Supervised Semi-Definite Embedding With Maximum Margin Objective

Abstract

In many applications data is represented in some high dimensional space; however, the important information often lie in some low-dimensional manifold. Finding the right data representation is key to the success of learning algorithms. Learning a low dimensional manifold is useful for data compression, interpretability, visualization and for improving machine learning performance and efficiency. Most existing manifold learning algorithms aim to find embeddings that either preserve the structure of the data while reducing its dimensionality, or to improve classification performance by separating the classes. In this paper, we introduce a manifold learning algorithm that performs both – it finds a low-dimensional manifold that preserves the structure of the data and where the classes are linearly separable. We achieve this by learning a kernel that maximizes the margin while unfolding the data through localy isometric variance maximization. The main result of the paper shows that the proposed approach (supervised Semidefinite embedding) reduces to a convex semidefinite program, that jointly identifies the kernel and the low dimensional embedding manifold. Experiments on synthetic and real data show that sSDE is able to obtain better classification performance compared to unsupervised manifold learning approaches and better data preservation than supervised methods and a reasonable trade-off between both objectives compared to competing alternatives.

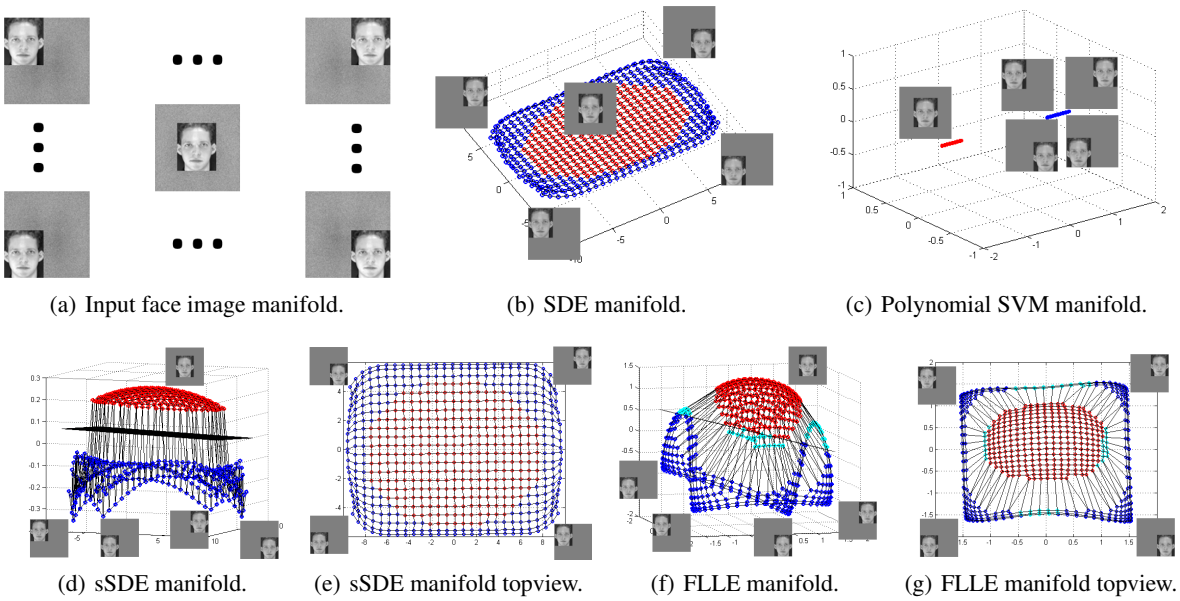
1. Introduction

Several data sets (e.g., images, video, sound, text, gene) are often characterized by high dimensions. However, most of these dimensions are unnecessary since the important information in these data typically lie in a low dimensional manifold. For example, the modes of variability in human faces (pose, expression), the degrees of freedom in rigid objects (rotation, translation), are often much smaller than the number of pixels in an image. Reducing the dimensionality to find this underlying manifold is an important component of learning algorithms. Low dimensional representations enable interpretability and visualization. Keeping only the relevant dimensions improves classification performance.

Manifold learning is an important research area in machine learning and is widely used for data compression, classification, and visualization. Many approaches have been proposed. Principal Component Analysis (PCA) (Jolliffe, 1986) is the earliest and the most popular linear manifold embedding algorithm. However, in many applications, a nonlinear mapping is needed; thus, Local Linear Embedding (LLE) (Roweis & Saul, 2000), Isomap (Tenenbaum et al., 2000), Laplacian Eigenmap (Belkin & Niyogi, 2001), Semidefinite Embedding (SDE) (Weinberger & Saul) have been proposed to find these low-dimensional nonlinear embeddings. These classical manifold learning algorithms are unsupervised – all aim to find a relatively low-dimensional space where the data representatives lie and the topology of the input data is preserved. It has been shown that they are useful on improving classification accuracy and efficiency in some cases (Turk & Pentland, 1991; Nilsson et al., 2004). However, since all these methods learn the manifolds unsupervised, the data representatives on the learned manifold are not guaranteed to be linearly separable. For example, given a set of face images as shown in Figure 1(a) with a classification task of identifying images that may be centered or not and their corresponding labels/annotations, the SDE algorithm unwraps the manifold as a 2D manifold shown in Figure 1(b). Because SDE does not utilize the labels, the positive (red) and negative (blue) samples are not linearly separable on the resulting manifold.

Supervised dimensionality reduction algorithms utilize the labels to directly find a low-dimensional space suitable for classification. Linear Discriminant Analysis (LDA) (Fisher, 1936) is a popular supervised dimensionality reduction algorithm that maps the input data into a $c-1$ dimensional space (where c is the number of classes) by minimizing the inter-class distances and maximizing the intra-class distances. One can also view the Support Vector Machine (SVM) classifier (Cortes & Vapnik, 1995) as a dimensionality reduction algorithm that maps the input

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.



(a) Input face image manifold. (b) SDE manifold. (c) Polynomial SVM manifold.

(d) sSDE manifold. (e) sSDE manifold topview. (f) FLLE manifold. (g) FLLE manifold topview.

Figure 1. Input face manifold and the learned manifold.

data to a space that maximizes the margin. To handle the nonlinearity, kernels are exploited to raise the feature space dimensionality so that the samples in the high dimensional kernel space become linearly separable. Examples include Kernel Discriminant Analysis (Mika et al.) and kernel SVM (Schlkopf & Burges, 1999). However, these supervised manifold learning approaches do not preserve the data structure. For example, a polynomial kernel SVM projects the face images onto a 1D manifold as shown in Figure 1(c). Note that the data structure is discarded on the learned manifold.

In this paper, our goal is to design a manifold learning algorithm that maps the input data to a low-dimensional manifold that preserves the structure of the data while rendering the classes in the new space linearly separable. We introduce an algorithm that preserves local isometry while maximizing the margin and unfolding the data via maximizing the variance. We introduce a new algorithm, termed supervised Semidefinite Embeddind (sSDE) that aims at obtaining linearly separable low dimensional classes that preserve the data structure. This is accomplished by unfolding the data via a locally isometric variance maximization. As shown in the paper, this approach leads to a convex optimization problem that can be efficiently solved using widely available software.

Related Work: In recent years, several algorithms have been proposed that combine unsupervised and supervised dimension reduction. One such class of algorithms is motivated by semi-supervised learning where the goal is to take advantage of the availability of a large number of unlabeled data and a few labeled data to learn the low-dimensional space for the classification task. Semi-supervisded Discriminant Analysis (SDA/SKDA)(Cai et al.) incorporates unlabeled information into LDA/KDA by adding a graph based regularization term. Belkin et al. (2006) provided a general framework on using the graph based regularization term to enable semi-supervised SVM learning. Another class of algorithms combine the goal of learning the manifold that separates the classes and preserves data characteristics. Local Discriminant Embedding (LDE)(Chen et al.), Marginal Fisher Analysis (MFA)(Yan et al., 2007), Fisher-LLE (FLLE)(De Ridder et al.) find a nonlinear low dimensional manifold that optimizes for within-class compactness and between-class separability. Note that the manifolds learned by these methods are not necessarily linearly separable as shown in Figure 1(f). Here we display the manifold learnt by FLLE with the errors shown in cyan. Further, since these methods do not preserve the interclass distances, the data structure along the boundary is not preserved either. On the other hand, as shown in Fig 1(f) and (e), the proposed algorithm finds a manifold where the classes are linearly separable, while preserving structure, even along the boundary. Finally, note that sSDE optimizes a max-margin, rather than a Fisher objective. Alternatively, methods such as SVDM (Pereira & Gordon) or (Rish et al., 2008), seek a compromise between classification and reconstruction errors. However, these methods are limited to learning linear subspaces and lead to challenging non-convex optimization problems. In contrast, sSDE leads to a tractable convex optimization (that can be solved in poly-

nomial time) and can learn non-linear manifolds

Contributions: In this paper, we propose an algorithm that nonlinearly maps the data to a low-dimensional manifold. The data in this new feature space is linearly separable, while its structure is preserved. The proposed approach offers the following advantages over existing methods: (1) It optimizes the nonlinear manifold objective and maximum margin objective together to achieve better classification performance than the unsupervised manifold learning approaches and smaller reconstruction error than supervised classification algorithms; (2) It solves a convex SDP problem so that the global optimum is guaranteed; (3) Our approach is able to learn a kernel, $\mathbf{K}$, for SVM; (4) The manifold dimensionality is given by the rank of $\mathbf{K}$, rather that as a result of trial and error; (5) It provides a scalable approximation where the manifold can be learned more efficiently; (6) It has an adjustable parameter that allows the flexibility to trade off classification errors versus dimensionality. The extreme values of the parameter range correspond to unsupervised SDE and supervised maximum margin classifiers, respectively; and finally, (7) The proposed algorithm allows for obtaining manifolds leading to better classification rates with a computational complexity similar to that of SDE.

1.1. Notation

For the sake of easy explaination of the algorithm details, we make following definitions.
$\mathbf{X}$: $\left[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\right]$ $\mathbf{X} \in R^{D\times N}$, where N is the number of samples and D is the number of features.
$\mathbf{Z}$: $\left[\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N\right]$ $\mathbf{Z} \in R^{d\times N}$, where N is the number of samples and d is the dimension of the manifold.
$\mathbf{y}$: $\left[y_1, y_2, \cdots, y_N\right]$ is the label of the data.

2. Problem Formulation

In this paper, a supervised Semi-Definite Embedding (sSDE) algorithm is proposed to obtain an embedding manifold from annotated data, preserving its structure while *simultaneously* allowing for linear classification with a high margin. The main goal is to find a low dimensional manifold embedding that captures the geometrical constraints among neighboring samples, regardless of their class, on which the annotated classes are linearly separable. To solve such a problem, we formulate an optimization objective function of the form

$$\boxed{\begin{array}{l}\min_{\mathbf{Z}} f(\mathbf{Z}) + \lambda g(\mathbf{Z}) \\ \text{s.t. Manifold Embedding Constraints} \\ \qquad \text{Data Annotation Constraints}\end{array}} \tag{1}$$

where $g(\mathbf{Z})$ is a manifold embedding cost that favors low dimensional embeddings preserving the local structure of the data and $f(\mathbf{Z})$ is a classification cost that penalizes miss-classifications using the low dimensional data. In the sequel, we describe our choices for $g(\mathbf{Z})$ and $f(\mathbf{Z})$ and show how the resulting joint convex objective cost can be efficiently optimized.

2.1. Embedding Cost: Semi-Definite Embedding (SDE)

For the embedding cost, $g(\mathbf{Z})$, we will use the SDE cost introduced in (Weinberger & Saul) to learn a low rank linear kernel $\mathbf{K} \doteq \mathbf{Z}^T\mathbf{Z}$ that preserves the local geometric constraints $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \approx \|\mathbf{z}_i - \mathbf{z}_j\|_2^2$ if and only if the i^{th} and j^{th} samples are neighbors. Defining $\mathbf{L} \doteq \mathbf{X}^T\mathbf{X}$ and $\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = \mathbf{L}_{ii} + \mathbf{L}_{jj} - 2\mathbf{L}_{ij}$, then the linear kernel matrix $\mathbf{K}$ of the low dimensional data representatives can be found by solving an SDP problem[1] to maximize their covariance:

$$\begin{aligned} \min_{k_{ij}} \ & -\mathsf{trace}(\mathbf{K}) \\ \text{s.t.} & (1-\epsilon)\delta_{ij} \leq \mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij} \leq (1+\epsilon)\delta_{ij} \\ & \mathbf{K} \succeq 0 \ , \ \sum_{i,j} \mathbf{K}_{ij} = 0 \end{aligned} \tag{2}$$

Once the above SDP problem is solved, the data representatives on the embedded manifold can be obtained from $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}}$, where $\mathbf{K} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$.

2.2. Classification Cost: Soft Margin SVM

While the embedded manifolds obtained using (2) have lower dimensionality and preserve the data structure better than the ones obtained using other approaches, in general, they do not help in achieving better classification accuracy (Weinberger & Saul). Thus, the need for the term $f(\mathbf{Z})$ in (1) for which we propose to use a classification cost based on a soft margin SVM to encourage classification boundaries with maximum margin and good performance in the presence of sparse outliers in the training data.

Recall that the soft margin SVM is formulated as follows:

$$\begin{array}{ll} \min_{\mathbf{w},\gamma,\xi} & C\sum \xi_i + \frac{1}{2}||\mathbf{w}||_2^2 \\ \text{s.t.} & y_i(\mathbf{z}_i^T\mathbf{w} - \gamma) + \xi_i - 1 \geq 0 \ , \ \xi_i \geq 0 \end{array} \tag{3}$$

where $\mathbf{w}$ is the weight vector, γ is the offset, and ξ is the error. Using the Lagrangian method, its dual problem is:

$$\begin{array}{ll} \max_{\mathbf{u}} & \sum \mathbf{u} - \frac{1}{2}\mathbf{u}^T\mathbf{D}\mathbf{K}\mathbf{D}\mathbf{u} \\ \text{s.t.} & 0 \leq u_i \leq C \ , \ \mathbf{y}^T\mathbf{u} = 0 \end{array} \tag{4}$$

where D is a diagonal matrix such that $d_{ii} = y_i$.

[1] Note that to allow for noise in the input, we added ϵ to the SDE constraint in our formulation 2.

2.3. Maximum Margin Manifold Learning Problem

Note that both, the SDE and the soft margin SVM problems can be expressed in terms of the Grammian matrix $\mathbf{K}$ rather than the data embedding itself $\mathbf{Z}$. Thus, the optimization problem (1) can be re-written as a maximization problem in terms of $\mathbf{K}$ when using

$$g(\mathbf{K}) = \mathrm{Tr}(\mathbf{K}) \tag{5}$$
$$f(\mathbf{K}) = \max_{\mathbf{u}} \mathbf{1}^T\mathbf{u} - \frac{1}{2}\mathbf{u}^T\mathbf{DKDu}$$
$$\text{s.t. } 0 \le u_i \le C \ , \ \mathbf{y}^T\mathbf{u} = 0$$

Furthermore, it is easy to see that although there is a polynomial term with variables $\mathbf{u}$ and $\mathbf{K}$, $f(\mathbf{K})$ is a convex function of $\mathbf{K}$ (Lanckriet et al., 2002): first, since $\mathbf{1}^T\mathbf{u} - \frac{1}{2}\mathbf{u}^T\mathbf{DKDu}$ is an affine function of $\mathbf{K}$ it is a convex function; second, note that $f(\mathbf{K})$ is the point-wise maximum of such convex functions and is thus convex. Moreover, the constraints in equations (2) and (4) are all linear. Finally, $g(\mathbf{K})$ is a linear function of K. Hence we can form a convex optimization problem with semidefinite constraints as follows:

$$\begin{array}{ll} \max_{\mathbf{K}} & f(\mathbf{K}) + \lambda g(\mathbf{K}) \\ \text{s.t.} & (1-\epsilon)\delta_{ij} \le \mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij} \le (1+\epsilon)\delta_{ij} \\ & \mathbf{K} \succeq 0 \ , \ \sum_{i,j}\mathbf{K}_{ij} = 0 \end{array} \tag{6}$$

The intuition behind the maximum margin manifold learning problem 6 is as follows. Imagine a net where the vertices of the net are the data samples and the links in the net are the neighborhood edges whose length is defined by the corresponding pairwise distances. In the original high dimensional space, this net may be rolled, twisted or wrinkled. The goal of the manifold embedding term, $g(\mathbf{K})$, is to unfold this net by maximizing the variance without breaking it (i.e., preserving the neighborhood distances) to find the low dimensional embedding. One can think of $g(\mathbf{K})$ as a force that unfolds the net; whereas, the maximum margin cost, $f(\mathbf{K})$, is a force that pulls the positive and negative samples apart. If the latter force is in the same direction as the data manifold, the final manifold remains the same as that of SDE; otherwise, the maximum margin cost will increase the dimensionality and fold the net in the direction of the boundary to separate the classes.

3. sSDE Problem and Solution

The convex optimization problem (6) is difficult to solve because we do not have a closed form for the term $f(\mathbf{K})$ and because this term is generally non-differentiable. In this section, we will show how to transform problem (6) into an SDP optimization problem with a linear objective function which can be solved using first order gradient or interior point methods. However, the computational cost of the resulting SDP problem increases quickly with the number of samples since it has $\mathcal{O}(N^2)$ variables and a $N \times N$ matrix constraint. To address this issue we will show below that i) it is possible to significantly reduce the number of free variables by approximating the samples using a small subset of "landmark" samples; and ii) it is possible to reduce the size of the matrix constraint by applying the Schur Complement Lemma. As a result, the resulting algorithm is capable of handling large datasets.

3.1. SDP Formulation

Problem (6) can be converted into an equivalent SDP problem by eliminating the variable $\mathbf{u}$ from $f(\mathbf{K})$ proceeding as in (Lanckriet et al., 2002). Let L be the Lagrangian of $f(\mathbf{K})$:

$$\begin{aligned} L(\mathbf{u}, \mathbf{v}_1, \mathbf{v}_2, v_3) = \ & \mathbf{1}^T\mathbf{u} - \tfrac{1}{2}\mathbf{u}^T\mathbf{DKDu} + \mathbf{v}_1^T\mathbf{u} \\ & + \mathbf{v}_2^T(C\mathbf{1} - \mathbf{u}) + v_3\mathbf{y}^T\mathbf{u} \end{aligned} \tag{7}$$

with multipliers $v_3 \in R$ and $\mathbf{v}_1, \mathbf{v}_2 \in R^N$. Then,

$$f(\mathbf{K}) = \min_{\mathbf{v}_1 \ge 0, \mathbf{v}_2 \ge 0, \mathbf{v}_3} \max_{\mathbf{u}} L(\mathbf{u}, \mathbf{v}_1, \mathbf{v}_2, v_3) \tag{8}$$

Setting the derivative of $L(\mathbf{u}, \mathbf{v}_1, \mathbf{v}_2, v_3)$ $w.r.t.$ $\mathbf{u}$ equal to zero at the optimum value $\mathbf{u}^*$, we have

$$\left.\frac{\partial L}{\partial \mathbf{u}}\right|_{\mathbf{u}^*} = \mathbf{DKDu}^* - (\mathbf{1} + \mathbf{v}_1 - \mathbf{v}_2 + v_3\mathbf{y}) = 0 \tag{9}$$

where $\mathbf{v} = \mathbf{1} + \mathbf{v}_1 - \mathbf{v}_2 + v_3\mathbf{y}$. Low dimensional manifolds require $\mathbf{K}$ to be rank deficient. Therefore, in this case, there are multiple optimal solutions for $\mathbf{u}^*$. However, since the objective function is convex, the corresponding optimal value is unique. Thus, we can use the optimal solution with minimum $||\mathbf{u}||_2$ given by: $\mathbf{u}^* = (\mathbf{DK}^\dagger\mathbf{D})\mathbf{v}$ where $\mathbf{K}^\dagger$ is the pseudo inverse of $\mathbf{K}$. Substituting $\mathbf{u}^*$ in (8), we have

$$f(\mathbf{K}) = \min_{\mathbf{v}_1, \mathbf{v}_2 \ge 0, v_3} \frac{1}{2}\mathbf{v}^T(\mathbf{DKD})^\dagger\mathbf{v} + C\mathbf{v}_2^T\mathbf{1} \tag{10}$$

Then, for any $t > 0$, the constraint $f(\mathbf{K}) \le t$ holds if and only if there exist $\mathbf{v}_1, \mathbf{v}_2 \ge 0$ and v_3 such that

$$\frac{1}{2}\mathbf{v}^T(\mathbf{DKD})^\dagger\mathbf{v} + C\mathbf{v}_2^T\mathbf{1} \le t$$

Finally, using Schur Complement Lemma, this is equivalent to the semidefinite constraint in $\mathbf{K}$ and $\mathbf{v}$

$$\begin{bmatrix} \mathbf{DKD} & \mathbf{v} \\ \mathbf{v}^T & 2t - 2C\mathbf{v}_2^T\mathbf{1} \end{bmatrix} \succeq 0 \tag{11}$$

Therefore $f(\mathbf{K})$ is equivalent to find the minimum t under the above constraint. Replacing in (6) $f(\mathbf{K})$ with t, changing the sign of $g(\mathbf{K})$ to make it a minimization problem,

and adding the above constraint on $\mathbf{K}$ and $\mathbf{v}$ results in the following SDP **sSDE problem**:

$$\begin{aligned} \min_{\mathbf{K},t,\mathbf{v}} \; & t - \lambda \operatorname{Tr}(\mathbf{K}) \\ \text{s.t.} \; & \begin{bmatrix} \mathbf{DKD} & \mathbf{v} \\ \mathbf{v}^T & 2t - 2C\mathbf{v}_2^T\mathbf{1} \end{bmatrix} \succeq 0 \\ & (1-\epsilon) \le \mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij} \le (1+\epsilon)\delta_{ij} \\ & \sum_{i,j} \mathbf{K}_{ij} = 0 \;\; , \;\; \mathbf{v}_1, \mathbf{v}_2 \ge 0 \end{aligned} \tag{12}$$

Remarks: The sSDE problem is a convex problem, and hence its solution (12) is the global optimum. It should also be noted that the constraint $\mathbf{K} \succeq 0$ was dropped, because (11) holds only if $\mathbf{DKD} \succeq 0$ and $\mathbf{DKD} \succeq 0$ is equivalent to $\mathbf{K} \succeq 0$, since $\mathbf{D}$ is a full rank diagonal matrix.

The significance of this result is that the cost function in (12) is a linear differentiable function and hence this problem can be solved using first order gradient methods or many off-the-shelf SDP solvers.

3.2. Approximation with Landmarks

While (12) is an SDP problem, it has $\mathcal{O}(N^2)$ variables and hence its computational and memory complexity grows quickly with the number of data points. To overcome this difficulty, following the approach proposed in (Weinberger et al., 2005), we will parameterize $\mathbf{K}$ using a much smaller matrix $N_l \times N_l$ matrix $\tilde{\mathbf{K}}$ such that $\mathbf{K} = \mathbf{Q}\tilde{\mathbf{K}}\mathbf{Q}^T$, where the fixed matrix $\mathbf{Q}$ captures the local geometry of the nonlinearity. Intuitively, the products $\mathbf{x}_i^T\mathbf{x}_j$ are approximated by combinations of products of a relatively small number N_l of "landmarks", encapsulated in $\mathbf{Q}$ and we assume that the same description applies *locally* to the products $\mathbf{z}_i^T\mathbf{z}_j$[2]. Then, as shown in (Weinberger et al., 2005), an optimal choice of $\mathbf{Q}$, in the sense that it minimizes the ℓ_2 norm of the reconstruction error, can be found by solving the following optimization problem while selecting for each measurement $\mathbf{x}_i$ up to k nearest neighbors:

$$\zeta(\mathbf{P}) = \sum_i ||\mathbf{x}_i - \sum_j P_{ij}\mathbf{x}_j|| \tag{13}$$

subject to $\sum_j P_{ij} = \mathbf{1}$ and $P_{ij} = 0$ if $\mathbf{x}_j$ is not a k nearest neighbor of $\mathbf{x}_i$. Finally, assuming, by reordering points if necessary, that the landmarks correspond to the first n_ℓ points of the set $\{\mathbf{x}_i\}$, the optimal $\mathbf{Q}$ is given by

$$\mathbf{Q} = \begin{bmatrix} \mathbf{I} \\ -(\mathbf{\Phi}_{22})^{-1}\mathbf{\Phi}_{21} \end{bmatrix} \tag{14}$$

where $\mathbf{\Phi}_{ij}$ denotes the blocks of $\mathbf{\Phi} \doteq (\mathbf{I}-\mathbf{P})^T(\mathbf{I}-\mathbf{P})$, partitioned so that its top left $n_\ell \times n_\ell$ block corresponds to the landmarks. Replacing $\mathbf{K}$ in terms of $\tilde{\mathbf{K}}$ in (12) leads to a optimization problem with substantially fewer variables. However, to make the problem always feasible, the local geometric constraints have to be relaxed (Weinberger et al., 2005) resulting in the following SDP problem:

$$\begin{aligned} \min_{\tilde{\mathbf{K}},t,\mathbf{v}} \; & t - \lambda \operatorname{Tr}(\mathbf{K}) \\ \text{s.t.} \; & \begin{bmatrix} \mathbf{DKD} & \mathbf{v} \\ \mathbf{v}^T & 2t - 2C\mathbf{v}_2^T\mathbf{1} \end{bmatrix} \succeq 0 \\ & \mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij} \le \delta_{ij} \\ & \textstyle\sum_{i,j} \mathbf{K}_{ij} = 0 \;\; , \;\; \mathbf{K} = \mathbf{Q}\tilde{\mathbf{K}}\mathbf{Q}^T \;\; , \mathbf{v}_1, \mathbf{v}_2 \ge 0 \end{aligned} \tag{15}$$

3.3. Efficiency Improvements

The computational complexity of problem (15) can be further reduced by taking advantage of the structure of the positive semidefinite matrix constraint (11) to decrease its size from $(N+1)\times(N+1)$ to a much smaller size of $(N_l+1)\times(N_l+1)$, as described below.

Start by applying the Schur Complement Lemma to (11) to obtain the following equivalent constraint:

$$\mathbf{DQ}\tilde{\mathbf{K}}\mathbf{Q}^T\mathbf{D} - \frac{1}{2t - 2C\mathbf{v}_2^T\mathbf{1}}\mathbf{v}\mathbf{v}^T \ge 0 \tag{16}$$

Since $\mathbf{Q}$ is a $N \times N_l$ matrix, with $N > N_l$, there exists a matrix $\mathbf{Q}_\perp$ such that $\mathbf{Q}_\perp^T\mathbf{Q} = \mathbf{0}$. Then, we have

$$\begin{aligned} & \mathbf{Q}_\perp^T\mathbf{D}\left[(\mathbf{DQ}\tilde{\mathbf{K}}\mathbf{Q}^T\mathbf{D} - \tfrac{1}{2t-2C\mathbf{v}_2^T\mathbf{1}}\mathbf{v}\mathbf{v}^T\right]\mathbf{DQ}_\perp \\ & = -\tfrac{1}{2t-2C\mathbf{v}_2^T\mathbf{1}}\mathbf{Q}_\perp^T\mathbf{D}\mathbf{v}\mathbf{v}^T\mathbf{DQ}_\perp \ge 0 \end{aligned} \tag{17}$$

Therefore, $\mathbf{Q}_\perp^T\mathbf{Dv} = 0$, and hence $\mathbf{Dv} = \mathbf{Q}\alpha$ for some $\alpha \in \mathbf{R}^{N_l}$. Since $\mathbf{D}^2 = I$, this implies that $\mathbf{v} = \mathbf{DQ}\alpha$. Thus, the matrix in constraint (11) can be decomposed as

$$\begin{bmatrix} \mathbf{DQ} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{K}} & \alpha \\ \alpha^T & 2t - 2C\mathbf{v}_2^T\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{DQ} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}^T \tag{18}$$

and (11) reduces to $\begin{bmatrix} \tilde{\mathbf{K}} & \alpha \\ \alpha^T & 2t - 2C\mathbf{v}_2^T\mathbf{1} \end{bmatrix} \succeq 0$. Then, the problem (15) is equivalent to the smaller **sSDE-l problem**:

$$\begin{aligned} \min_{\tilde{\mathbf{K}},t,\alpha,\mathbf{v}} \; & t - \lambda \operatorname{Tr}(\mathbf{K}) \\ \text{s.t.} \; & \begin{bmatrix} \tilde{\mathbf{K}} & \alpha \\ \alpha^T & 2t - 2C\mathbf{v}_2^T\mathbf{1} \end{bmatrix} \succeq 0 \\ & \mathbf{v} = \mathbf{DQ}\alpha \\ & \mathbf{K}_{ii} + \mathbf{K}_{jj} - 2\mathbf{K}_{ij} \le \delta_{ij} \\ & \sum_{i,j} \mathbf{K}_{ij} = 0 \, , \; \mathbf{K} = \mathbf{Q}\tilde{\mathbf{K}}\mathbf{Q}^T \, , \; \mathbf{v}_1, \mathbf{v}_2 \ge 0 \end{aligned} \tag{19}$$

[2] This is equivalent to assuming that $\Pi^{-1}(.)$ is locally linearizable.

The number of geometric inequality constraints is linear with kN. It is possible to further reduce the problem size, by starting with a small subset of these constraints and then iteratively adding violated inequality constraints to the problem as in (Weinberger et al., 2005). By using this approach, our experiments show that the problem can be solved with less than half of the geometric inequality constraints. Furthermore, although this technique requires that we solve the problem multiple times (usually 10), the overall time is much less than the time required to solve the original problem. (See Figure 3).

3.4. Out of Sample Extension

After embedding, a new sample can be mapped quickly by first obtaining the weights needed to reconstruct the sample from its k-nearest neighbors in the training set, as it is done in LLE. Then, its corresponding embedding is obtained by simply using these weights to combine the embeddings of its neighbors.

4. Experiments

In this section we give implementation details and describe a set of experiments to evaluate the proposed algorithms using synthetic and real datasets. As a benchmark, we compared the performance against the ones achieved when using the SDE, SVDM and FLLE algorithms.

4.1. Implementation Details

We used the CVX toolbox with the SEDUMI solver[3] to solve the SDP problems in sSDE and SDE.

In all the experiments, the samples in the datasets were evenly randomly divided into 6 folds. One of these folds was selected for independent validation of the parameters. Each of the remaining 5 folds were used for testing while using the other 4 folds for training.

The parameters validated for each algorithm are listed on the first row of table 1. All the algorithms, except SVDM, require finding the k-nearest neighbors for the data and use a soft margin linear SVM for classification. The parameter k was validated within $[4, 7]$. The parameter C for the SVM was validated within $[1, 16384]$. For the sSDE and sSDE-l algorithms, λ was validated within $[10^{-4}, 10^{3}]$ and ϵ was set to 0.1. The parameter α for FLLE was validated between $[0, 1]$. The dimensionality d of the learned manifolds was determined as follows. For SDE, sSDE-l and sSDE we thresholded the eigenvalues of the kernel matrix $\mathbf{K}$, λ_i using the expression:

$$d = \underset{m}{\operatorname{argmax}} \left(\frac{\sum_1^m \lambda_i}{\sum \lambda_i} \leq T_1 \text{ or } \frac{\lambda_m}{\sum \lambda_i} \geq T_2\right) \qquad (20)$$

with $T_1 = 0.985$ and $T_2 = 0.015$, respectively. For FLLE, d was determined by thresholding the eigenvalues of the local variance matrix using a threshold value $T = 0.95$. For SVDM, d is an input parameter and was determined using cross-validation. Because, in general, the obtained dimensionality was different for each of these algorithms, we also compared the algorithms using a common dimensionality d set to the value obtained for sSDE-l. We label these experiments using the subscript d to distinguish them from the original ones.

As described in section 3.3, iteratively adding the geometric constraints in the sSDE-l algorithm helps reducing its running time when working with large datasets. On one hand, the smaller the number of constraints added at an iteration, the faster that iteration would be. On the other hand, adding too few constraints at a time would increase the number of iterations needed to converge. In all of our experiments with the sSDE-l algorithm, we added at each iteration the 200 most violated geometric constrains.

Finally, sSDE, sSDE-l, SDE, and FLLE use the same method for out of sample extension. For SVDM, since its linear projection matrix is known, we computed the embedded data by solving a least square problem.

4.2. Data Sets

Swiss-Roll: 960 uniformly random distributed 2D samples were wrapped as a 3D swissroll manifold. A circle boundary defined on the 2D manifold divides the samples into positive and negative classes.

Face Grid: As shown in Figure 1(b), 594 images were generated by moving a 112×92 face image template on a 200×200 background image with a step of 4 pixels in each direction. The images in which the centroid of the face template is within a radius of 32 pixels from the center were labeled as positive samples.

Table Top Dataset: It consists of silhouette images of staplers, mugs and mice. There are totally 160 silhouette images of each type, of 10 different objects under 16 viewpoints (8 view angles and 2 different heights). All the images were cropped with the tightest bounding box and re-scaled to 100×50 pixels. The labels were given using the height of the viewpoints. Since this dataset cannot be evenly divided into 6 folders, we randomly selected 30 samples for the validation set and then evenly divided the remaining 130 images into 5 folds for evaluation.

PAL Face Dataset: There are two classes and 90 face images in each class. The class label was decided based on

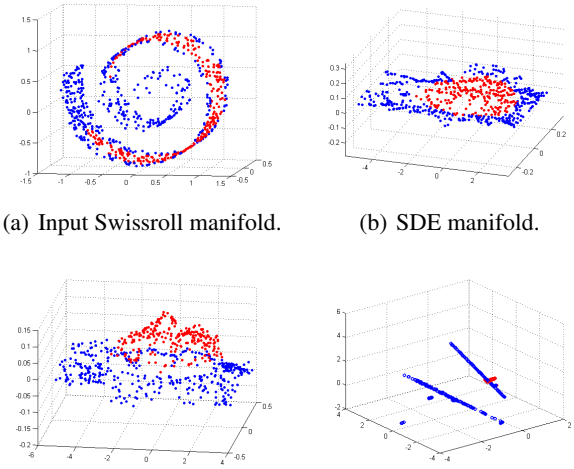
[3] Using this solver, the computation complexity of solving a LMI problem is $\mathcal{O}(n^2 m^{2.5} + m^{3.5})$, where n is the number of variables and m is the number of constraints.

the gender of the people. The images in each class were evenly divided into 6 folds. A histogram equalization preprocessing was applied to this dataset.

Ionosphere, Sonar: These datasets are from the UCI repository. Since these datasets cannot be evenly divided into 6 folds, we randomly selected 61 and 33 samples in each dataset as the validation set and evenly divided the remaining samples into 5 folds for evaluation.

All the datasets were normalized such that $\sum_i \mathbf{x}_i = \mathbf{0}$ and $\sum_i \mathbf{x}_i^T \mathbf{x}_i = N$.

4.3. Classification Performance Analysis



(a) Input Swissroll manifold. (b) SDE manifold.

(c) sSDE manifold. (d) FLLE manifold, $\alpha = 0.8$.

Figure 2. Input Swissroll manifold and the learned manifolds.

Figure 2 shows the results of applying the proposed sSDE, the SDE, and the FLLE algorithms to the 3D swissroll dataset shown in Figure 2(a). Although the dimensionality of the sSDE manifold is higher than the SDE's, the sSDE data representatives became linearly separable due to the influence of the annotation prior to decrease the maximum margin objective $f(\mathbf{K})$. On the other hand, the data representatives on the unsupervised SDE's manifold embedding remained not linearly separable. Figure 2(d) shows that the data structure was destroyed on the FLLE's manifold as this method penalized the inter-class distance. Finally, as shown in Table 1 the proposed algorithm had the best classification performance.

Next, we report the effects of using a subset of landmarks samples on classification performance and computation time. This was studied by running a set of experiments using the same training and testing data and the same parameters validated used in the previous experiment, but using randomly selected N_l landmarks. Each of these experiments was repeated 8 times so that the landmark set was different but with the same number of samples. As shown in Figure 3, the mean mis-classification rate decreases and the computation time increases as the number of landmarks N_l increases. Furthermore, when the number of landmarks exceeds 35% of the data, solving the smaller problem (19) achieves the same classification accuracy as solving the full problem (12). This suggests that using $25\% - 40\%$ of the data as landmarks is a reasonable trade-off between classification accuracy and computational cost. Thus, we always used 30% of the samples as the landmarks for all the other experiments reported next. It should be noted that using landmarks to reduce the problem is not the same as simply sub-sampling the manifold. This is because the non-landmark samples are approximated with landmarks and still contribute to the classification and the local geometric constraints. To show the difference, we also solved the problem (12) using only the landmarks and show its mis-classification rate as blue dotted line in Figure 3. From these plots, we can see that under the same conditions, sSDE learns a manifold where a linear SVM can classify the samples more accurately.

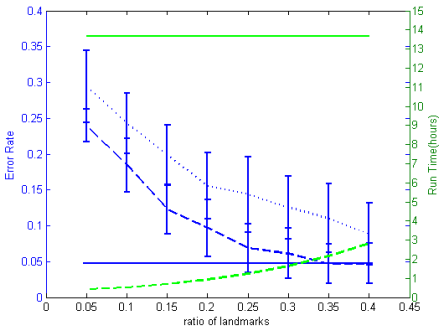


Figure 3. Mis-classification rates and run time are shown in blue and green lines, respectively. Plots for (12) and (19) are shown using solid and dash lines, respectively. The blue dotted line shows the mis-classification rate for (12) using landmarks alone.

Finally, the classification accuracy performance for all the algorithms on all the datasets is summarized in Table 1. As seen there, the proposed algorithm performs better than the other algorithms when there is a hidden "wrapped" manifold in the original data set, such as is the case when the view point changes. On generic machine learning datasets, where the manifolds are already "flat", then our algorithm shows similar classification accuracy as the others.

4.4. Reconstruction Performance

In this experiment, we evaluated how well the learned manifold preserves the original data structure. To this effect, we used as a metric the *local linear weight reconstruction*

Table 1. Mis-Classification Rate(%) $MEAN \pm STD$

	SDE k, C	SDE$_d$ k, C	SVDM D, d	SVDM$_d$ D	FLLE k, α, C	FLLE$_d$ k, α, C	FSDE k, C, λ	sSDE-l k, C, λ
Swissroll	26.0 ± 10.0	20.6 ± 8.3	35.1 ± 3.1	36.4 ± 1.9	12.1 ± 1.9	10.6 ± 1.8	3.9 ± 2.5	4.9 ± 1.2
face grid	10.1 ± 5.4	11.7 ± 5.3	17.2 ± 4.6	24.4 ± 5.4	6.9 ± 3.7	14.3 ± 3.6	5.7 ± 2.7	5.9 ± 2.2
mug	43.8 ± 12.9	26.9 ± 13	23.1 ± 5.4	34.6 ± 11.2	21.5 ± 14	24.6 ± 12.3	15.4 ± 7.7	18.5 ± 3.2
staple	43.1 ± 11.7	37.7 ± 12.3	36.9 ± 4.4	49.2 ± 14.2	31.5 ± 3.2	30.8 ± 11.5	26.2 ± 5.7	25.4 ± 7.5
mouse	50.8 ± 8.8	43.8 ± 11.1	47.7 ± 6.4	45.4 ± 7.4	36.9 ± 4.4	34.6 ± 2.7	33.1 ± 8.9	30.8 ± 4.7
Ionosphere	28.6 ± 7.9	15.2 ± 2.6	26.9 ± 11.1	24.8 ± 11.5	17.9 ± 6.5	14.8 ± 5.5	16.2 ± 4.7	13.4 ± 3.1
PAL	21.3 ± 5.1	18.7 ± 5.1	20.7 ± 4.9	23.3 ± 4.1	20 ± 4.1	18.7 ± 3.8	18 ± 5.6	18.7 ± 3.8
sonar	32.6 ± 6.6	38.9 ± 9.4	41.1 ± 5.9	36.6 ± 9.6	19.4 ± 5.5	20.6 ± 14.5	19.4 ± 10.6	18.9 ± 8.7

Table 2. Local Linear Weight Reconstruction Error. The smallest error is marked with red in each row.

	SDE	SDE$_d$	SVDM	SVDM$_d$	FLLE	FLLE$_d$	sSDE	sSDE-l
Swissroll	0.0003	0.0003	0.2535	0.2433	0.1456	0.0435	0.0005	0.0359
face	0.0584	0.0797	0.0742	0.0859	1.8464	0.0622	0.0452	0.0772
mug	0.7273	14.8759	0.9475	3.3948	4.2587	5.1829	0.6001	7.8384
stapler	0.7796	6.1433	0.5695	3.1467	3.1743	1.4801	0.4889	6.3756
mouse	0.6146	4.9941	0.5658	6.3078	5.3863	6.2212	0.5980	6.9540
Ionosphere	0.7793	1.3606	0.5216	0.8166	6.0691	7.5529	0.6983	3.1642
PAL	0.7887	6.7462	0.6829	1.0719	14.4109	7.4636	0.7656	3.7395
sonar	0.3511	2.6845	0.4421	0.8644	4.3626	3.5120	0.3416	2.0427

error, defined as $e_i = ||\mathbf{x}_i - \sum P_{ij}\mathbf{x}_j||_2^2$ where P_ij are the local linear weights that minimize $||\mathbf{z}_i - \sum P_{ij}\mathbf{z}_j||_2^2$, where $P_{ij} = 0$ if $\mathbf{z}_j$ is not a neighbor of $\mathbf{z}_i$. The mean of e_i for each of the algorithms is shown in Table 2. SDE and sSDE perform better in most of the datasets. For SVDM, although the overall reconstruction error is part of its non-convex objective, e_i based on the local neighbors is larger than for the SDE based methods due to the manifold non-linearity and the sub-optimal solution.

5. Conclusions

We developed a new manifold learning algorithm, sSDE, that learns an embedding that preserves local structure of the data while simultaneously allowing for linear classification with a high margin. sSDE achieves this by learning a kernel that unfolds the data by maximizing variance and separates the classes by maximizing the margin, subject to the local geometric constraints. We formulated the problem as an SDP problem, which can be solved by off-the-shelf solvers, and has a guaranteed global optimum. We also introduced an approximation of sSDE using landmark samples, enabling our algorithm to be scalable for large datasets. sSDE preserves data structure as well as SDE, yet obtains a manifold that enables better classification without increasing the computational complexity. Furthermore, if a linear class boundary exist on the SDE manifold, sSDE finds the SDE manifold; when such boundary does not exist, sSDE folds the unfolded SDE manifold in the direction of the boundary, leading to better classification performance. On the other end, sSDE performs as well as SVM for classification, with the advantage of being able to learn the kernel and at the same time preserve the data structure for data representation and interpretability. Finally, experimental results on synthetic and real data show that sSDE is able to provide a reasonable trade-off between classification performance and preserving data structure compared to competing methods.

References

Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Adv. in neural inform. proc. sys.*, 14:585–591, 2001.

Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The J. of Machine Learning Research*, 7:2399–2434, 2006.

Cai, D., He, X., and Han, J. Semi-supervised discriminant analysis. In *Computer Vision, 2007. ICCV 2007. IEEE 11th Int. Conf. on*, pp. 1–7. IEEE.

Chen, H.T., Chang, H.W., and Liu, T.L. Local discriminant embedding and its variants. In *Computer Vision and Pattern Recognition, 2005.* , volume 2, pp. 846–853. IEEE.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

De Ridder, D., Loog, M., and Reinders, M. JT. Local fisher embedding. In *Pattern Recognition, 2004. ICPR 2004.* , volume 2, pp. 295–298. IEEE.

Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

Jolliffe, I. T. *Principal component analysis*, volume 487. Springer-Verlag New York, 1986.

Lanckriet, G., Cristianini, N., Bartlett, P., and Ghaoui, L. El. Learning the kernel matrix with semi-definite programming. *J. of Machine Learning Research*, 5:2004.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, KR. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999.*, pp. 41–48. IEEE.

Nilsson, J., Fioretos, T., Hglund, M., and Fontes, M. Approximate geodesic distances reveal biologically relevant structures in microarray data. *Bioinformatics*, pp. 874–880, 2004.

Pereira, F. and Gordon, G. The support vector decomposition machine. In *Proc. of the 23rd Int. Conf. on Machine learning*, pp. 689–696. ACM.

Rish, I., Grabarnik, G., Cecchi, G., Pereira, F., and Gordon, G. J. Closed-form supervised dimensionality reduction with generalized linear models. In *Proc. of the 25th Int. Conf. on Machine learning*, pp. 832–839. ACM, 2008.

Roweis, S. T. and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290 (5500):2323–2326, 2000.

Schlkopf, B. and Burges, C. JC. *Advances in kernel methods: support vector learning*. The MIT press, 1999.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–+, 2000.

Turk, M.A. and Pentland, A.P. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition.*, pp. 586–591, 1991.

Weinberger, K. Q. and Saul, L. K. Unsupervised learning of image manifolds by semidefinite programming. In *IEEE Conf. CVPR*, volume 2, pp. 988–995. IEEE.

Weinberger, K. Q, Packer, B. D, and Saul, L. K. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *Proc. of the tenth int. workshop on artificial intelligence and statistics*, pp. 381–388, 2005.

Yan, S., Xu, D., Zhang, Z., Zhang, H.J., Yang, Qi., and Lin, S. Graph embedding and extensions: a general framework for dimensionality reduction. *PAMI, IEEE Transactions on*, 29(1):40–51, 2007. IEEE.