SUPERVISED SEMI-DEFINITE EMBEDDING FOR IMAGE MANIFOLDS

Benyu Zhang [1], Jun Yan [2], Ning Liu [3], Qiansheng Cheng [2], Zheng Chen[1], Wei-Ying Ma[1]

[1] Microsoft Research Asia, No. 49 Zhichun Road, Beijing, P.R. China
{byzhang, zhengc, wyma}@microsoft.com
[2] LMAM, Department of Information Science, School of Mathematical Science, Peking University, Beijing, P.R. China
{yanjun, qcheng}@math.pku.edu.cn
[3] Department of Mathematical Science, Tsinghua University, Beijing, P.R. China
liun01@mails.tsinghua.edu.cn

ABSTRACT

Semi-definite Embedding (SDE) has been a recently proposed to maximize the sum of pair wise squared distances between outputs while the input data and outputs are locally isometric, i.e. it pulls the outputs as far apart as possible, subject to unfolding a manifold without any furling or fold for unsupervised nonlinear dimensionality reduction. The extensions of SDE to supervised feature extraction, named as Supervised Semi-definite Embedding (SSDE) was proposed by the authors of this paper. Here, the method is unified in a mathematical framework and applied to a number of benchmark data sets. Results show that SSDE performs very well on high-dimensional data which exhibits a manifold structure.

1. INTRODUCTION

In many pattern recognition problems, sensors provide a large amount of features, such as in images and sound signals. Traditional methods to perform dimensionality reduction are mainly linear, and include selection of subsets of features and linear mappings to lower-dimensional spaces [2].Over the years, a number of techniques have been proposed to perform nonlinear mappings, such as such as Local Linear Embedding (LLE) [9], Isomap [12], Laplacian Eigenmap [7] and mixtures of linear models [8].

Recently, a conceptually simple yet powerful method for nonlinear mapping has been proposed by Weinberger and Saul [5, 6, 13, 14]: Semi-definite Embedding (SDE). SDE is based fundamentally on the notion of *isometry*. Like Isomap and LLE, it relies on efficient and tractable optimization that is not plagued by spurious local minima. Isomap estimates geodesic distances between inputs; LLE estimates the coefficients of local linear reconstructions; SDE estimates local angles and distances. Comparing the algorithms, the theoretical and experimental results of SDE showed that it overcomes certain limitation of previous works interestingly. However, the original SDE algorithm is unsupervised and was originally intended for multidimensional image data visualization. The unsupervised algorithms ignore the valuable label information used in classification problems. Our contribution in this paper is a supervised variant of SDE for image data. To learn a mapping from a high-dimensional space into low-dimensional space, a nonlinear regression analysis [10] is used, i.e. mapping new arrival data points into the embedding space, where the classification of these points is done by a classifier, such as K Nearest Neighbor (KNN) [1]. Our experimental results on various benchmark data sets show that this algorithm is effective for classification problems in contrast to SLLE, and SDE. There is an interesting observation: SSDE could visualize the information of the different classes of data in the reduced space.

The rest of this paper is organized as follows. In Section 2, we introduce some necessary background knowledge of SDE. In Section 3, we present our proposed Supervised SDE algorithm mathematically. In Section 4, we demonstrate the experimental results on some benchmark data sets. Conclusion of this paper is given in Section 5.

2. BACKGROUND

In this paper, our main contribution is to propose a supervised nonlinear dimension reduction (manifold learning) algorithm for classification problems. Our algorithm originated from an unsupervised manifold learning algorithm called as Semi-definite Programming. Some necessary basic ideas of unsupervised SDE are shown in this section.

0-7803-9332-5/05/$20.00 ©2005 IEEE

2.1. Isometry

As a beginning, we give the definition of "neighbor" to describe what local information is. Consider two data sets $X \in R^{N\times D}$ and $Y \in R^{N\times d}$ that are in one-to-one correspondence $X \to Y$, here $X \in R^{N\times D}$ is a matrix with each line a data point. Let matrix $\Gamma_x = (\tau_{ij}) \in R^{N\times N}$ and $\Gamma_y \in R^{N\times N}$ indicate a neighborhood relation matrix (adjacent matrix) on a data matrix *X* and *Y* respectively, in other words, we regard X_j as a neighbor of X_i if and only if $\tau_{ij} = 1$ (the same to Y_j and Y_i). Then we can say that X and Y are locally isometric if and only if X_i and X_j are themselves neighbors (that is, $\Gamma_x = \Gamma_y$), we have:

$$\left|Y_i - Y_j\right|^2 = \left|X_i - X_j\right|^2 \qquad (\mathbf{1})$$

Let $G_{ij} = X_i \cdot X_j$ and $K_{ij} = Y_i \cdot Y_j$ denote the Gram matrices of the inputs and outputs, respectively. Then we can rewrite eq. (1) as by simple linear algebra transformations:

$$K_{ii} + K_{jj} - K_{ij} - K_{ji} = G_{ii} + G_{jj} - G_{ij} - G_{ji} \qquad (\mathbf{2})$$

Eq. (2) expresses the conditions for local isometry purely in terms of Gram matrices; it is in fact this formulation that will form the basis of SDE algorithm for manifold learning that we will introduce in the next section.

2.2. Semi-definite Embedding

The recently proposed SDE algorithm is proposed to maximize the sum of pair wise squared distances between outputs while the input data and outputs are locally isometric, i.e. it pulls the outputs as far apart as possible, subject to unfolding a manifold without any furling or fold. Mathematically, SDE obtains:

$$\max \mathrm{D(Y)} = \sum_{i,j} \left|Y_i - Y_j\right|^2 \qquad (\mathbf{3})$$

In addition, SDE also constrain the outputs Y_i to be centered on the origin:

$$\sum_i Y_i = 0$$

Then, Eq. (3) can be translated into the following form by adding the constraint:

$$\mathrm{D(Y)} = \sum_{i,j} \left|Y_i - Y_j\right|^2 = \sum_i \left|Y_i\right|^2 = Tr(K) \qquad (\mathbf{4})$$

So the problem of SDE is to maximize the variance of the outputs $Y_i \in R^d$ subject to the constraints that they are centered on the origin and locally isometric to the inputs $X_i \in R^D$. The optimization problem can be written as an instance of semi-definite programming problem below:

$$\begin{aligned} &Max\ Tr(K) \text{ subject to } \mathrm{K} \geq 0, \sum_{\mathrm{ij}} \mathrm{K} = 0, \\ &\quad \text{and } \forall ij \text{ such that } \tau_{ij} = 1: \\ &K_{ii} + K_{jj} - K_{ij} - K_{ji} = G_{ii} + G_{jj} - G_{ij} - G_{ji} \end{aligned} \qquad (\mathbf{5})$$

The problem we discussed above is an illustration of semi-definite programming (SDP). There are a large amount of papers focusing on efficiently solving the SDP problem, as well as a number of general-purpose toolboxes. The experimental results in this paper were obtained using the SeDuMi and CSDP 4.7 toolbox [3, 11] to solve the semi-definite programming in a supervised manner.

After compute the Gram matrix K by semi-definite programming, we can regain the outputs $Y_i \in R^d$. That $V_{\alpha i}$ denotes the i^{th} element of the α^{th} eigenvector, with eigenvalue λ_α. Then the Gram matrix can be written as:

$$K_{ij} = \sum_{\alpha=1}^{N} \lambda_\alpha V_{\alpha i} V_{\alpha j}$$

A d-dimensional embedding that is locally isometric to the inputs $X_i \in R^D$ is obtained by identifying the α^{th} element of the output Y_i as:

$$Y_{\alpha i} = \sqrt{\lambda_\alpha} V_{\alpha i}$$

3. MATHMATICAL DESCRIPTION OF THE SUPERVISED SDE ALGORITHM

SDE is an unsupervised dimensionality reduction algorithm. It aims at taking a set of high dimensional data and mapping them into a low dimensional space while preserving local isometry structure of the data. However, it discards the class information, which is significant for classification tasks such as face recognition and email text categorization. To complement the original SDE with the additional class information, we propose a supervised SDE algorithm (SSDE) which utilizes the classes label information efficiently. Let ω_i to denote the label of sample data X_i, $i = 1, 2, \cdots, N$. The name of this proposed algorithm implies that membership information is employed to form the neighborhood of each point, that is, nearest neighbors of a given X_i are chosen only from representatives of the same class as X_i. In other words, the idea of SSDE is to select the neighbors of X_i in SDP from only the class that X_i itself belongs to. This nearest neighbor finding procedure can be conducted since we assume that all the training data are labeled.

The essence of the supervised SDE consists of the following steps. Suppose the dataset $\Delta = \{X_i, i = 1, 2, \cdots, N\}$ includes all the labeled training sample data. First, the whole data set Δ is divided into

subsets $\Delta_1, \Delta_2, \ldots \Delta_m$ such that $\Delta = \Delta_1 \cup \Delta_1 \ldots \cup \Delta_m$ and $\Delta_i \cap \Delta_j = \phi, \forall i \neq j$. Each Δ_i holds the data of one class only and m is the total number of classes known a priori.

Each Δ_i is treated separately from others as follows. For each data point $X_i \in \Delta_1$, we look for its K nearest neighbors also belongs to Δ_1, i.e., both X_i and its neighbors have the same class membership. When applied to all data points, this procedure leads to a construction of the neighborhood matrix Γ_x. Thus, whenever X_j is the neighbor of X_i and they belong to the same class, $\tau_{ij} = 1$.

After that, Step 2 and 3 are just as in case of the unsupervised SDE.

Since SSDE is a supervised algorithm, the discussion above is only the training process. For the testing process after embedding, to learn a mapping from a high-dimensional space into low dimensional space, a nonlinear regression analysis approach [4, 10] is used, i.e. mapping unseen points into the embedding space, where the classification of these points is done by classifier, such as the K Nearest Neighbor (KNN) classifier. The detail steps of the SSDE algorithm are summarized in Table 1.

Table 1 SSDE algorithm

Training process: using the training data X and the label. ω_i with each X_i

Step 1 select the neighbors of X_i just from the class that X_i itself belongs to. Get the binary matrix Γ, such that X_j is the neighbor of X_i if and only if $\tau_{ij} = 1$.

Step 2 compute the Gram matrix K through the following optimize problem with $G_{ij} = X_i \cdot X_j$:

$$Max\ Tr(K) \text{ subject to } \mathrm{K} \geq 0, \sum_{\mathrm{ij}} \mathrm{K} = 0,$$

and $\forall ij$ such that $\tau_{ij} = 1$:

$$K_{ii} + K_{jj} - K_{ij} - K_{ji} = G_{ii} + G_{jj} - G_{ij} - G_{ji}$$

Step 3 extract a low-dimension embedding from the dominant eigenvector of the Gram matrix K.

Testing process: using unlabeled data

Project all of the unlabeled data u to a low-dimensional representation by nonlinear regression. Then, the classification of these points is done by classifier, just like the K Nearest Neighbor (KNN).

4. EXPERIMENTS

To illustrate the property of our proposed Supervised Semi-definite Embedding (SSDE) algorithm, we give an intuitive illustration of this algorithm on a synthetic data set in the first subsection. To give experimental results on public dataset, we conduct our proposed algorithm on some benchmark datasets and we take SLLE and the original unsupervised SDE as the baselines. The results of experiments show that the supervised SDE performs very well on the data which exhibits a manifold structure.

4.1. Synthetic Data

For better comprehension of our proposed SSDE algorithm, we give a group of intuitive pictures on the traditional synthetic data [14] of nonlinear dimension reduction algorithms to illustrate the nonlinear property of it in Figure 1. The (a) picture of Figure 1 is the original dataset which is composed of four classes, i.e. dots, triangles, circles and stars respectively. They are mixed together by SLLE in 2-dimensional (b) picture. The (c) picture is these four classes of data after calculated by our algorithm. It can be seen that they are separated into different groups clearly. This is one of the advantages of SSDE.

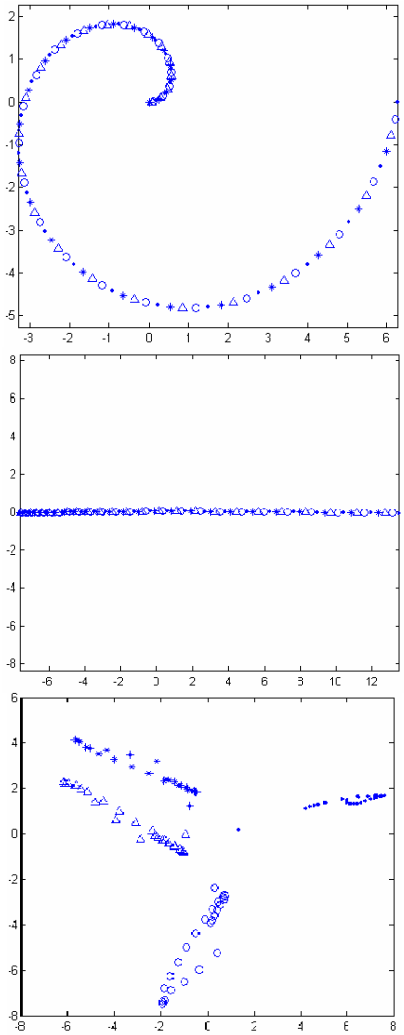

Figure 1 Nonlinear property of SSDE: (a) original data, (b) SLLE, (c) SSDE

4.2. Image Data

In this experiment, all the results presented here were obtained using the well known US Postal Service (USPS) handwritten recognition corpus. This dataset include 36 classes and 39 examples of each class. Each sample is binary 20x16 digits (320 dimensions) of "0" through "9" and capital "A" through "Z".
(*http://www.cs.toronto.edu/~roweis/data.html*)
The handwritten dataset was randomly split into a training set (80%) and a test set (20%) for 10 times. The experimental results are the average of the ten runs. Moreover, a nonlinear regression analysis algorithm to project the test data and a KNN classifier with k=3 are used. Figure 2 shows the precision on the test data. The X-axis denotes the dimension of the feature.

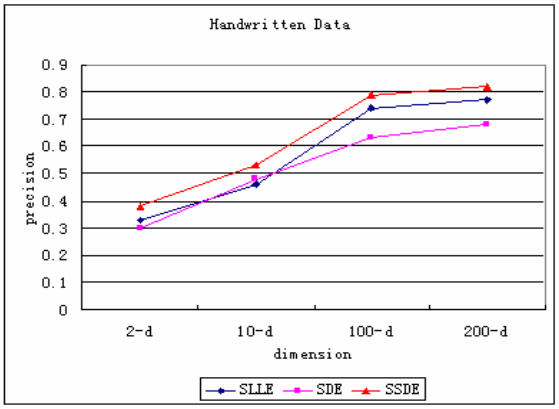

Figure 2 Precisions of handwritten dataset

5. CONCLUSION

In this paper, we propose a novel supervised learning of image manifold by semi-definite programming. Unlike the original unsupervised SDE algorithm, we studied this algorithm when initial dataset were drawn from several classes. It aims at taking a set of high dimensional data and mapping them into a low dimensional space while preserving not only the local isometry structure of the data but also the class information of the data. In contrast to other dimension reduction algorithms, which could be used to clean the noise features, such as SLLE and the original SDE, experiments on a number of benchmark datasets which clearly exhibit manifold structure demonstrated that SSDE is a powerful embedding. Further research will address the problem of choosing the dimension of lower space in a more well-founded way for SSDE.

6. ACKNOWLEDGEMENT

Ning Liu would like to thank the kindly help of K. Q. Weinberger for discussion of SDE source code usage.

7. REFERENCES

[1] Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison Wesley Longman, 1999.
[2] Bimbo, A.D. *Visual Information Retrieval*. Morgan Kaufmann, May 23, 2000.
[3] BORCHERS, B. CSDP, A C Library for Semidefinite Programming. *Optimization Methods and Software*, *11*. 613-623.
[4] Fukunaga, K. *Introduction to Statistical Pattern Recognition (2 edition)*. Academic Press, New York, 1990.
[5] John Blitzer, Kilian Weinberger, Lawrence Saul and Pereira., F., Hierarchical distributed representations for statistical language modeling hierarchical distributed representations for statistical language modeling. In *Proceedings of the Neural Information Processing Systems 17,* (Cambridge, MA, 2005), MIT.
[6] K. Q. Weinberger, F. Sha and Saul, L.K., Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the the Twenty First International Confernence on Machine Learning (ICML-04),* (Banff, Canada., 2004).
[7] Mikhail Belkin and Niyogi, P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation, 15* (6). 1373 - 1396.
[8] Ridder, D.D. Locally linear embedding for classification Technical report PH-2002-01, Pattern Recognition Group, Dept. of Imaging Science & Technology, Delft University of Technology, 2002, 1-15.
[9] Saul, L.K. and Roweis, S.T. Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds. *Machine Learning Research*, *4*. 119-155.
[10] Seber, G.A.F. and Wild, C.J. *Nonlinear regression*. Wiley, New York, 1989.
[11] Sturm, J.F. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*. 625-653.
[12] Tenenbaum, J.B., Silva, V.d. and Langford, J.C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, *290*. 2319--2323.
[13] Weinberger, K.Q., Packer, B.D. and Saul, L.K., Nonlinear Dimensionality Reduction by Semidefinite Programming and Kernel Matrix Factorization. In *Proceedings of the the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS-05),* . (Barbados, 2005).
[14] Weinberger, K.Q. and Saul, L.K., Unsupervised Learning of ImageManifolds by Semidefinite Programming. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR*, (Washington, DC, 2004).