phi_m).\end{aligned} \tag{28}$$

We note that for $\alpha = 1$ ($S = B = 1/\beta$), the last term in (28) vanishes and the GTM expression is recovered.

C.1 Update Equation for $\mathbf{W}$

Concentrating on just the last term in (28) and for the moment disregarding the constant coefficients and subscripts m, n, we have

$$\begin{aligned}&(\mathbf{y} - \mathbf{W}\boldsymbol{\phi})^T\mathbf{W}\Psi\mathbf{W}^T(\mathbf{y} - \mathbf{W}\boldsymbol{\phi})\\ &\quad= \text{tr}\left[\left(\mathbf{W}^T\mathbf{y} - \mathbf{W}^T\mathbf{W}\boldsymbol{\phi}\right)^T\Psi\left(\mathbf{W}^T\mathbf{y} - \mathbf{W}^T\mathbf{W}\boldsymbol{\phi}\right)\right]\\ &\quad= \text{tr}\left[\mathbf{C}\mathbf{C}^T\Psi\right],\end{aligned} \tag{29}$$

where we have defined $\mathbf{C} = \mathbf{W}^T\mathbf{y} - \mathbf{W}^T\mathbf{W}\boldsymbol{\phi}$. The derivative of (29) with respect $\mathbf{W}$ is

$$\begin{aligned}\frac{d\ \text{tr}\left[\mathbf{C}\mathbf{C}^T\Psi\right]}{d\mathbf{W}} &= 2\left[(\mathbf{y} - \mathbf{W}\boldsymbol{\phi})\mathbf{C}^T\Psi - \mathbf{W}\Psi\mathbf{C}\boldsymbol{\phi}^T\right]\\ &= 2(\mathbf{y} - \mathbf{W}\boldsymbol{\phi})(\mathbf{y} - \mathbf{W}\boldsymbol{\phi})^T\mathbf{W}\Psi - 2\mathbf{W}\Psi\mathbf{W}^T(\mathbf{y} - \mathbf{W}\boldsymbol{\phi})\boldsymbol{\phi}^T.\end{aligned} \tag{30}$$

Therefore, the derivative of the penalized log-likelihood can be expressed as

$$\frac{d\langle\mathcal{L}_c\rangle}{d\mathbf{W}} = \sum_{n=1}^{N}\sum_{m=1}^{M} r_{mn}\left(\frac{d\ln p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)}{d\mathbf{W}} + \lambda\mathbf{W}\right),$$

where

$$\begin{aligned}&\frac{d\ln p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)}{d\mathbf{W}}\\ &\quad= \left[\frac{1}{B}\ \mathbf{I}_D - \frac{S-B}{BS}\mathbf{W}\Psi_m\mathbf{W}^T\right](\mathbf{y}_n - \mathbf{W}\boldsymbol{\phi}_m)\boldsymbol{\phi}_m^T + \frac{S-B}{BS}\\ &\qquad\left[(\mathbf{y}_n - \mathbf{W}\boldsymbol{\phi}_m)(\mathbf{y}_n - \mathbf{W}\boldsymbol{\phi}_m)^T\mathbf{W}\boldsymbol{\Psi}_m\right].\end{aligned} \tag{31}$$

It can be seen from (31) that the derivative of the log-likelihood is nonlinear in $\mathbf{W}$ and, therefore, an analytic solution for $\mathbf{W}$ does not exist in the M-step. At this point, there are two possible approaches to finding $\mathbf{W}_{\text{new}}$. The first approach [46], [47] simply uses an approximation by solving the original GTM-likelihood equations (which assumes $\Sigma = 1/\beta$), i.e., solving for $\mathbf{W}_{\text{new}}^{\mathbf{T}}$ in the penalized log-likelihood equation

$$\left(\Phi\mathbf{G}_{\text{old}}\ \Phi^T + \lambda\mathbf{I}_L\right)\mathbf{W}_{\text{new}}^T = \Phi\mathbf{R}_{\text{old}}\mathbf{Y}^T,$$

where

$$\begin{aligned}\Phi &= [\,\boldsymbol{\phi}_1 \quad \cdots \quad \boldsymbol{\phi}_M\,]_{L\times M}\\ \mathbf{G}_{\text{old}} &= \text{diag}\left([\,\textstyle\sum_{n=1}^{N} r_{1n}^{\text{old}} \quad \cdots \quad \textstyle\sum_{n=1}^{N} r_{Mn}^{\text{old}}\,]\right)\\ \mathbf{Y} &= [\,\mathbf{y}_1 \quad \cdots \quad \mathbf{y}_N\,]_{D\times N},\end{aligned}$$

with $\mathbf{R}_{\text{old}} = \left\{r_{mn}^{\text{old}}\right\}$ as given in (20). In the second approach, we can iteratively update $\mathbf{W}$ using steepest ascent (with learning rate η)

$$\mathbf{W}^{k+1} = \mathbf{W}^k + \eta\frac{d\langle\mathcal{L}_c\rangle}{d\mathbf{W}}$$

until a local maxima is found. The second solution, though more computationally intensive, may be desirable as it fits into the generalized EM framework, which guarantees convergence [26].

C.2 Update Equation for β

First, we express $\ln p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)$ in (28) in terms of its original parameters α and β. After discarding constant terms, we have

$$\begin{aligned}&\ln p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m) =\\ &-\frac{1}{2}\ln\left[\left(\frac{1}{\beta}\right)^D\alpha^Q\left(\frac{D-\alpha Q}{D-Q}\right)^{D-Q}\right]\\ &-\frac{\beta(D-Q)}{2(D-\alpha Q)}\|\mathbf{y}_n - \mathbf{W}\boldsymbol{\phi}_m\|^2\\ &+\frac{\beta(\alpha-1)D}{2\alpha(D-\alpha Q)}(\mathbf{y}_n - \mathbf{W}\boldsymbol{\phi}_m)^T\mathbf{W}\Psi_m\mathbf{W}^T(\mathbf{y}_n - \mathbf{W}\boldsymbol{\phi}_m).\end{aligned} \tag{32}$$

Next, taking the derivative of (32) with respect to β yields

$$\begin{aligned}\frac{d\ln p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)}{d\beta} &= \frac{D}{2\beta} - \frac{D-Q}{2(D-\alpha Q)}\parallel \mathbf{y}_n - \mathbf{W}\boldsymbol{\phi}_m \parallel^2\\ &+\frac{(\alpha-1)D}{2\alpha(D-\alpha Q)}(\mathbf{y}_n - \mathbf{W}\boldsymbol{\phi}_m)^T\mathbf{W}\Psi_m\mathbf{W}^T(\mathbf{y}_n - \mathbf{W}\boldsymbol{\phi}_m),\end{aligned}$$

which can be substituted into the derivative of the log-likelihood to give

$$\begin{aligned}&\frac{d\langle\mathcal{L}_c\rangle}{d\beta}\\&=\sum_{n=1}^{N}\sum_{m=1}^{M}r_{mn}\frac{d\ln p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)}{d\beta}\\&=\sum_{n=1}^{N}\sum_{m=1}^{M}r_{mn}\left[\frac{D}{2\beta}-\frac{D-Q}{2(D-\alpha Q)}\|\mathbf{y}_n-\mathbf{W}\boldsymbol{\phi}_m\|^2\right.\\&\left.+\frac{(\alpha-1)D}{2\alpha(D-\alpha Q)}(\mathbf{y}_n-\mathbf{W}\boldsymbol{\phi}_m)^T\mathbf{W}\Psi_m\mathbf{W}^T(\mathbf{y}_n-\mathbf{W}\boldsymbol{\phi}_m)\right].\end{aligned}$$

Finally, solving for the stationary point of the above expression results in

$$\begin{aligned}&\frac{1}{\beta_{\text{new}}}\\&=\frac{1}{ND}\sum_{n=1}^{N}\sum_{m=1}^{M}r_{mn}^{\text{old}}\left[\frac{D-Q}{(D-\alpha Q)}\|\mathbf{y}_n-\mathbf{W}_{\text{new}}\phi_m\|^2-\frac{(\alpha-1)D}{\alpha(D-\alpha Q)}\right.\\&\left.(\mathbf{y}_n-\mathbf{W}_{\text{new}}\boldsymbol{\phi}_m)^T\mathbf{W}_{\text{new}}\Psi_m\mathbf{W}_{\text{new}}^T(\mathbf{y}_n-\mathbf{W}_{\text{new}}\boldsymbol{\phi}_m)\right].\end{aligned}\tag{34}$$

Note that setting $\alpha=1$ in (33) recovers the corresponding GTM update equation for β.

C.3 Update Equation for α

The derivative of (32) with respect to α yields

$$\begin{aligned}&\frac{d\ln p_{\vec{Y}|\vec{X}}(\mathbf{y}_n|\mathbf{x}_m)}{d\alpha}=\\&-\frac{QD(1-\alpha)}{2\alpha(D-\alpha Q)}+\frac{\beta Q(D-Q)}{2(D-\alpha Q)^2}\|\mathbf{y}_n-\mathbf{W}\boldsymbol{\phi}_m\|^2\\&+\frac{\beta D(D-2\alpha Q+\alpha^2 Q)}{2\alpha^2(D-\alpha Q)^2}(\mathbf{y}_n-\mathbf{W}\boldsymbol{\phi}_m)^T\mathbf{W}\Psi_m\mathbf{W}^T(\mathbf{y}_n-\mathbf{W}\boldsymbol{\phi}_m).\end{aligned}\tag{34}$$

Substituting (34) into the derivative of the log-likelihood function and solving for its stationary point, we obtain a cubic equation in α

$$\begin{aligned}&NQD\alpha(1-\alpha)(D-\alpha Q)=\\&\qquad\alpha^2\beta_{\text{new}}Q(D-Q)V_1+\beta_{\text{new}}D\big(D-2\alpha Q+\alpha^2 Q\big)V_2,\end{aligned}$$

where

$$\begin{aligned}V_1&=\sum_{n=1}^{N}\sum_{m=1}^{M}r_{mn}^{\text{old}}\|\mathbf{y}_n-\mathbf{W}_{\text{new}}\boldsymbol{\phi}_m\|^2,\\V_2&=\sum_{n=1}^{N}\sum_{m=1}^{M}r_{mn}^{\text{old}}(\mathbf{y}_n-\mathbf{W}_{\text{new}}\boldsymbol{\phi}_m)^T\\&\qquad\mathbf{W}_{\text{new}}\Psi_m^T\mathbf{W}_{\text{new}}(\mathbf{y}_n-\mathbf{W}_{\text{new}}\boldsymbol{\phi}_m),\end{aligned}$$

from which a valid α_{new} can be solved for.

Appendix D
Software Availability

The latest GNU Public Licensed (GPL) version of the PPS toolbox (MATLAB) is available for download at http://lans.ece.utexas.edu/~kuiyu.

Acknowledgments

This research was supported in part by US Army Research contracts DAAG55-98-1-0230 and DAAD19-99-1-0012, and US National Science Foundation grant ECS-9900353. The authors would like to thank Markus Svensen for his insightful comments and suggestions, Tai-Wu Chiang for his valuable discussions, and Shailesh Kumar for his careful proof reading. They would also like to thank the anonymous reviewers for their informative suggestions and comments.

References

[1] J.H. Friedman, "An Overview of Predictive Learning and Function Approximation," *From Statistics to Neural Networks, Proc. NATO/ASI Workshop,* V. Cherkassky, J.H. Friedman, and H. Wechsler, eds., pp. 1-61, 1994.

[2] R. Fisher, "The Case of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics,* vol. 7, no. part II, pp. 179-188, 1936.

[3] R.A. Johnson and D.W. Wichern, *Applied Multivariate Statistical Analysis,* third ed. Englewood Cliffs, N.J.: Prentice Hall, 1992.

[4] A. Hyvärinen, "Survey on Independent Component Analysis," *Neural Computing Surveys,* vol. 2, pp. 94-128, 1999.

[5] J.H. Friedman, "Exploratory Projection Pursuit," *J. Am. Statistical Assoc.,* vol. 82, no. 397, pp. 249-266, Mar. 1987.

[6] L. Fahrmeir and G. Tutz, *Multivariate Statistical Modelling Based on Generalized Linear Models.* New York: Springer-Verlag, 1994.

[7] G.W. Cottrell, P. Munroe, and D. Zipser, "Image Compression by Back Propagation: An Example of Extensional Programming," Technical Report ICS 8702, Univ. of California at San Diego, 1987.

[8] T. Kohonen, *Self-Organizing Maps.* Berlin, Heidelberg: Springer, 1995.

[9] T. Hastie and W. Stuetzle, "Principal Curves," *J. Am. Statistical Assoc.,* vol. 84, no. 406, pp. 502-516, June 1988.

[10] M. LeBlanc and R. Tibshirani, "Adaptive Principal Surfaces," *J. Am. Statistical Assoc.,* vol. 89, no. 425, pp. 53-64, Mar. 1994.

[11] C.M. Bishop, *Neural Networks for Pattern Recognition,* first ed. Oxford: Clarendon Press, 1995.

[12] M.E. Tipping and C.M. Bishop, "Mixtures of Probabilistic Principal Component Analysers," Technical Report NCRG/97/003, Aston Univ., June 1997.

[13] C.M. Bishop and M.E. Tipping, "A Hierarchical Latent Variable Model for Data Visualization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 3, pp. 281-293, Mar. 1998.

[14] H. Attias, "Independent Factor Analysis," *Neural Computation,* vol. 11, no. 4, pp. 803-851, May 1999.

[15] K.-y. Chang and J. Ghosh, "Probabilistic Principal Surfaces," *Proc. Int'l Joint Conf. Neural Networks,* p. 605, July 1999.

[16] C.M. Bishop, M. Svensén, and C.K.I. Williams, "GTM: The Generative Topographic Mapping," *Neural Computation,* vol. 10, no. 1, pp. 215-235, 1998.

[17] H. Ritter, T. Martinetz, and K. Schulten, *Neural Computation and Self-Organizing Maps: An Introduction.* Reading, Mass.: Addison-Wesley, 1992.

[18] F. Mulier and V. Cherkassky, "Self-Organization as an Iterative Kernel Smoothing Process," *Neural Computation,* vol. 7, pp. 1165-1177, 1995.

[19] C. de Boor, *A Practical Guide to Splines.* New York: Springer-Verlag, 1978.

[20] W. Cleveland and S. Devlin, "Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting," *J. Am. Statistical Assoc.,* vol. 83, pp. 596-610, 1988.

[21] T. Duchamp and W. Stuetzle, "Extremal Properties of Principal Curves in the Plane," *Annals of Statistics,* vol. 24, no. 4, pp. 1511-1520, 1996.

[22] J.D. Banfield and A.E. Raftery, "Ice Floe Identification in Satellite Images Using Mathematical Morphology and Clustering about Principal Curves," *J. Am. Statistical Assoc.,* vol. 87, no. 417, pp. 7-16, Mar 1992.

[23] K.-y. Chang and J. Ghosh, "Principal Curves for Nonlinear Feature Extraction and Classification," *SPIE: Applications of Artificial Neural Networks in Image Processing III,* vol. 3307, pp. 120-129, Jan. 1998.

[24] R. Tibshirani, "Principal Curves Revisited," *Statistics and Computing,* vol. 2, pp. 183-190, 1992.

[25] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum-Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc.,* vol. 39, no. 1, pp. 1-38, 1977.

[26] C.F.J. Wu, "On the Convergence Properties of the EM Algorithm," *Annals of Statistics,* vol. 11, pp. 95-103, 1983.

[27] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger, "Learning and Design of Principal Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 3, pp. 281-297, Mar. 2000.

[28] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger, "Principal Curves: Learning and Convergence," *Proc. IEEE Int'l Symp. Information Theory,* 1998.

[29] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger, "A Polygonal Line Algorithm for Constructing Principal Curves," *Neural Information Processing Systems,* vol. 11, pp. 501-507, 1998.

[30] P. Delicado, "Principal Curves and Principal Oriented Points," Technical Report 309, Departament d'Economia i Empresa, Universitat Pompeu Fabra, 1998.

[31] P. Delicado, "Another Look at Principal Curves and Surfaces," unpublished, 1999.

[32] T. Hastie, "Principal Curves and Surfaces," PhD thesis, Stanford Univ., 1984.

[33] J.H. Friedman, "Multivariate Adaptive Regression Splines," *Annals of Statistics,* vol. 19, pp. 1-141, 1991.

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

[35] E. Erwin, K. Obermayer, and K. Schulten, "Self-Organizing Maps: Ordering, Convergence Properties, and Energy Functions," *Biological Cybernetics,* vol. 67, pp. 47-55, 1992.

[36] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation,* Redwood City: Calif.: Addison-Wesley, 1991.

[37] C.M. Bishop and M. Svensén, "GTM: The Generative Topographic Mapping," Technical Report NCRG/96/015, Aston Univ., Apr. 1997.

[38] T.K. Moon, "The Expectation-Maximization Algorithm," *IEEE Signal Processing Magazine,* vol. 13, no. 6, pp. 47-60, Nov. 1996.

[39] K.-y. Chang, "Image and Signal Processing Using Neural Networks," MS thesis, Dept. of Electrical Eng., Univ. of Hawaii at Manoa, Dec. 1994.

[40] H. Bourland and Y. Kamp, "Auto-Association by Multilayer Perceptrons and Singular Value Decomposition," *Biological Cybernetics,* vol. 59, pp. 291-294, 1988.

[41] M.A. Kramer, "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks," *Am. Inst. Chemical Eng. J.,* vol. 37, no. 2, pp. 233-243, 1991.

[42] S. Tan and M.L. Mavrovouniotis, "Reducing Data Dimensionality through Optimizing Neural Network Inputs," *Am. Inst. Chemical Eng. J.,* vol. 41, no. 6, pp. 1471-1480, June 1995.

[43] D. Dong and T.J. McAvoy, "Nonlinear Principal Component Analysis-Based on Principal Curves and Neural Networks," *Computers and Chemical Eng.,* vol. 20, no. 1, pp. 65-78, 1996.

[44] J. Karhunen, E. Oja, L. Wang, R. Vigário, and J. Joutsensalo, "A Class of Neural Networks for Independent Component Analysis," *IEEE Trans. Neural Networks,* vol. 8, no. 3, pp. 486-504, May 1997.

[45] E.C. Malthouse, "Limitations of Nonlinear PCA as Performed with Generic Neural Networks," *IEEE Trans. Neural Networks,* vol. 9, no. 1, pp. 165-173, Jan. 1998.

[46] C.M. Bishop, M. Svensén, and C.K.I. Williams, "Developments of the Generative Topographic Mapping," *Neurocomputing,* vol. 21, pp. 203-224, 1998.

[47] M. Svensén, "GTM: The Generative Topographic Mapping," PhD thesis, Aston Univ., Birmingham, UK, 1998, http://www.ncrg.aston.ac.uk/GTM/.

[48] C.L. Blake and C.J. Merz, "UCI Repository of Machine Learning Databases," 1998.

[49] K.-y. Chang and J. Ghosh, "Three-Dimensional Model-Based Object Recognition and Pose Estimation Using Probabilistic Principal Surfaces," *SPIE: Applications of Artificial Neural Networks in Image Processing V,* pp. 192-203, Jan. 2000.

[50] K.-y. Chang, "Nonlinear Dimensionality Reduction Using Probabilistic Principal Surfaces," PhD thesis, Dept. of Electrical and Computer Eng., Univ. of Texas at Austin, May 2000.

[51] J.R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics.* John Wiley & Sons, 1988.

Kui-yu Chang received the BS degree in electrical engineering from the National Taiwan University (1992), the MS degree in electrical engineering from the University of Hawaii at Manoa (1994), and the PhD degree in electrical and computer engineering from the University of Texas at Austin (2000). His research interests include statistical pattern recognition, artificial neural networks, and data-mining. Since July 2000, he has been working as a member of the technical staff at Interwoven, Inc. (Austin wireless division).

Joydeep Ghosh received the BTech degree from the Indian Institute of Technology, Kanpur in 1983 and the MS and PhD degrees from the University of Southern California, Los Angeles, in 1988. Subsequent to that, he joined the the Department of Electrical and Computer Engineering at the University of Texas, Austin, where he has been a full professor since 1998 and a holder of the Archie Straiton Endowed Fellowship. Dr. Ghosh directs the Laboratory for Artificial Neural Systems (LANS), where his research group is studying the theory and applications of adaptive pattern recognition, data mining, including web mining, and multilearner systems. He has published more than 200 refereed papers and edited eight books. He received gold medals for standing first in high school and preuniversity, both with record scores, the NTS scholarship (1977-1983), All-University Predoctoral Merit Fellowship (first such engineering recipient at USC, 1983-1987), Haliburton Excellence Award (1993), Engineering Foundation Faculty Award (1994), and Dean's Fellow (1997). Six of his papers have received awards, including the 1992 Darlington Prize for the Best Journal Paper from *IEEE Circuits and Systems Society* and Best Applications Paper at ANNIE '97. Dr. Ghosh served as the general chairman for the SPIE/SPSE Conference on Image Processing Architecture, (1990), as conference cochair of Artificial Neural Networks in Engineering (ANNIE) in 1993, 1996, 1998, 1999, and 2000, and on the program or organizing committees of several conferences on neural networks and parallel processing. More recently, he has coorganized workshops on web mining (with SIAM International Conference on Data Mining, 2001) and one on parallel and distributed data mining (with KDD-2000). He is an associate editor of *Pattern Recognition*, *IEEE Transactions on Neural Networks, Neural Computing Surveys*, and the *International Journal of Smart Engineering Design*. He was a plenary speaker for ANNIE '97. He is a member of the IEEE.