Published in final edited form as:
Pattern Recognit Lett. 2014 March 1; 38: 120–131. doi:10.1016/j.patrec.2013.11.022.

Manifold Learning by Preserving Distance Orders

Esra Ataer-Cansizoglu[a,1], **Murat Akcakaya**[a,1], **Umut Orhan**[a], and **Deniz Erdogmus**[a]

Esra Ataer-Cansizoglu: ataer@ece.neu.edu; Murat Akcakaya: akcakaya@ece.neu.edu; Umut Orhan: orhan@ece.neu.edu; Deniz Erdogmus: erdogmus@ece.neu.edu

[a]Cognitive Systems Laboratory, Northeastern University, Boston, MA

Abstract

Nonlinear dimensionality reduction is essential for the analysis and the interpretation of high dimensional data sets. In this manuscript, we propose a distance order preserving manifold learning algorithm that extends the basic mean-squared error cost function used mainly in multidimensional scaling (MDS)-based methods. We develop a constrained optimization problem by assuming explicit constraints on the order of distances in the low-dimensional space. In this optimization problem, as a generalization of MDS, instead of forcing a linear relationship between the distances in the high-dimensional original and low-dimensional projection space, we learn a non-decreasing relation approximated by radial basis functions. We compare the proposed method with existing manifold learning algorithms using synthetic datasets based on the commonly used residual variance and proposed percentage of violated distance orders metrics. We also perform experiments on a retinal image dataset used in Retinopathy of Prematurity (ROP) diagnosis.

Keywords

Machine Learning; Manifold Learning; Nonlinear Dimensionality Reduction

1. Introduction

Due to the recent advances, acquisition of large volumes of high dimensional data has become more common in every aspect of daily life: stock market, social media, medical data, etc. Analysis and interpretation of such data requires finding meaningful low-dimensional structures in these huge data sets. Manifold learning attempts to accomplish such data explorations and dimensionality reductions.

Manifold learning can be regarded as identifying a nonlinear mapping from the original higher dimensional data space to a lower dimensional representation. Existing methods can be classified into three categories: global methods that tend to preserve global properties in

Correspondence to: Esra Ataer-Cansizoglu, ataer@ece.neu.edu.

[1]Esra Ataer-Cansizoglu and Murat Akcakaya contributed equally to this manuscript.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

the low-dimensional representation, local methods that aim to preserve the local geometry in the embedded space and techniques based on global alignment of multiple linear models [1]. Multidimensional scaling (MDS) is one of the global methods that finds a projection of the original data while preserving the pairwise Euclidean distances [2]. In the literature, various techniques are proposed to minimize MDS cost function [3, 4, 5, 6]. Similarly, in Isomap, one uses a geodesic distance estimation to use with MDS [7]. Different variations of Isomap have been proposed in the literature: landmark and conformal Isomap [8]. On the other hand, local methods [9, 10, 11, 12, 13, 14, 15] constructs the lower dimensional data using the local linear relations in the original space. Local tangent space alignment (LTSA) [9] represents the local geometry of the manifold with local tangent spaces that are learned through the neighborhood of each sample. Similarly, local linear embedding (LLE) [10] aims to preserve local neighborhood information, while Semidefinite Embedding (SDE) [11] involves preserving local isometries on a k-nn graph. Coifman et al. [14] presents a method that constructs local coordinates by learning a family of diffusion maps (DM). Another use of local geometry is by locally smooth manifold learning (LSML) [12, 13] which is based on learning a warping function, that takes any sample in the manifold and generates its neighbors. Stochastic neighborhood embedding (SNE) [15] and its variations [16, 17] are among probabilistic approaches that construct the neighborhood relations based on Gaussian kernels. Although local approaches have computational advantages, they might have limitations in preserving global geometry, especially if the data is sparse. Other methods that are based on global alignment of linear models aim to combine the local and global techniques by fitting a number of local linear models and merging them with a global alignment. Local linear coordination (LLC) [18] and manifold charting [19] methods fall into this category.

In this manuscript, we propose a nonlinear dimensionality reduction method that extends the basic idea used by the MDS and its variations. Although the ultimate goal is to preserve the distance orders, MDS algorithm only focuses on minimizing the mean-squared error between the input and output distances [2]. There is no explicit constraint on the distance orders during the solution of this optimization problem. Moreover, this minimization results in a linear relationship between distance spaces. Linear fit assumption between the distances in the original and low-dimensional projection spaces is very restricted and embedding is achieved in this restrictive family. To address these two important issues, we first generalize the mean squared error cost function to include more general relations between distance spaces. Instead of assuming a predefined relationship, we propose to also learn the relationship between distances while we project the data from the original space. Our only assumption on the relationship is to have a monotonic non-decreasing function in order to preserve the distance relationships observed in the original space in the projected space. Then, we develop a constrained optimization problem by incorporating the distance orders as inequality constraints. As a solution of this problem, we not only learn the data in the projected space but also learn the non-linear relation between distance spaces. The final form of the proposed optimization problem is a generalization of the existing global MDS-based manifold learning algorithms such that the existing methods are approximate solutions of the simplified version of this problem. In this manuscript, we focus on the formulation and theoretical aspects of the problem. Possible acceleration of the proposed method by convex relaxations and further approximations to analyze real data will be part of our future research.

Another commonly used manifold learning algorithm which has a nonlinear mapping between distance spaces is Sammons mapping [20]. Sammon's map, a nonlinear extension of MDS, first maps the input data to a nonlinear predefined feature space and tries to preserve the distances in this feature space. Different than Sammon's mapping, we assume an unknown nonlinear relationship between the input and output distances while preserving the distance orders from the original space.

The rest of the paper is organized as follows: We first define the notation used throughout the paper in section 2.1. Next, problem formulation is presented in section 2.2. The solution of the optimization problem and performance evaluation metrics are explained in section 2.3 and section 2.4 respectively. In section 3, we report the experiments and results and the paper is concluded in section 4.

2. Learning Algorithm

In this section, we describe the proposed method for manifold learning. We first define the data model and notations to be used throughout the manuscript. Then using this model we formulate the desired manifold learning problem and develop our algorithm. We derive an optimization problem that solves the manifold learning algorithm, starting with the commonly used cost function, mean-squared error minimization, and demonstrate that this cost function can be extended to include different distance relations between the original and projection space data points, and explicit constraints that preserve distance orders in the projected space.

2.1. Data Model and Notation

We represent the original and the projected data spaces as $\mathscr{X}$ and $\mathscr{Y}$, respectively. Then, $\boldsymbol{x}_i \in \mathscr{X}$ and $\boldsymbol{y}_i \in \mathscr{Y}$ with $i = 1, \ldots, N$ are the data points. In this representation, $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors and N is the number of the data points. We assume that $\dim(\mathscr{X}) = d \geq \dim(\mathscr{Y}) = \tilde{d}$. Moreover, we have $d^x_{i,j}$ and $d^y_{i,j}$ as the distances between the ith and jth data points in the original and the low-dimensional data spaces, $\|\cdot\|$ represents the L2-norm of a vector.

2.2. Problem Formulation

We formulate the manifold learning algorithm as a constrained optimization problem. Our approach restricts the minimum mean-squared error solutions used by some existing manifold learning algorithms [2, 7]. Specifically, these aim to minimize the difference between the distances of any two points in the original and projected spaces. That is, the difference between $d^y_{i,j}$ and $d^x_{i,j}$ for $\forall i, j = 1, \ldots, N$ is minimized, which on average results in a linear relationship between each $d^x_{i,j}$ and $d^y_{i,j}$ pair (as a result of the least-square solution).

$$\min_{\boldsymbol{y}_k k=1,\ldots,N} \sum\nolimits_{i=1}^{N-1} \sum\nolimits_{j=i+1}^{N} \|d^y_{i,j} - d^x_{i,j}\|^2 \qquad (1)$$

where $d^{y}_{i,j}=\|\boldsymbol{y}_i-\boldsymbol{y}_j\|$ is the Euclidean distance and $d^{x}_{i,j}$ is the distance between the *i*th and *j*th points. Note here that the minimization is performed over the data points $\boldsymbol{y}_i$ in the projected space.

In the proposed algorithm, we compute $d^{x}_{i,j}$ as the estimated geodesic distance between the *i*th and *j*th data points. We follow the method described in [7] to compute the geodesic distances in the original space. First, the Euclidean distance between every pair of data points in the original space (data pairwise distance matrix) is computed. Then, a k-nearest neighbor (knn) graph or ε-ball graph is generated. That is, k-nearest neighbors of a data point or neighbors within ε distance for each datum is taken, and the edge lengths from points outside these areas to the reference datum are set to be infinite, and the pairwise distance matrix is updated accordingly. Finally, Floyd algorithm is applied over this matrix to find approximate geodesic distances between the data pairs [7]. Floyd's algorithm, an example of dynamic programming, finds the shortest path between each pair of vertices in a weighted graph [21].

In our algorithm, we propose to generalize the minimum mean-squared error approach in (1) to include a broader relationship between the distances in the original and the low-dimensional projection space. For that purpose we rewrite (1) as

$$\min_{\boldsymbol{y}_k k=1,\ldots,N}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|d^{y}_{i,j}-h(d^{x}_{i,j})\|^2 \quad (2)$$

where *h*(·) is a monotonic nondecreasing function. We represent the derivative of *h*(·) as

$$h'(d^{x}_{i,j})=\sum_{k=1}^{N-1}\sum_{l=k+1}^{N}w_{kl}k(d^{x}_{i,j}-d^{x}_{k,l}) \quad (3)$$

where *k*(η) is a translation-invariant kernel function and w_{kl} 's are the multiplicative coefficients. We force $w_{kl} \geq 0$ to have *h*(·) as a nondecreasing function. That is, we represent the derivative of *h*(·) as a nonnegative weighted sum of kernel functions. Monotonic nondecreasing functions *h*(·) will guarantee that we preserve the order of original distances in the projected space.

Then, we have

$$h(d^{x}_{i,j})=\sum_{k=1}^{N-1}\sum_{l=k+1}^{N}w_{kl}K(d^{x}_{i,j}-d^{x}_{k,l}) \quad (4)$$

where

$$K(d^{x}_{i,j}-d^{x}_{k,l})=\int_{-\infty}^{d^{x}_{i,j}}k(\xi-d^{x}_{k,l})d\xi \quad (5)$$

Kernel functions are positive semidefinite and hence are appropriate to represent the derivative of a monotonic nondecreasing function *h*(·) [22]. A translation-invariant kernel

$k(\eta)$ is chosen such that $0 \leq k(\eta) < \infty$, $\int_{-\infty}^{\infty} k(\eta)\mathrm{d}\eta=1$, and $\lim_{\eta\to 0} k(\eta) = \delta(\eta)$. We specifically use Gaussian radial basis functions (rbf), since they were shown to universally approximate any function with any desired accuracy [23]. Then, we rewrite (5) as

$$K(d^x_{i,j}-d^x_{k,l})=\Phi\left(\frac{d^x_{i,j}-d^x_{k,l}}{\sigma}\right) \quad (6)$$

where $\Phi(\cdot)$ is the Gaussian cumulative distribution function, and σ is the kernel width (standard deviation), which is estimated using Silverman's rule [24].

From (3), we have $M=\frac{N(N-1)}{2}$ different non-negative w_{kl} coefficients. We form the $M \times 1$ size vector $\boldsymbol{w}$ as

$$\boldsymbol{w}=[w_{12}w_{13}\cdots w_{(N-1)N}]^T=[w_1w_2\cdots w_M]^T \quad (7)$$

We also define $\boldsymbol{y_d}=[\boldsymbol{y}_1^T\cdots\boldsymbol{y}_N^T]^T$ as the $N d \tilde{\times} 1$ size data vector in the projected space. Using the definitions for $\boldsymbol{w}$ and $\boldsymbol{y_d}$, and the specifications provided for the function $h(\cdot)$, we then formulate the manifold learning algorithm as the following optimization problem.

$$\begin{aligned}&\min_{\boldsymbol{y_d},\boldsymbol{w}}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|d^y_{i,j}-h(d^x_{i,j})\|^2\\ &\qquad\text{s.t.}\boldsymbol{w}\succeq\mathbf{0}\end{aligned} \quad (8)$$

where $\succeq$ represents an element-wise inequality, such that we require each element of $\boldsymbol{w}$ is greater than or equal to zero. Note that in (8), we aim to avoid the global minimum solution where all y_i, $i = 1, \ldots, N$ are the same and $\boldsymbol{w} = \mathbf{0}$, which is not a valid projection solution. Therefore, we update the problem as

$$\begin{aligned}&\min_{\boldsymbol{y_d},\boldsymbol{w}} && \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|d^y_{i,j}-h(d^x_{i,j})\|^2\\ &\text{s.t.} && \boldsymbol{w}\succeq\mathbf{0}\text{and}\sum_{m=1}^{M}w_m>0\end{aligned} \quad (9)$$

The existing manifold learning algorithms do not explicitly enforce constraints to preserve the distance relationships observed in the original space. In our method, we propose to consider the following constraint: If T distance pairs in the original space are ordered as $d^x_{i_t,j_t}\leq d^x_{i'_t,j'_t}$, we require to have $d^y_{i_t,j_t}\leq d^y_{i'_t,j'_t}$. This is an approximation to preserving the order relation-ship between every distance pair in the original space. Selecting T achieves a reduction in complexity, but also may reflect preference about which distance orders are more important to preserve for the user. Then, we have the following optimization problem

$$\begin{aligned}&\min_{\boldsymbol{y_d},\boldsymbol{w}} && \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|d^y_{i,j}-h(d^x_{i,j})\|^2\\ &\text{s.t.} && \boldsymbol{w}\succeq\mathbf{0},\sum_{m=1}^{M}w_m>0\text{and}d^y_{i_t,j_t}\leq d^y_{i'_t,j'_t}t=1,\ldots,T\end{aligned} \quad (10)$$

Depending on the data under consideration, feasibility in (10) could be difficult to achieve, it might not even be possible to satisfy all the inequality constraints; therefore, to obtain a solution to the proposed constrained optimization problem, we further update it by perturbing (relaxing) the distance inequality constraints to

$$\begin{array}{cc}\underset{\boldsymbol{y_d},\boldsymbol{w},\boldsymbol{\xi}}{\min} & \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|d_{i,j}^{y}-h(d_{i,j}^{x})\|^2+\gamma\sum_{t=1}^{T}\alpha_t\xi_t \\ \text{s.t.} & \boldsymbol{w}\succeq\boldsymbol{0},\boldsymbol{\xi}\succeq\boldsymbol{0},\sum_{m=1}^{M}w_m>0\text{and}d_{i_t',j_t'}^{y}-d_{i_t,j_t}^{y}+\xi_t\succeq 0t=1,\ldots,T\end{array} \quad (11)$$

where ξ = [$\xi_1 \cdots \xi_T]^T$ is the vector of the perturbation slack variables which store the deviation from the nonlinear inequality constraints representing the order relationship between the distance pairs in the projection space, γ is the penalty factor forcing the problem to satisfy as many order relationships as possible, and α's demonstrate the importance of different ξ's such that preserving some inequalities could be more important than others.

In order to use in 2.3, we follow the above discussions, and define

- θ = $[\boldsymbol{y_d}^T, \boldsymbol{w}^T, \xi^T]^T$ is an $(N\tilde{d} + M + T) \times 1$ vector,
- $f(\boldsymbol{\theta})=\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|d_{i,j}^{y}-h(d_{i,j}^{x})\|^2+\gamma\sum_{t=1}^{T}\alpha_t\xi_t$,
- $\boldsymbol{A} = [\boldsymbol{0}\ \boldsymbol{I}]$ as $(M + T) \times (M + T + N\tilde{d})$ size matrix,
- $\boldsymbol{g}(\boldsymbol{\theta})=[g_1(\boldsymbol{\theta}),\ldots,g_T(\boldsymbol{\theta})]^T=[d_{i_1',j_1'}^{y}-d_{i_1,j_1}^{y}+\xi_1,\ldots,d_{i_T',j_T'}^{y}-d_{i_T,j_T}^{y}+\xi_T]^T$,
- $\boldsymbol{c}(\boldsymbol{\theta})=\begin{bmatrix}\boldsymbol{A\theta}\\ \boldsymbol{g}(\boldsymbol{\theta})\end{bmatrix}=[c_1(\boldsymbol{\theta}),\ldots,c_{M+2T}(\boldsymbol{\theta})]^T$.

Note that we improve the problem in (11) with the condition $\sum_{m=1}^{M}w_m \geq e_c$, and note also that due to the quadratic dependency of the cost function to $\boldsymbol{w}$, we can always find a scaled $\boldsymbol{w}$ to obtain the minimum of the cost function at $\sum_{m=1}^{M}w_m=e_c$, and hence we define $c_{eq}=\sum_{m=1}^{M}w_m-e_c=0$. Then, we rewrite the problem as

$$\begin{array}{cc}\underset{\boldsymbol{\theta}}{\min} & f(\boldsymbol{\theta}) \\ \text{s.t.} & c_{eq}=0\text{and}\boldsymbol{c}(\boldsymbol{\theta})\succeq\boldsymbol{0}\end{array} \quad (12)$$

In our implementation, we always used a chain of $T = M - 1$ inequalities $d_{i_1,j_1}^{y} \leq d_{i_2,j_2}^{y} \leq \ldots \leq d_{i_M,j_M}^{y}$ but one can choose arbitrary pairs of inequalities. In the following we also refer d_{i_k,j_k}^{y} as d_k^{y}, then $g_k(\boldsymbol{\theta})=d_{k+1}^{y}-d_k^{y}+\xi_k$.

2.3. Solution of the Optimization Problem

In this section, we explain the solution method that we employ to solve the constrained optimization problem proposed in (12). We first redefine the problem using extra slack variables as

$$\begin{array}{ll}\min\limits_{\boldsymbol{\theta},\boldsymbol{s}} & f(\boldsymbol{\theta}) \\ \text{s.t.} & c_{eq}{=}0, \boldsymbol{c}(\boldsymbol{\theta}) - \boldsymbol{s}{=}\mathbf{0}, \text{and} \boldsymbol{s} \succeq \mathbf{0}\end{array} \quad (13)$$

Then the Lagrangian of the problem is computed as

$$L(\boldsymbol{\theta}, \boldsymbol{s}, \boldsymbol{z}){=}f(\boldsymbol{\theta}) - c_{eq}q - (\boldsymbol{c}(\boldsymbol{\theta}) - \boldsymbol{s})^T \boldsymbol{z} \quad (14)$$

where q and $\boldsymbol{z} = [z_1, \ldots, z_{3M-2}]^T$ are the Lagrange multipliers.

Using the proposed problem in (13), and the Lagrangian in (14), we compute the Karush-Kuhn-Tucker (KKT) conditions for this problem [25]

$$\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) - q\nabla_{\boldsymbol{\theta}} c_{eq} - \boldsymbol{D}^T(\boldsymbol{\theta})\boldsymbol{z}{=}\mathbf{0} \quad (15a)$$

$$\boldsymbol{S}\boldsymbol{z}{=}\mathbf{0} \quad (15b)$$

$$c_{eq}{=}0 \quad (15c)$$

$$\boldsymbol{c}(\boldsymbol{\theta}) - \boldsymbol{s}{=}\mathbf{0} \quad (15d)$$

$$\boldsymbol{s} \succeq \mathbf{0} \text{and} \boldsymbol{z} \succeq \mathbf{0} \quad (15e)$$

where

- $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}){=}\left[\frac{\partial f(\boldsymbol{\theta})}{\partial \theta_1}, \ldots, \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_{N\tilde{d}+2M-1}}\right]^T$ is the gradient of the cost function $f(\theta)$ with respect to θ,
- $\nabla_\theta c_{eq} = \mathbf{1}$, a vector of ones,
- $\boldsymbol{D}(\theta)$ is the Jacobian of the vector of functions $\boldsymbol{c}(\theta)$, such that $\boldsymbol{D}^T(\theta) = [\nabla_\theta c_1(\theta) \cdots \nabla_\theta c_{3M-2}(\theta)]$,
- $\boldsymbol{S}$ is a diagonal matrix with the $\boldsymbol{s}$ vector in the diagonal, such that $\boldsymbol{S} = \text{diag}(\boldsymbol{s})$.

Interior-point methods were shown to outperform active set [26] and augmented Lagrangian methods [27] in large scale problems [28, 29]. We therefore solve the problem in (13) using interior-point methods such that we define $\boldsymbol{p}_\theta$, $\boldsymbol{p}_{\boldsymbol{s}}$, p_q, and $\boldsymbol{p}_{\boldsymbol{z}}$ as the step lengths in the variables θ, $\boldsymbol{s}$, q, and $\boldsymbol{z}$, respectively, and apply the Newton's method to compute the primal-dual update matrix as

$$\begin{bmatrix} \nabla^2_{\boldsymbol{\theta\theta}} L(\boldsymbol{\theta}, \boldsymbol{s}, \boldsymbol{z}) & 0 & -\nabla_{\boldsymbol{\theta}} c_{eq} & -\boldsymbol{D}^T(\boldsymbol{\theta}) \\ \mathbf{0} & \boldsymbol{Z} & \mathbf{0} & \boldsymbol{S} \\ \nabla_{\boldsymbol{\theta}} c_{eq}^T & \mathbf{0} & 0 & \mathbf{0} \\ \boldsymbol{D}(\boldsymbol{\theta}) & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_{\boldsymbol{\theta}} \\ \boldsymbol{p}_{\boldsymbol{s}} \\ p_q \\ \boldsymbol{p}_{\boldsymbol{z}} \end{bmatrix} = \begin{bmatrix} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) - q\nabla_{\boldsymbol{\theta}} c_{eq} - \boldsymbol{D}^T(\boldsymbol{\theta})\boldsymbol{z} \\ \boldsymbol{S}\boldsymbol{z} \\ c_{eq} \\ \boldsymbol{c}(\boldsymbol{\theta}) - \boldsymbol{s} \end{bmatrix} \quad (16)$$

where

- $\nabla^2_{\boldsymbol{\theta\theta}} L(\boldsymbol{\theta}, \boldsymbol{s}, \boldsymbol{z}) = \nabla^2_{\boldsymbol{\theta\theta}} f(\boldsymbol{\theta}) - \sum_{m=1}^{3M-2} \nabla^2_{\boldsymbol{\theta\theta}} c_m(\boldsymbol{\theta}) z_i$ is the Hessian of the Lagrangian, and
- $\boldsymbol{Z}$ = diag($\boldsymbol{z}$).

We solve the system in (16) using the algorithm described in [30, 31, 32]. We approximate the Hessian of the Lagrangian using limited-memory Broyden-Fletcher-Goldfarb-Shanno (lBFGS) method [33]. lBFGS is a limited memory approximation of the BFGS method to be used in large scale optimization problems [25]. We provide the derivations of gradient $\nabla_\theta f(\theta)$, Jacobian $\boldsymbol{D}(\theta)$, and Hessian $\nabla^2_{\boldsymbol{\theta\theta}} L(\boldsymbol{\theta}, \boldsymbol{s}, \boldsymbol{z})$ in the appendix. The dimension of unknowns, n, in our problem is $O(N^2)$ and, we know that lBFGS takes $O(n^2)$ time per iteration, hence the run time for the optimization algorithm per iteration is $O(N^4)$. The optimization method converges to a local minima due to the non-convexity of the proposed problem. Convex relaxations to this problem and convergence analysis is a topic of our future work.

2.4. Performance Analysis

In this section, we describe a metric that we define to demonstrate the performance of the proposed method. The metric we consider is the percentage of the pairwise inequalities violated in the low-dimensional space:

$$\tau = \frac{T_v}{T_t} * 100 \quad (17)$$

where $T_t = M(M-1)/2$ is the total number of distance inequalities since we always choose to employ a chain of $M - 1$ inequalities and T_υ is the number of distance relationships violated in the projected space. In the proposed method, we compute T_t considering the estimated geodesic distance relationships observed in the original space. Recall that in our algorithm, for example if the geodesic distance $d^x_{i,j}$ is larger than $d^x_{k,l}$, we propose to preserve such relationships approximately for every data points indexed with $i, j, k, l \in [1, \ldots, N]$ in the projected space. We then accordingly compute T_υ by counting violated order relationships. According to the definition in (17), the lower the τ is, the more accurate the subspace projection is. We refer to this metric as constraint violation percentage (CVP).

Another commonly used performance metric is the residual variance between the estimated geodesic distances in the original space and the Euclidean distances in the projected space [7]. Defining $\boldsymbol{d_y} = [d^y_{1,2}, \ldots, d^y_{N-1,N}]^T$ and $\boldsymbol{d_x} = [d^x_{1,2}, \ldots, d^x_{N-1,N}]^T$ as two $M \times 1$ vectors, the residual variance is computed using

$$R = 1 - \frac{cor(\boldsymbol{d_x}, \boldsymbol{d_y})}{\sqrt{\sigma^2_{d_y} \sigma^2_{d_x}}} \quad (18)$$

where

- $\sigma^2_{d_y}=\sum_{i=1}^{N-1}\sum_{j=i}^{N}(d^y_{i,j}-\bar{d}_y)^2$, with $\bar{d}_y$ as the mean,
- $\sigma^2_{d_x}=\sum_{i=1}^{N-1}\sum_{j=i}^{N}(d^x_{i,j}-\bar{d}_x)^2$, with $\bar{d}_x$ as the mean,
- $cor(\boldsymbol{d_x},\boldsymbol{d_y})=\sum_{i=1}^{N-1}\sum_{j=i}^{N}(d^x_{i,j}-\bar{d}_x)(d^y_{i,j}-\bar{d}_y)$.

3. Numerical Examples

We demonstrate the performance of the proposed method on synthetic datasets and compare with the following existing methods: Isomap [7], LTSA [9], LLE [10], SDE [11], LSML [12, 13], DM [14]. Penalty factor γ in (11) is taken as 1 and the scalar e_c that bounds the sum of w's in (12) is selected as 10 in our experiments. α coefficient that indicate the importance of preserving an order constraints is taken as 1 for all order constraints. The variables that are being optimized are initialized randomly. The geodesic distance estimation is done with a knn-graph where $k = 5$. Similarly, neighborhood information is represented through knn-graphs with $k = 5$ in Isomap, LTSA, SDE and LSML methods. The number of neighbors for LLE is set as $k = 5$. We use the default values for all other parameters in the methods, that we compare against the proposed technique.

In the first set of experiments, we perform noise analysis on the synthetically generated growing band (GB) dataset with $N = 50$ samples, which contains a single dimensional manifold (see Figure 2(a)). For a given noise variance σ^2, the two dimensional dataset is generated by the following model

$$\mathbf{x}=[t\beta(t)]^T, t\in[0,\rho] \quad (19)$$

where ρ is selected as 10 and $\beta(t) = \mathcal{N}(0, \sigma^2 t)$ is a Gaussian random variable with mean 0 and variance $\sigma^2 t$. For this data set we use different noise levels, σ^2 changing from 0.001 to 0.1. In this experiment, our goal is to explore possible non-linear relationship between the distances in the original and the projection spaces. The performance metrics we use are constraint violation percentage (CVP) and residual variance as described in Section 2.4. In Figure 2, and Table 1, we report noise analysis results. Figure 2 displays the CVP and residual variance for different values of the noise variance using the proposed technique and all other techniques. Figure 2(b) is a zommed in version of Figure 2(a) displaying the results for LSML, DM, Isomap and the proposed methods. The residual variance for LLE is omitted in Figure 2(c) for better visualization, since it produces very high values compared to the other methods. As can be seen the proposed method shows a better performance than the other methods with respect to both evaluation criteria. The method that perfoms similar to the proposed method is Isomap. All the other methods are local and they perform poorly due to the sparsity of the data. The performance difference between Isomap and the proposed method increases as the noise variance increases, because for low noise variances, the data resembles to a line more and the distance relations are close to being linear. Notice that the residual variance shows the same trend with CVP, since both metrics aim to quantify the consistency between distances, see also Table 1 for the summary of performance analysis results. Figure 1(b) shows the $h(\cdot)$ function as a result of the proposed optimization scheme for $\sigma^2 = 0.1$. The resulting fit does not show a linear relation, instead it maps the small distances to even smaller distances and the large distances to even larger ones. Our analysis shows that this kind of relationship outperforms the linear fit assumed by MDS.

Secondly, we carry out experiments on a noisy spiral data with $N = 50$ samples (see Figure 3(a)). Resulting $h(\cdot)$ function is shown in Figure 3(b). The comparative results can be seen in Table 1. We outperform all other methods, but the performance of the proposed method is close to Isomap, and this can also be seen in $h(\cdot)$ function in Figure 3(b) which demonstrates an almost linear relationship. Even though we achieve an almost linear fitting, in the proposed method, we do not assume any predefined model in the algorithm, instead we learn this linear relationship as part of the solution. Other methods that give a comparable performance with the proposed method on this dataset are SDE and LSML.

Thirdly, we perform experiments on the Swiss roll dataset with $N = 85$ samples (Figure 4(a)). 3D samples from the original space is projected to a 2D space in this experiment. Our goal is to demonstrate the performance of our method on a dataset having different curvature levels. Table 1 and Figure 5 report the comparative results. Also, resulting $h(\cdot)$ function that represents the relationship between d^x and d^ys is reported in Figure 4(b). Since data is synthetically generated, original 2D coordinates are known and are displayed in Figure 5(a). Figures 5(b) and 5(c) show the 2D projection results with Isomap and the proposed methods, respectively. Each sample is identified with an index number and the numbers are displayed on the nodes overlayed with Delaunay triangulation edges for better visualization. In the figure, we only display the projection result for Isomap, since it has the closest performance to our method on this dataset. Although the 2D projection with the proposed method seems rotated with respect to the original data, it preserves the distance relations better than the Isomap method, and the results in Table 1 supports this observation. Note that, knn-graph does not contain any shortcuts between different sides of the Swiss roll due to the small k value. However, the motivation of the method is not to overcome the limitations of the geodesic distance estimation. Instead, we would like to solve the general optimization problem given an estimated distance matrix. With this purpose, same parameters are used for local neighborhood definitions in all of the methods.

Lastly, we carry out experiments on a retinal image dataset. Retinal images are widely used by doctors to follow, treat and diagnose various diseases. Retinopathy of prematurity (ROP) is among the diseases that can be diagnosed through the use of retinal images. It is a disease affecting low-birth weight infants, in which blood vessels in the retina of the eye develop abnormally and cause potential blindness. ROP is diagnosed from dilated retinal examination by an ophthalmologist, and may be successfully treated if diagnosed appropriately [34]. According to the international classification system, clinical ROP diagnosis has three classes, *plus disease*, *pre-plus* and *neither* [35]. *Plus disease* is a critical parameter which identifies severe ROP and is characterized by tortuosity of the arteries and dilation of the veins in the retina. *Pre-plus* represents vascular abnormalities insufficient for *plus disease* but with more arterial tortuosity and venous dilation than normal. Infants with *plus disease* require treatment to prevent blindness, whereas those without plus disease may be monitored by serial ophthalmic examination without treatment. Studies have found that clinical plus disease diagnosis is often subjective and qualitative, and that there is significant inconsistency even among experts [36, 37]. Hence, diagnosis of ROP is a vital and challenging task. In this experiment, we use a retinal image dataset, that consists of 34 images that are diagnosed by 22 experts [38]. Vessels are manually segmented in the images. Based on manual segmentation, we compute cumulative tortuosity for each center line point of vessels and employ mean and second central moment (CM2) of these values as features [39]. For a curve, cumulative tortuosity is defined as the ratio of the curve length to the distance between the two endpoints. As can be seen in the scatter plot in Figure 6(a), the features represent a one dimensional manifold with varying noise levels as in GB dataset. Figure 6(b) shows result of our manifold learning algorithm on the dataset along with some example images. Expert diagnostic decisions are also displayed on top of each image where the numbers represent the number of experts decided *plus disease* (+), *pre-plus* (±) and neither (−) respectively. The most important observation that we make from this figure is that the amount of tortuos vessels increases as we go through the manifold. Tortuosity plays an important role for ROP diagnosis. We also observe this role by analyzing the expert diagnostic decisions on the manifold such that the number of *plus-disease* decisions increases, while the number of *neither* decisions decreases as we trace the projected points from left to right. Moreover, the number of *pre-plus* decisions is high in the images in the middle of the projected curve compared to the images at the left and rightmost ends. In this example, preserving distance orders during dimensionality reduction is crucial, since tortuosity is a critical factor for ROP diagnosis.

4. Conclusion

We developed a nonlinear dimensionality reduction method that is a generalization of multidimensional scaling technique. Rather than using the common mean-squared error as an unconstrained cost function, we formulated a constrained optimization problem. In this problem, we assumed an unknown monotonic nondecreasing relationship modeled by radial basis function interpolation between the distances in the original and the projected spaces to be learned as a part of the manifold learning problem. Moreover, we incorporated explicit constraints on the distance orders in the projected space. Using interior-point methods, we solved this optimization problem, and in addition to obtaining the low-dimensional representation, we also learned the nonlinear relationship between the distances. We also proposed a new performance evaluation metric based on the number of the violated distance orders. Using this metric and residual variance as the performance measures, we compared our algorithm with other popular methods on synthetic datasets. The experiments demonstrated that the proposed method outperforms the other algorithms. The local algorithms performed poorly due to the sparsity of the data. The algorithm that performed close to the proposed method was Isomap, which is global and is a special case of our method. We applied the proposed algorithm on a real dataset where preserving distance orders is crucial for correct disease diagnosis. Future work includes extending the formulation of the optimization problem with convex relaxation to obtain faster solutions for larger datasets.

Acknowledgments

This work is partially supported by NIH 1R21EY022387-01A1, NSF IIS-1118061, NSF IIS-1149570, MGHPCC 2012 Seed Funds, and Northeastern University 2013 Tier 1 Seed Funds.

Appendix

In this appendix, we compute the gradient of the cost function, the Jacobian of the inequality constraints, and the Hessian of the Lagrangian that we define in (15) and (16). We first start with the derivation of the gradient $\nabla_\theta f(\theta)$. Following the definition of θ

$$\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})=\begin{bmatrix} \nabla_{\boldsymbol{y_d}} f(\boldsymbol{\theta}) \\ \nabla_{\boldsymbol{w}} f(\boldsymbol{\theta}) \\ \nabla_{\boldsymbol{\xi}} f(\boldsymbol{\theta}) \end{bmatrix}. \quad (20)$$

Then rewriting $h(d^x_{i,j})=\boldsymbol{w}^T\phi(i,j)$ and $\gamma\sum_{m=1}^{M}\alpha_m\xi_m=\gamma\boldsymbol{\xi}^T\boldsymbol{\alpha}$, where $\boldsymbol{w}$ is defined in (7), $\phi_{ij}=\left[\Phi\left(\frac{d^x_{i,j}-d^x_{1,2}}{\sigma}\right),\ldots,\Phi\left(\frac{d^x_{i,j}-d^x_{N-1,N}}{\sigma}\right)\right]^T$, and $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{M-1}]$, we compute

$$\nabla_{\boldsymbol{w}} f(\boldsymbol{\theta})=\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}-2[d^y_{i,j}-h(d^x_{i,j})]\phi_{ij} \quad (21a)$$

$$\nabla_{\boldsymbol{\xi}} f(\boldsymbol{\theta})=\gamma\boldsymbol{\alpha} \quad (21b)$$

Note here that $\nabla_{\boldsymbol{y_d}} f(\boldsymbol{\theta})=\left[\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_1^T},\ldots,\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_N^T}\right]^T$, then

$$\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_k}=\sum_{i=1 \text{and} i\neq k}^{N} 2[\,\|\boldsymbol{y}_i-\boldsymbol{y}_k\|-h(d^x_{i,k})]\|\boldsymbol{y}_i-\boldsymbol{y}_k\|^{-1}(\boldsymbol{y}_k-\boldsymbol{y}_i) \quad (22)$$

Recall that $\boldsymbol{D}^T(\theta) = [\nabla_\theta c_1(\theta) \cdots \nabla_\theta c_{3M-2}(\theta)]$, then using the definition of $\boldsymbol{c}(\theta)$, we have $\boldsymbol{D}^T(\theta) = [\boldsymbol{A}^T \nabla_\theta g_1(\theta) \cdots \nabla_\theta g_{M-1}(\theta)]$. Here $\nabla_{\boldsymbol{\theta}} g_n(\boldsymbol{\theta})=[\nabla^T_{\boldsymbol{y_d}} g_n(\boldsymbol{\theta})\nabla^T_{\boldsymbol{w}} g_n(\boldsymbol{\theta})\nabla^T_{\boldsymbol{\xi}} g_n(\boldsymbol{\theta})]^T$. If $d_{n+1}=d^y_{k,l}$ and $d_n=d^y_{i,j}$, from Section 2.2, $g_n(\theta) = \|\boldsymbol{y}_l-\boldsymbol{y}_k\| - \|\boldsymbol{y}_i-\boldsymbol{y}_j\| + \xi_n$, then $\nabla_{\boldsymbol{w}} g_n(\theta) = 0$, $\nabla_\xi g_n(\theta) = [0, \ldots, 0, 1, 0, \ldots, 0]$, a vector of zeros with 1 at the n^{th} location, and $\nabla_{\boldsymbol{y_d}} g_n(\boldsymbol{\theta})=[\nabla^T_{\boldsymbol{y}_1} g_n(\boldsymbol{\theta}),\ldots,\nabla^T_{\boldsymbol{y}_N} g_n(\boldsymbol{\theta})]^T$. From the definition of $g_n(\theta)$, we have $\nabla_{\boldsymbol{y}_m} g_n(\theta) = 0$ for $m = 1, \ldots, N$, and $m \neq i, j, k, l$, and

$$\nabla_{\boldsymbol{y}_i} g_n(\boldsymbol{\theta})=\frac{(\boldsymbol{y}_j-\boldsymbol{y}_i)}{\|\boldsymbol{y}_i-\boldsymbol{y}_j\|},\nabla_{\boldsymbol{y}_j} g_n(\boldsymbol{\theta})=\frac{(\boldsymbol{y}_i-\boldsymbol{y}_j)}{\|\boldsymbol{y}_i-\boldsymbol{y}_j\|},\ \nabla_{\boldsymbol{y}_k} g_n(\boldsymbol{\theta})=\frac{(\boldsymbol{y}_k-\boldsymbol{y}_l)}{\|\boldsymbol{y}_k-\boldsymbol{y}_l\|},\nabla_{\boldsymbol{y}_l} g_n(\boldsymbol{\theta})=\frac{(\boldsymbol{y}_l-\boldsymbol{y}_k)}{\|\boldsymbol{y}_k-\boldsymbol{y}_l\|} \quad (23)$$

Recall the definition of Hessian $\nabla^2_{\boldsymbol{\theta\theta}} L(\boldsymbol{\theta},\boldsymbol{s},\boldsymbol{z})=\nabla^2_{\boldsymbol{\theta\theta}} f(\boldsymbol{\theta})-\sum_{m=1}^{3M-2}\nabla^2_{\boldsymbol{\theta\theta}} c_m(\boldsymbol{\theta})z_i$, using this definition, we first demonstrate the computation of the Hessian of the cost function

$$\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} f(\boldsymbol{\theta})=\begin{bmatrix} \nabla^2_{\boldsymbol{y_d}\boldsymbol{y_d}} f(\boldsymbol{\theta}) & \nabla^2_{\boldsymbol{y_d}\boldsymbol{w}} f(\boldsymbol{\theta}) & \nabla^2_{\boldsymbol{y_d}\boldsymbol{\xi}} f(\boldsymbol{\theta}) \\ (\nabla^2_{\boldsymbol{y_d}\boldsymbol{w}} f(\boldsymbol{\theta}))^T & \nabla^2_{\boldsymbol{w}\boldsymbol{w}} f(\boldsymbol{\theta}) & \nabla^2_{\boldsymbol{w}\boldsymbol{\xi}} f(\boldsymbol{\theta}) \\ (\nabla^2_{\boldsymbol{y_d}\boldsymbol{\xi}} f(\boldsymbol{\theta}))^T & (\nabla^2_{\boldsymbol{w}\boldsymbol{\xi}} f(\boldsymbol{\theta}))^T & \nabla^2_{\boldsymbol{\xi}\boldsymbol{\xi}} f(\boldsymbol{\theta}) \end{bmatrix}. \quad (24)$$

Then using (21), it is easy to show that $\nabla^2_{\boldsymbol{y_d}\boldsymbol{\xi}} f(\boldsymbol{\theta})=\nabla^2_{\boldsymbol{w}\boldsymbol{\xi}} f(\boldsymbol{\theta})=\nabla^2_{\boldsymbol{\xi}\boldsymbol{\xi}} f(\boldsymbol{\theta})=\mathbf{0}$. Again from (21), we compute

$$\nabla^2_{\boldsymbol{w}\boldsymbol{w}} f(\boldsymbol{\theta})=\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} 2\phi_{ij}^T\phi_{ij}. \quad (25)$$

Noticing that $\nabla^2_{\boldsymbol{y_d}\boldsymbol{w}} f(\boldsymbol{\theta})=\begin{bmatrix} \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_1 \partial \boldsymbol{w}} \\ \cdots \\ \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_N \partial \boldsymbol{w}} \end{bmatrix}$, we have

$$\frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_k \partial \boldsymbol{w}}=\sum_{i=1 \text{and} i\neq k}^{N} - 2\phi_{ik}^T \|\boldsymbol{y}_i-\boldsymbol{y}_k\|^{-1}(\boldsymbol{y}_k-\boldsymbol{y}_i). \quad (26)$$

We can write $\nabla^2_{\boldsymbol{y_d}\boldsymbol{y_d}} f(\boldsymbol{\theta})=\begin{bmatrix} \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_1 \partial \boldsymbol{y}_1} & \cdots & \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_1 \partial \boldsymbol{y}_N} \\ \cdot & \cdot & \cdot \\ \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_N \partial \boldsymbol{y}_1} & \cdots & \frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_N \partial \boldsymbol{y}_N} \end{bmatrix}$, then

$$\frac{\partial^2 f(\boldsymbol{\theta})}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_j}=(\|\boldsymbol{y}_j-\boldsymbol{y}_k\|^{-1}-2)\boldsymbol{I}-\frac{(\boldsymbol{y}_j-\boldsymbol{y}_k)^T(\boldsymbol{y}_j-\boldsymbol{y}_k)}{\|\boldsymbol{y}_j-\boldsymbol{y}_k\|^3} \quad (27)$$

To complete the derivation of the Hessian of the Lagrangian, we continue with the computation of $\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} c_m(\boldsymbol{\theta})$ for $m = 1, \ldots, 3M - 2$. Recalling the formula for the Jacobian of the inequality constraints we derive above, it is straightforward to show that $\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} c_m(\boldsymbol{\theta})=\mathbf{0}$ for $m = 1, \ldots, 2M - 1$, and $\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} c_m(\boldsymbol{\theta})=\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} g_m(\boldsymbol{\theta})$ for $m = 2M, \ldots, 3M - 2$ such that

$$\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} g_m(\boldsymbol{\theta})=\begin{bmatrix} \nabla^2_{\boldsymbol{y_d}\boldsymbol{y_d}} g_m(\boldsymbol{\theta}) & \nabla^2_{\boldsymbol{y_d}\boldsymbol{w}} g_m(\boldsymbol{\theta}) & \nabla^2_{\boldsymbol{y_d}\boldsymbol{\xi}} g_m(\boldsymbol{\theta}) \\ (\nabla^2_{\boldsymbol{y_d}\boldsymbol{w}} g_m(\boldsymbol{\theta}))^T & \nabla^2_{\boldsymbol{w}\boldsymbol{w}} g_m(\boldsymbol{\theta}) & \nabla^2_{\boldsymbol{w}\boldsymbol{\xi}} g_m(\boldsymbol{\theta}) \\ (\nabla^2_{\boldsymbol{y_d}\boldsymbol{\xi}} g_m(\boldsymbol{\theta}))^T & (\nabla^2_{\boldsymbol{w}\boldsymbol{\xi}} g_m(\boldsymbol{\theta}))^T & \nabla^2_{\boldsymbol{\xi}\boldsymbol{\xi}} g_m(\boldsymbol{\theta}) \end{bmatrix}. \quad (28)$$

Then using (23) and the computation of the Jacobian matrix, we can show that $\nabla^2_{\boldsymbol{y_d}\boldsymbol{\xi}} g_m(\boldsymbol{\theta})=\nabla^2_{\boldsymbol{w}\boldsymbol{\xi}} g_m(\boldsymbol{\theta})=\nabla^2_{\boldsymbol{\xi}\boldsymbol{\xi}} g_m(\boldsymbol{\theta})=\nabla^2_{\boldsymbol{w}\boldsymbol{w}} g_m(\boldsymbol{\theta})=\nabla^2_{\boldsymbol{y_d}\boldsymbol{w}} g_m(\boldsymbol{\theta})=\mathbf{0}$, and

$$\nabla^2_{\boldsymbol{y_d}\boldsymbol{y_d}} g_m(\boldsymbol{\theta})=\begin{bmatrix} \nabla^2_{\boldsymbol{y}_1\boldsymbol{y}_1} g_m(\boldsymbol{\theta}) & \cdots & \nabla^2_{\boldsymbol{y}_1\boldsymbol{y}_N} g_m(\boldsymbol{\theta}) \\ \cdot & \cdot & \cdot \\ \nabla^2_{\boldsymbol{y}_N\boldsymbol{y}_1} g_m(\boldsymbol{\theta}) & \cdots & \nabla^2_{\boldsymbol{y}_N\boldsymbol{y}_N} g_m(\boldsymbol{\theta}) \end{bmatrix}. \quad (29)$$

From (23) and the definition of g_n (θ) recall that $\nabla_{y_m} g_n$ (θ) = 0 for *m* = 1, …, *N*, and *m* ≠ *i*, *j*, *k*, *l*, then $\nabla^2_{\boldsymbol{y}_m \boldsymbol{y}_{m'}} g_n(\boldsymbol{\theta})=\mathbf{0}$, for *m′* = 1, …, *N*, also $\nabla^2_{\boldsymbol{y}_i \boldsymbol{y}_k} g_n(\boldsymbol{\theta})=\nabla^2_{\boldsymbol{y}_i \boldsymbol{y}_l} g_n(\boldsymbol{\theta})=\nabla^2_{\boldsymbol{y}_j \boldsymbol{y}_k} g_n(\boldsymbol{\theta})=\nabla^2_{\boldsymbol{y}_j \boldsymbol{y}_l} g_n(\boldsymbol{\theta})=\mathbf{0}$. We compute

$$\begin{aligned}
\nabla^2_{\boldsymbol{y}_i \boldsymbol{y}_i} g_n(\boldsymbol{\theta}) &= \frac{-\boldsymbol{I}\|\boldsymbol{y}_i-\boldsymbol{y}_j\|^2+(\boldsymbol{y}_i-\boldsymbol{y}_j)(\boldsymbol{y}_i-\boldsymbol{y}_j)^T}{\|\boldsymbol{y}_i-\boldsymbol{y}_j\|^3}\\
\nabla^2_{\boldsymbol{y}_i \boldsymbol{y}_j} g_n(\boldsymbol{\theta}) &= \frac{\boldsymbol{I}\|\boldsymbol{y}_i-\boldsymbol{y}_j\|^2-(\boldsymbol{y}_j-\boldsymbol{y}_i)(\boldsymbol{y}_j-\boldsymbol{y}_i)^T}{\|\boldsymbol{y}_i-\boldsymbol{y}_j\|^3}\\
\nabla^2_{\boldsymbol{y}_j \boldsymbol{y}_j} g_n(\boldsymbol{\theta}) &= \frac{-\boldsymbol{I}\|\boldsymbol{y}_i-\boldsymbol{y}_j\|^2+(\boldsymbol{y}_j-\boldsymbol{y}_i)(\boldsymbol{y}_j-\boldsymbol{y}_i)^T}{\|\boldsymbol{y}_i-\boldsymbol{y}_j\|^3}\\
\nabla^2_{\boldsymbol{y}_k \boldsymbol{y}_k} g_n(\boldsymbol{\theta}) &= \frac{\boldsymbol{I}\|\boldsymbol{y}_k-\boldsymbol{y}_l\|^2-(\boldsymbol{y}_k-\boldsymbol{y}_l)(\boldsymbol{y}_k-\boldsymbol{y}_l)^T}{\|\boldsymbol{y}_k-\boldsymbol{y}_l\|^3}\\
\nabla^2_{\boldsymbol{y}_k \boldsymbol{y}_l} g_n(\boldsymbol{\theta}) &= \frac{-\boldsymbol{I}\|\boldsymbol{y}_k-\boldsymbol{y}_l\|^2+(\boldsymbol{y}_k-\boldsymbol{y}_l)(\boldsymbol{y}_k-\boldsymbol{y}_l)^T}{\|\boldsymbol{y}_k-\boldsymbol{y}_l\|^3}\\
\nabla^2_{\boldsymbol{y}_l \boldsymbol{y}_l} g_n(\boldsymbol{\theta}) &= \frac{\boldsymbol{I}\|\boldsymbol{y}_k-\boldsymbol{y}_l\|^2-(\boldsymbol{y}_l-\boldsymbol{y}_k)(\boldsymbol{y}_l-\boldsymbol{y}_k)^T}{\|\boldsymbol{y}_k-\boldsymbol{y}_l\|^3}.
\end{aligned} \tag{30}$$

References

1. Van der Maaten L, Postma E, Van Den Herik H. Dimensionality reduction: A comparative review. Journal of Machine Learning Research. 2009; 10:1–41.
2. Kruskal J. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika. 1964; 29(1):1–27.
3. Dzwinel W, Blasiak J. Method of particles in visual clustering of multidimensional and large data sets. Future Generation Computer Systems. 1999; 15(3):365–379.
4. Pawliczek P, Dzwinel W, Yuen DA. Visual exploration of data by using multidimensional scaling on multicore CPU, GPU, and MPI cluster. Concurrency and Computation: Practice and Experience.
5. Pawliczek P, Dzwinel W. Interactive Data Mining by Using Multidimensional Scaling. Procedia Computer Science. 2013; 18:40–49.
6. Andrecut M. Molecular dynamics multidimensional scaling. Physics Letters A. 2009; 373(23): 2001–2006.
7. Tenenbaum J, De Silva V, Langford J. A global geometric framework for nonlinear dimensionality reduction. Science. 2000; 290(5500):2319–2323. [PubMed: 11125149]
8. Silva V, Tenenbaum J. Global versus local methods in nonlinear dimensionality reduction. Advances in Neural Information Processing Systems (NIPS). 2002; 15:705–712.
9. Zhang Z, Zha H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. Journal of Shanghai University (English Edition). 2004; 8(4):406–424.
10. Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding. Science. 2000; 290(5500):2323–2326. [PubMed: 11125150]
11. Weinberger K, Saul L. Unsupervised learning of image manifolds by semidefinite programming. International Journal of Computer Vision. 2006; 70(1):77–90.
12. Dollár P, Rabaud V, Belongie S. Learning to Traverse Image Manifolds. Advances in Neural Information Processing Systems (NIPS). 2006
13. Dollár, P.; Rabaud, V.; Belongie, S. Non-Isometric Manifold Learning: Analysis and an Algorithm; International conference on Machine Learning (ICML); 2007.
14. Coifman RR, Lafon S. Diffusion maps. Applied and computational harmonic analysis. 2006; 21(1): 5–30.

15. Hinton GE, Roweis ST. Stochastic neighbor embedding. Advances in Neural Information Processing Systems (NIPS). 2002:833–840.

16. Xie B, Mu Y, Tao D, Huang K. m-SNE: Multiview stochastic neighbor embedding, IEEE Transactions on Systems. Man, and Cybernetics. 2011; 41(4):1088–1096.

17. Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research. 2008; 9(2579–2605):85.

18. Teh Y, Roweis S. Automatic alignment of local representations. Advances in Neural Information Processing Systems (NIPS). 2002; 15:841–848.

19. Brand M. Charting a manifold. Advances in Neural Information Processing Systems (NIPS). 2003:985–992.

20. Sammon J Jr. A nonlinear mapping for data structure analysis. IEEE Transactions on Computers. 1969; 100(5):401–409.

21. Floyd R. Algorithm 97: shortest path. Communications of the ACM. 1962; 5(6):345.

22. Hofmann T, Schölkopf B, Smola A. Kernel methods in machine learning. The annals of statistics. 2008:1171–1220.

23. Park J, Sandberg I. Universal approximation using radial-basis-function networks. Neural computation. 1991; 3(2):246–257.

24. Silverman, B. Density estimation for statistics and data analysis. Vol. 26. Chapman & Hall/CRC; 1986.

25. Nocedal, J.; Wright, S. Numerical optimization. Springer verlag; 1999.

26. Boggs P, Tolle J. Sequential quadratic programming. Acta numerica. 1996; 4(1):1–51.

27. Friedlander M, Saunders M. A globally convergent linearly constrained Lagrangian method for nonlinear optimization. SIAM Journal on Optimization. 2005; 15(3):863–897.

28. Forsgren A, Gill P, Wright M. Interior methods for nonlinear optimization. SIAM review. 2002; 44(4):525–597.

29. Gould N, Orban D, Toint P. Numerical methods for large-scale non-linear optimization. Acta Numerica. 2005; 14:299–361.

30. Byrd R, Gilbert J, Nocedal J. A trust region method based on interior point techniques for nonlinear programming. Mathematical Programming. 2000; 89(1):149–185.

31. Byrd R, Hribar M, Nocedal J. An interior point algorithm for large-scale nonlinear programming. SIAM Journal on Optimization. 1999; 9(4):877–900.

32. Waltz R, Morales J, Nocedal J, Orban D. An interior algorithm for nonlinear optimization that combines line search and trust region steps. Mathematical Programming. 2006; 107(3):391–408.

33. Nocedal J. Updating quasi-Newton matrices with limited storage. Mathematics of computation. 1980; 35(151):773–782.

34. Early Treatment For ROP Cooperative Group. Revised indications for the treatment of retinopathy of prematurity; results of the early treatment for retinopathy of prematurity randomized trial. Arch Ophthalmol. 2003; 121:1684–1694. [PubMed: 14662586]

35. The Committee for the Classification of Retinopathy of Prematurity. The international classification of retinopathy of prematurity revisited. Arch Ophthalmol. 2005; 123:991–999. [PubMed: 16009843]

36. Chiang M, Jiang L, Gelman R, Du Y. Inter expert agreement of plus disease diagnosis in retinopathy of prematurity. Arch Ophthalmol. 2007; 125:875–880. [PubMed: 17620564]

37. Wallace DK, Quinn GE, Freedman SF, Chiang MF. Agreement among pediatric ophthalmologists in diagnosing plus and pre-plus disease in retinopathy of prematurity. Journal of American Association for Pediatric Ophthalmology and Strabismus. 2008; 12(4):352. [PubMed: 18329925]

38. Gelman R, Jiang L, Du Y, Martinez-Perez M, Flynn J, Chiang M. Plus disease in retinopathy of prematurity: pilot study of computer-based and expert diagnosis. Journal of American Association for Pediatric Ophthalmology and Strabismus. 2007; 11(6):532–540. [PubMed: 18029210]

39. Ataer-Cansizoglu E, You S, Kalpathy-Cramer J, Keck K, Chiang M, Erdogmus D. Observer and feature analysis on diagnosis of retinopathy of prematurity. IEEE International Workshop on Machine Learning for Signal Processing (MLSP). 2012:1–6.

highlights

- A novel manifold learning method which preserves pairwise distance order relations in the projected space
- Theoretical formulation of the constrained optimization problem extending classical MDS-based mean-squared error minimization
- A new performance metric that involves number of preserved distance orders
- Proposed method also provides the relation between distances in original and lower dimensional spaces

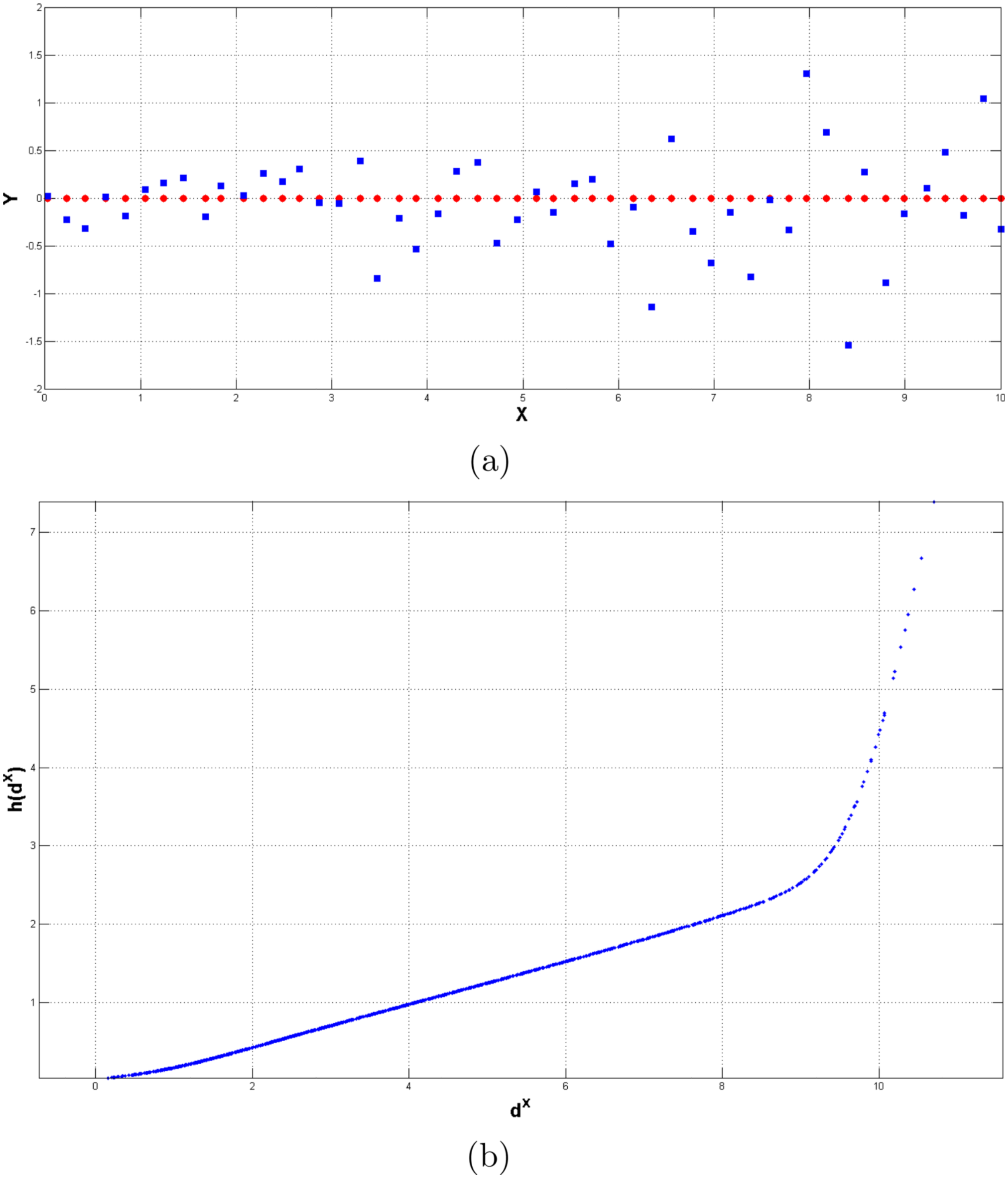
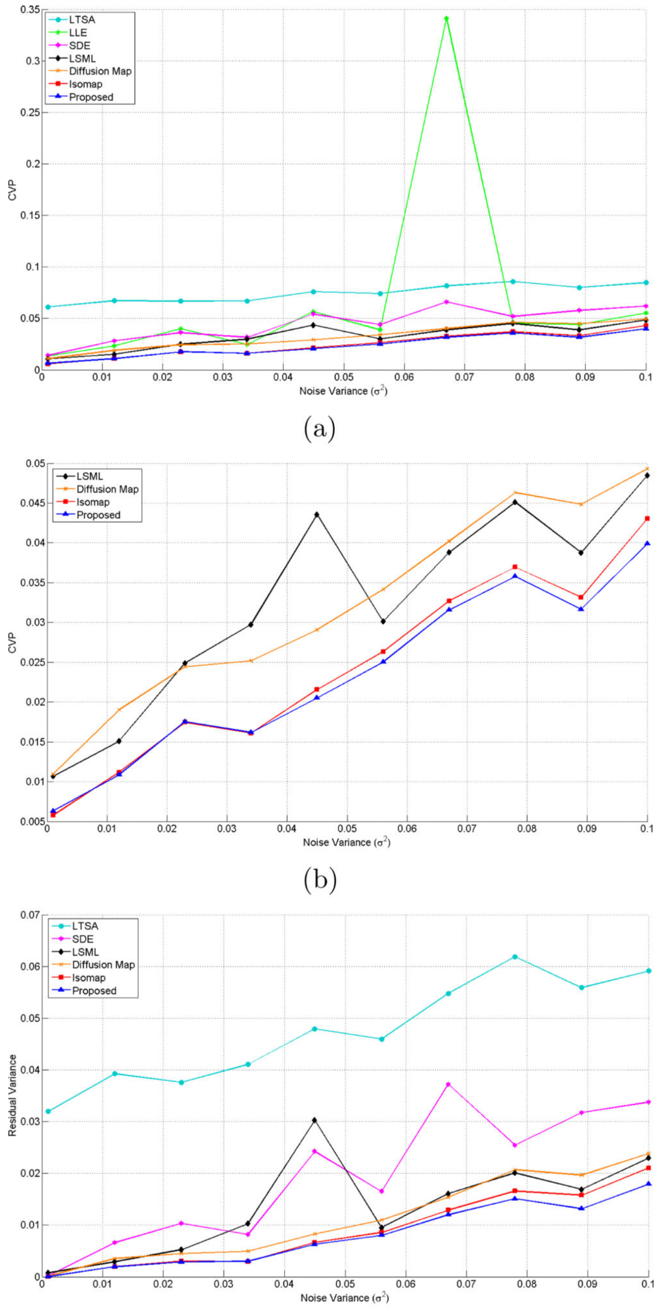


Figure 1.

(a) Growing band dataset for noise variance $\sigma^2 = 0.045$. The 2D data samples are indicated with blue squares. X coordinates of red dots illustrate the original 1D manifold. (b) Resulting $h(\cdot)$ function on GB dataset with $\sigma^2 = 0.1$.



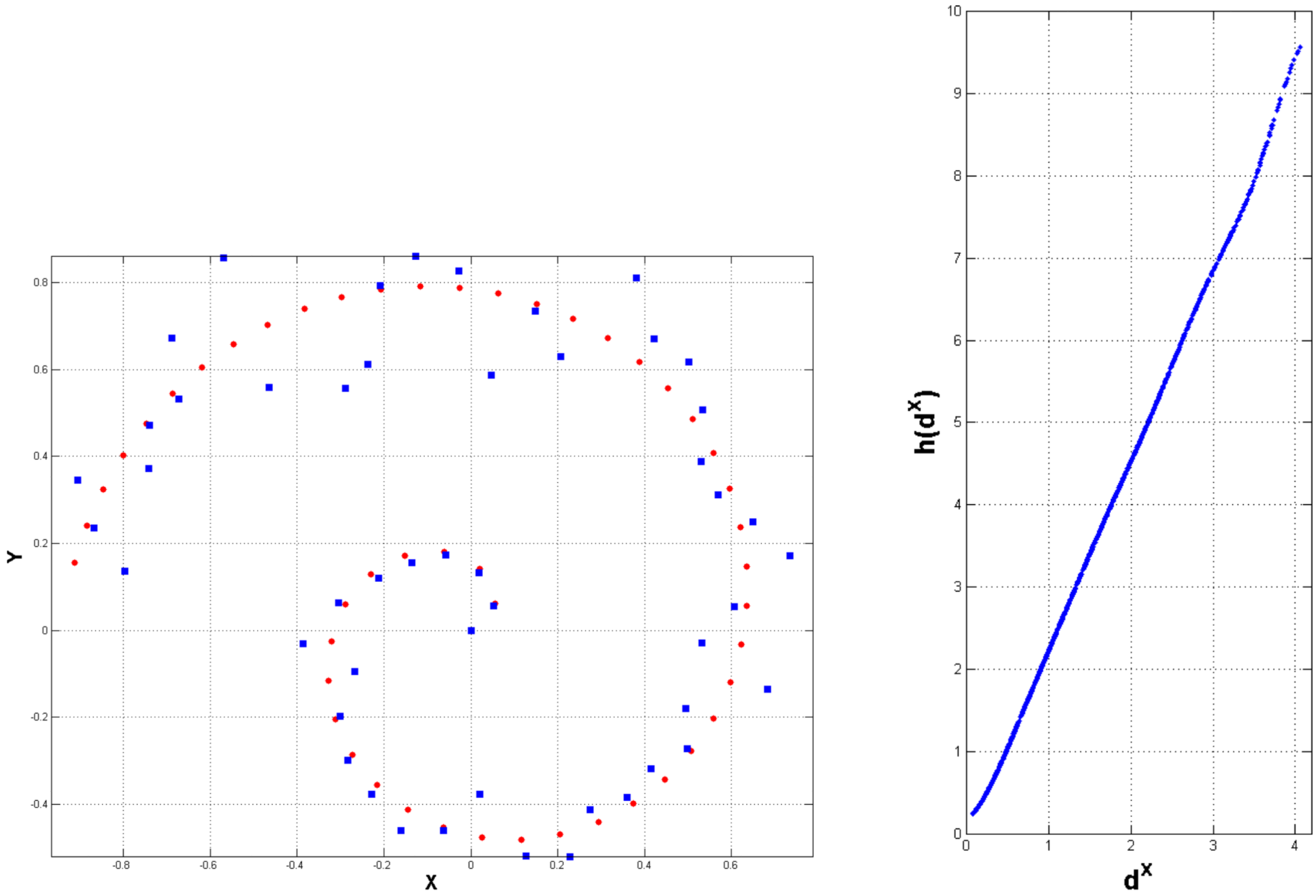
(a)

(b)

(c)

Figure 2.

Noise analysis on GB dataset: (a) CVP on datasets with different noise variances using LTSA [9], LLE [10], SDE [11], LSML [12, 13], DM [14], Isomap [7] and the proposed methods, (b) Zoomed in version of (a) with LSML [12, 13], DM [14], Isomap [7] and the proposed methods for better visualization, (c) Comparative residual variance results. Residual variance for LLE is omitted since it gives very high values.

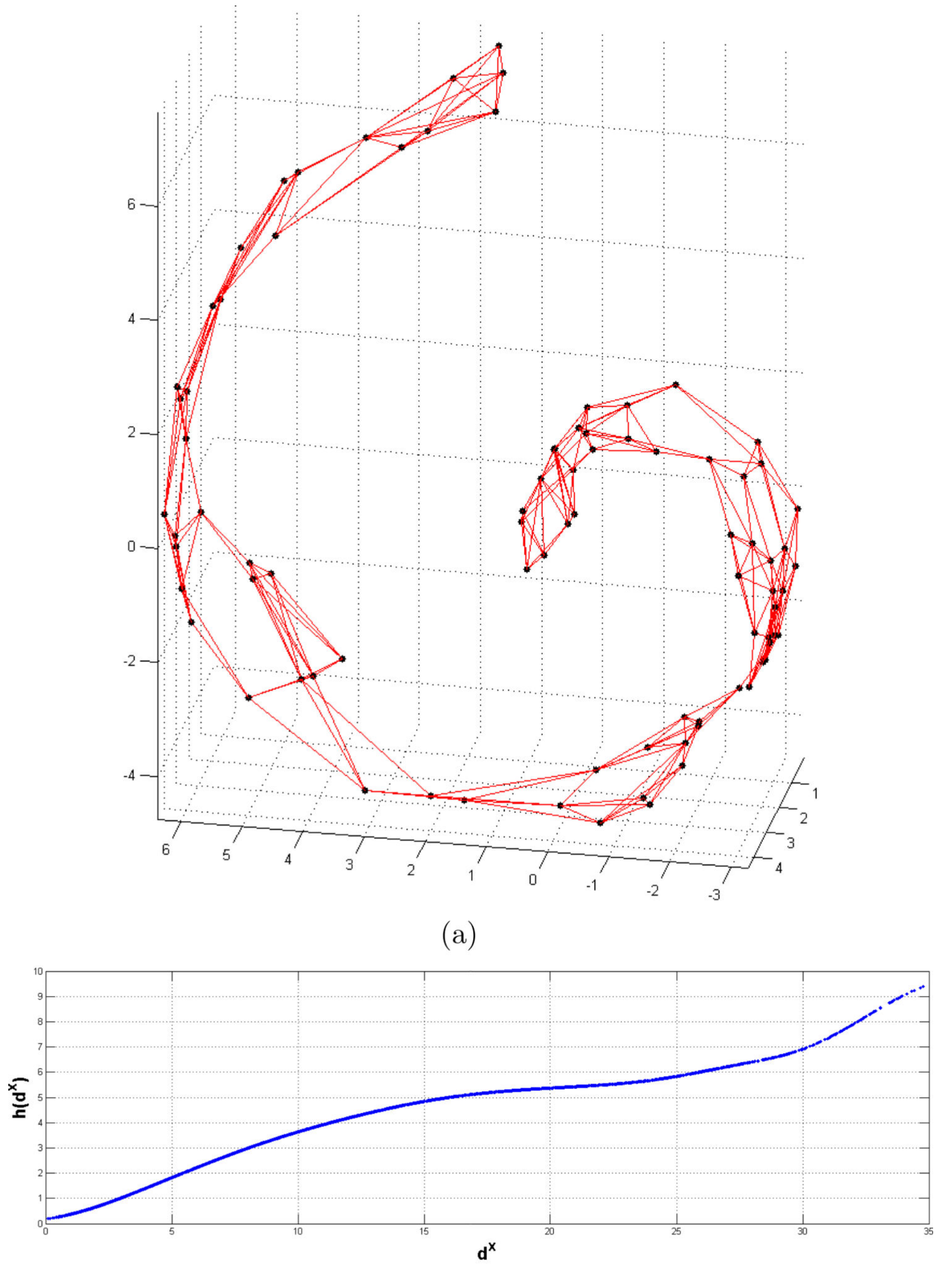


(a)

(b)

Figure 3.

Spiral dataset: (a) Noisy data samples are shown with blue squares, red dots indicate the original samples before adding noise, (b) resulting $h(\cdot)$ function.

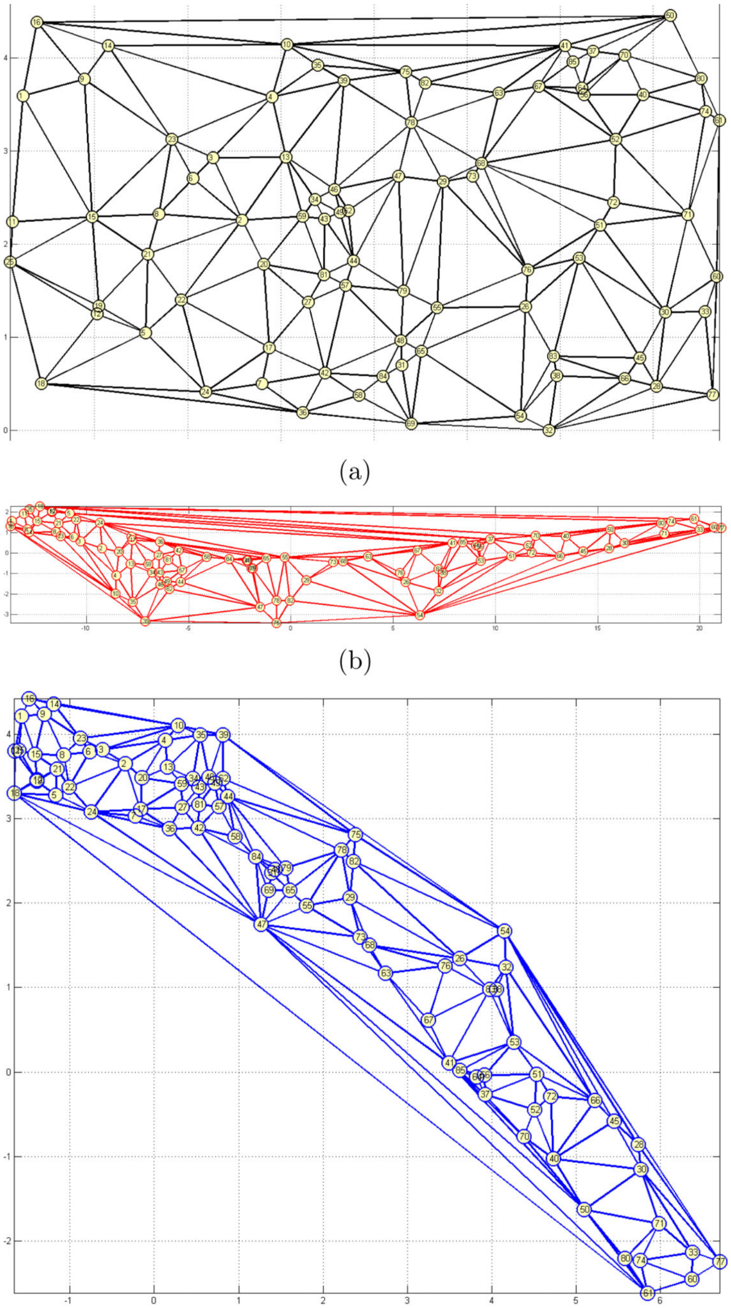


(a)

(b)

Figure 4.

Swiss roll dataset: (a) Data samples are shown with black dots overlayed with the knn-graph edges (red) where $k = 5$, (b) resulting $h(\cdot)$ function.



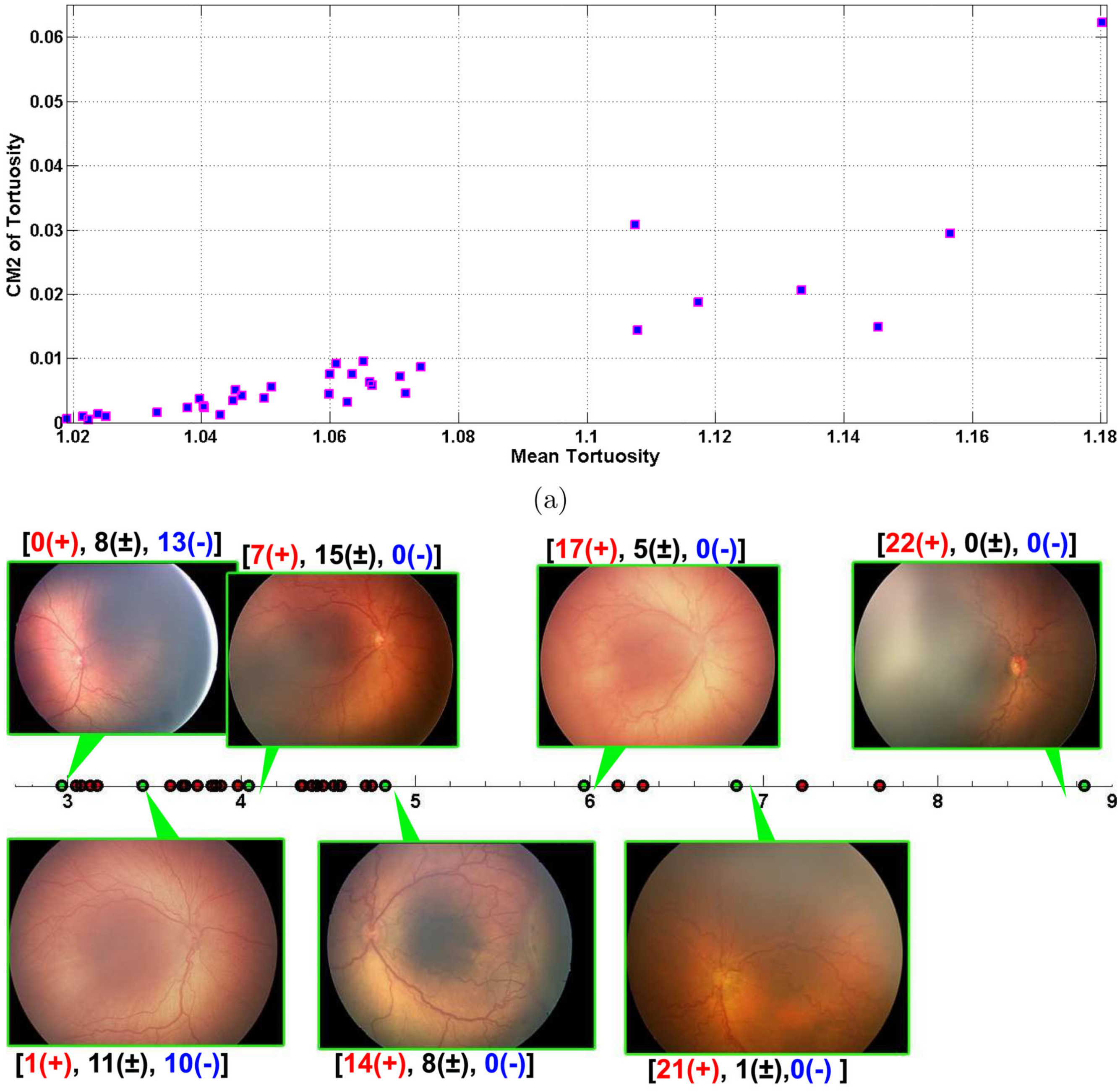
(a)

(b)

(c)

Figure 5.

2D representation of the Swiss roll dataset: (a) Original data in 2D, (b) Result with Isomap, and (c) Result with the proposed method.



(a)

(b)

Figure 6.

Experiment on retinal images. (a) scatter plot of the features: mean tortuosity and second central moment (CM2) of tortuosity, (b) result of the proposed method along with some example images indicated with green dots. Horizontal axis represents the projected points. The numbers above each image show the number of expert decisions *plus disease* (+), *pre-plus* (±) and *neither* (−) respectively. Notice that, amount of curvy vessels in images increases as we go from left to right. This correlates with the expert diagnostic decisions such that the number of *plus disease* decisions increases from left to right, while the number of *neither* decisions decreases. Moreover, the number of *pre-plus* decisions is high in the images in the middle of the manifold and lower for images at the left and rightmost ends.

Table 1

Performance analysis on GB dataset with different noise variances (σ^2) (top), on spiral dataset (middle) and on Swiss roll dataset (bottom).

	σ^2	**LTSA**	**LLE**	**SDE**	**LSML**	**DM**	**Isomap**	**Proposed**
# Violated Constraints	0.001	45856	10085	10500	7999	8204	4325	4738
	0.012	50324	17520	21111	11305	14250	8369	**8140**
	0.023	49974	29876	27048	18646	18306	**13077**	13154
	0.034	50137	18397	23712	22296	18881	**12074**	12126
	0.045	56904	42447	40638	32656	21800	16168	**15380**
	0.056	55594	29117	33005	22574	25582	19745	**18766**
	0.067	61210	255814	49481	29075	30130	24515	**23671**
	0.078	64284	34232	38870	33809	34714	27706	**26834**
	0.089	59952	32911	43207	29071	33610	24856	**23735**
	0.1	63429	41436	46387	36335	36975	32244	**29907**
Residual Variance	0.001	0.0320	0.9640	0.0002	0.0007	0.0001	**0.00003**	0.00004
	0.012	0.0392	0.9503	0.0066	0.0029	0.0035	0.0019	**0.0019**
	0.023	0.0376	0.9863	0.0103	0.0052	0.0045	0.0030	**0.0029**
	0.034	0.0411	0.9597	0.0082	0.0103	0.0049	**0.00297**	0.00299
	0.045	0.0479	0.9826	0.0242	0.0303	0.0082	0.0066	**0.0063**
	0.056	0.0460	0.9768	0.0165	0.0095	0.0110	0.0086	**0.0080**
	0.067	0.0548	1.0	0.0372	0.0161	0.0153	0.0129	**0.0120**
	0.078	0.0619	0.9611	0.0255	0.0201	0.0207	0.0166	**0.0151**
	0.089	0.0559	0.9626	0.0317	0.0169	0.0197	0.0158	**0.0132**
	0.1	0.0591	0.9838	0.0338	0.0230	0.0238	0.0210	**0.0179**

	LTSA	**LLE**	**SDE**	**LSML**	**DM**	**Isomap**	**Proposed**
#Violated Constraints	114430	210611	17138	16814	231680	8729	**8519**
CVP	0.1526	0.2809	0.0229	0.0224	0.3090	0.0116	**0.0114**
Residual Variance	0.2151	0.9063	0.0040	0.0049	0.78	0.0013	**0.0010**

	LTSA	**LLE**	**SDE**	**LSML**	**DM**	**Isomap**	**Proposed**
#Violated Constraints	1151299	440909	262700	183962	1281232	97043	**68838**
CVP	0.1807	0.0692	0.0412	0.0289	0.2011	0.0152	**0.0108**
Residual Variance	0.3646	0.0386	0.0136	0.0094	0.4917	0.0019	**0.0011**