Sparse Subspace Denoising for Image Manifolds

Bo Wang

Department of Computer Science, University of Toronto
wangbo.yunze@gmail.com

Zhuowen Tu
Lab of Neuro Imaging, University of California, Los Angeles
zhuowen.tu@gmail.com

Abstract

With the increasing availability of high dimensional data and demand in sophisticated data analysis algorithms, manifold learning becomes a critical technique to perform dimensionality reduction, unraveling the intrinsic data structure. The real-world data however often come with noises and outliers; seldom, all the data live in a single linear subspace. Inspired by the recent advances in sparse subspace learning and diffusion-based approaches, we propose a new manifold denoising algorithm in which data neighborhoods are adaptively inferred via sparse subspace reconstruction; we then derive a new formulation to perform denoising to the original data. Experiments carried out on both toy and real applications demonstrate the effectiveness of our method; it is insensitive to parameter tuning and we show significant improvement over the competing algorithms.

1. Introduction

In computer vision and machine learning, input data are often given as a set of high dimensional data points. A common technique to deal with data of large dimension is principal component analysis (PCA). For simple data such as well-aligned faces, PCA techniques are shown to be well applicable; for data of high complexity, a single linear subspace then may not be sufficient to capture the intrinsic data structure. In this case, mixture models (i.e. mixture of PCA [22]) and product of linear subspaces (generalized principal component analysis [24]) are proposed.

In more general cases of manifold learning, nonlinearality is often assumed to preserve the lower dimension embedding [18, 21, 16, 3]. Both the ISOMAP [21] and LLE [16] make an assumption of local Euclidean space and try to preserve similar neighborhood structures in the embedded spaces. The spectral manifold learning approaches [3] compute the Eigen-maps for the graph Laplacian to directly perform non-linear dimensionality reduction. From a different angle, diffusion maps [8] defines diffusion distances between data samples; an input similarity measurement is improved through a diffusion process. Diffusion-based methods [8, 14] essentially perform denoising on the similarity metric on which spectral manifold learning methods can be applied. The idea is to explicitly construct a new embedding space with a corresponding metric which is more faithful to the manifold structure and hence induces a better distance/similarity measure. These diffusion-based algorithms [8, 14] have been successfully applied to clustering. The same idea can be extended to the semi-supervised learning, where a limited portion of data labels are given [30].

All the manifold learning methods above depend on the construction of a good neighborhood graph, either explicitly [21, 16] or implicitly [3, 8, 25, 26]. When there are a considerable amount of noises, the neighborhood assumption starts to break down, leading to unsatisfactory results for metric learning, clustering, and classification.

Essentially, the neighborhood or k-nearest neighborhood definition forms a sparse sample assumption. Recent seminal work in compressive sensing [4] demonstrates the significance of having the sparse assumption in reconstruction, if the data is indeed sparse. The idea of local sparse data reconstruction [27] has achieved some significant improvement in codebook-based image classification task. Moreover, as discussed before, data manifolds are often composed of multiple subspaces. The sparse subspace clustering algorithm (SSC) [9] provides a tractable way of learning subspaces with the in-class sparse self-reconstruction assumption. Here, inspired by sparse subspace learning and diffusion-based approaches, we propose a new manifold denoising algorithm in which data neighborhoods are adaptively inferred via sparse subspace reconstruction; we then

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.67

derive a new formulation to perform denoising to the original data. In the experiments, we show significant improvements over the existing manifold denoising algorithms. Although we use the assumption in the SSC, our approach differs to SSC in the objective, formulation, algorithm design, and the application. Next, we review related work in manifold denoising and then give the detailed formulation of our algorithm.

2. Related Work

Although manifold learning is an important topic in machine learning, manifold denoising has received relatively less attention. Unlike the traditional manifold learning and dimension reduction algorithms, the goal of a denoising algorithm is to obtain a cleaner output in the same dimensionality as the input data. The outputs of a denoising algorithm can be further used to facilitate tasks such as feature selection, clustering, and classification. The simplest denoising technique is PCA which adopts a strategy of projection and backward mapping. This two-phase method is often prone to errors and sensitive to noises.

Some more sophisticated methods proposed recently fall into two categories: manifold-based averaging [5, 28, 20, 23] and Laplacian-based spectral analysis [13, 11]. The first line of research focuses on different designs of smooth kernels which are used to average the local manifolds. The idea behind them is that noises in the samples follows a normal distribution and averaging local manifolds can reduce the noise degree. These methods benefit from small computational burden and good effectiveness when dealing with the standard noises. However, in high dimensional data spaces, the assumption of standard noise often fails; in addition, averaging tends to over-smooth the data samples. Furthermore, it is essential to choose the right parameters in kernel construction for these methods. The second category makes use of the Laplacian structures to smooth the data samples. These methods obtain denoised outputs by reversing a diffusion process of which graph Laplacian is the generator. However, the problem associated with them is that graph Laplacian only captures global gradient distribution of the manifolds and it fails to reduce the local noise inherited in sub-manifolds which are important to pattern discovery such as clustering and classification. Lastly, both categories face the same challenge to choose the number of neighbors when either inferring the local manifolds or choosing local reliable clusters.

In this paper, we try to overcame these problems using two strategies: 1) We use a recent subspace learning algorithm to capture the subspace structures which indicate a good potential clusters and also a reliable similarity measure. 2) We design a local sparse regularization term which shares several common algebraic properties with graph Laplacian but with the difference in emphasizing the local subspace distributions. These two strategies offer us the advantages to be less sensitive to the of scale of local neighbors and free of the problem of over-smoothing.

3. Method

3.1. Notations

We assume a given set of n data samples/points, denoted as $S = \{x_1, x_2, \ldots, x_n\}$ where each $x_i \in \mathbb{R}^m$ and m indicates the input data dimension. We then denote them with a matrix representation as $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n}$. The objective of manifold denoising in this paper is therefore to compute a new denoised data matrix $\bar{X} \in \mathbb{R}^{m \times n}$ to facilitate clustering and classification in further applications. We assume the existence of C subspaces. Each data sample x_i here is associated with a label indicator vector $\mathbf{y}_i \in \{0,1\}^C$ such that $\mathbf{y}_i(k) = 1$ if x_i belongs to the k-th subspace, and otherwise $\mathbf{y}_i(k) = 0$. So a *Partition Matrix* $\mathbf{Y} = [\mathbf{y}_1^T; \mathbf{y}_2^T; \ldots; \mathbf{y}_n^T]^T \in \{0,1\}^{n \times C}$ is used to represent a clustering scheme. A similarity graph is then represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertices $\mathcal{V}$ correspond to the data samples $\{x_1, x_2, \ldots, x_n\}$, and the edges $\mathcal{E}$ are weights denoted by an $n \times n$ similarity matrix W with W_{ij} indicating the similarity between x_i and x_j. $\mathcal{N}_i$ represents a set of x_i's neighbors in graph $\mathcal{G}$, excluding x_i and $K_i = |\mathcal{N}_i|$.

3.2. Background: Sparse Subspace Clustering

Sparse subspace clustering(SSC) [9] builds clustering on sparse learning with applications demonstrated in motion segmentation [9]. The motivation of SSC is simple in that each data sample x_i can be expressed as a linear combination of all the data within the cluster, $x_i = \sum_{j \neq i} a_{ij} x_j$ with implicit enforcement of sparsity. That is to learn a sparse coefficient matrix $A \in \mathbb{R}^{n \times n}$ such that $a_{ij} = 1$ if x_i and x_j belong to a same subspace. Formally, SSC solves a following convex optimization problem:

$$\begin{aligned} &\min_A \|XA - X\|_F^2 + \beta \|A\|_1 \\ &s.t.\ \ diag(A) = 0, \end{aligned} \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm and diag(A) is the diagonal vector of matrix A.

The sparse matrix A has two important usages: 1) For each datum x_i, its neighbors $\mathcal{N}_i$ can be easily inferred by the nonzero elements in the i-th column of A. Therefore, we don't need to specify the number of nearest neighbors for each datum as most of other methods do. In addition, we can theoretically guarantee that all chosen neighbors of x_i belong to the same subspace with x_i. 2) A natural similarity graph can be constructed in a easy way: $W = |A| + |A|^T + I$, where I, the identity matrix, is used to enhance self-similarity.

3.3. Sparse Subspace Denoising

3.3.1 Motivations

A straightforward idea behind a good denoising algorithm is that, for each subspace, the new denoised data still reserves the intrinsic allocation of subspaces and also can be easily reconstructed by simple PCA algorithm. Hence, we proposed an optimization framework that aims to minimize both subspace coherence error and reconstruction error. Take Fig.(1) as an illustrative examples. Initially we have 3 overlapping subspaces, and some noisy points are located at the overlapping part of subspaces(see the left part of Fig.(1)). An good denoised output should satisfy two metrics: 1) Points belonging to the same subspaces should be still in the same subspaces; 2) They are close to what they were. The first metric can be achieved by minimizing the subspace coherence error while the second is satisfied by minimizing the reconstruction error. The right side of Fig.(1) is the output of our algorithm. We can see that, those noisy points are moved towards the center of the subspace they belong to and at the same time, we reserve the positions of most of the points.

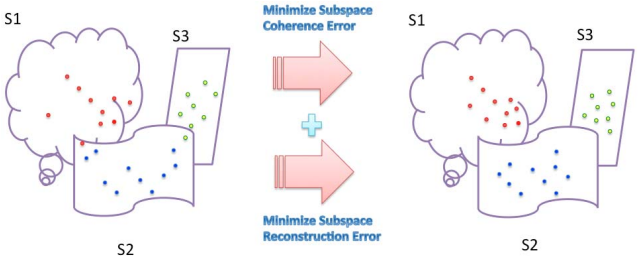


Figure 1. Illustrative examples for Sparse Subspace Denoising(SSD). Best seen in larger resolution.

3.3.2 Subspace Coherence Error

Instead of using partition matrix $\mathbf{Y}$ directly, we adopted a *Scaled Partition Matrix*, $\mathbf{G}$, such that

$$\mathbf{G} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} = [\mathbf{g}^1, \mathbf{g}^2, \ldots, \mathbf{g}^C],$$

where $\mathbf{g}^k = [g_{1k}, \ldots, g_{nk}]^T (1 \leq k \leq C)$ is the k-th column of $\mathbf{G}$. $g_{ik} = Y_{ik}/\sqrt{n_k}$, where n_k is the size of the k-th subspace, can be regarded as the confidence that x_i is assigned to the k-th subspace. It is easy to verify that

$$\mathbf{G}^T\mathbf{G} = (\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}\mathbf{Y}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}} = I,$$

where I is the identity matrix.

The principle of local denoising is that the subspace assignments in the neighborhood of each patient should be as smooth as possible. Specifically, it assumes that the cluster indicator value at x_i should be well inferred by the local neighbors $\mathcal{N}_i$. So given a similarity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we extract a subgraph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ such that $\mathcal{V}_i = \mathcal{N}_i \cup x_i$ and $\mathcal{E}_i = \mathcal{E}(\mathcal{V}_i)$. The similarity matrix associated with the subgraph $\mathcal{G}_i$ is $W^i = W(\mathcal{E}_i)$. Using the label diffusion algorithm [29], we can reconstruct a virtual label indicator vector $\mathbf{p}_k$ such that

$$\mathbf{p}_k = (1-\alpha)(I - \alpha\mathbf{L}_i)^{-1}\mathbf{g}^k_{(\mathcal{V}_i)}, 1 \leq k \leq C, \quad (2)$$

where α is a constant ($0 < \alpha < 1$) and $\mathbf{g}^k_{(\mathcal{V}_i)}$ is the scaled subspace indicator vector on the subgraph $\mathcal{V}_i$. $\mathbf{L}_i$ represents the normalized transition matrix for the subgraph W^i, i.e., $\mathbf{L}_i(u,v) = \frac{W^i(u,v)}{\sum_{l=1}^{K+1} W^i(u,l)}$. Note that we do not actually perform any diffusion, since our setting is completely unsupervised, rather we use $\mathbf{p}_k$ as another way to get to an estimate of g_{ik}. The $\mathbf{p}_k$ is a vector of $K_i + 1$ and $\widehat{g}_{ik} = \mathbf{p}_k(K_i + 1)$ is the estimated cluster assignment indicator for g_{ik}. We should have $\widehat{g}_{ik} \approx g_{ik}$. Denote $\beta_i \in \mathbb{R}^{K+1}$ as the last row of the matrix $(1-\alpha)(I - \alpha\mathbf{L}_i)^{-1}$, then we have

$$\widehat{g}_{ik} = \beta_i \mathbf{g}^k_{(\mathcal{V}_i)}; \quad (3)$$

Hence

$$\widehat{g}_{ik} \approx \frac{\beta_i(1:K_i)\mathbf{g}^k_{(\mathcal{N}_i)}}{1 - \beta_i(K_i+1)}; \quad (4)$$

where $\beta_i(1:K_i)$ represents the first K_i elements of β_i and $\beta_i(K_i+1)$ is the $(K+1)$-th element in β_i.

Therefore, we can construct a matrix B such that

$$B_{ij} = \begin{cases} \frac{\beta_i(l)}{1-\beta_i(K_i+1)} & \text{if } x_j \text{ is the } l\text{-th element in } \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

This sparse matrix B represents a linear relationship that $\widehat{\mathbf{g}}^k \approx B\mathbf{g}^k$, $(k = 1, \ldots, C)$. To infer the subspace indicators, one objective is to minimize the difference between $\widehat{\mathbf{g}}^k$ and $\mathbf{g}^k$:

$$\begin{aligned} & \sum_{i=1}^{n}\sum_{k=1}^{C}(\widehat{g}_{ik} - g_{ik})^2 = \sum_{k=1}^{C}\|\widehat{\mathbf{g}}^k - \mathbf{g}^k\|^2 \\ \approx & \sum_{k=1}^{C}\|\mathbf{g}^k - B\mathbf{g}^k\|^2 \\ = & \; Trace(\mathbf{G}^T(I-B)^T(I-B)\mathbf{G}) \end{aligned} \quad (6)$$

Denote $Z = (I-B)^T(I-B)$, then our subspace assignment results can be obtained by solving the following optimization problems:

$$\begin{aligned} \min_{\mathbf{G}\in\mathbb{R}^{n\times C}} \quad & Trace(\mathbf{G}^T Z\mathbf{G}) \\ s.t. \quad & \mathbf{G}^T\mathbf{G} = I. \end{aligned} \quad (7)$$

This problem (7) can be solved by performing eigen-decomposition of matrix Z [1]. That is, the optimal continuous solution for (7) is the first C eigenvectors of the matrix Z corresponding to the first C smallest eigenvalues.

The idea behind the denoising principle is that the denoised features should be coherent to the subspace assignments. So for each new feature $\mathbf{f}$, we define the coherence error CE as follows:

$$CE(\mathbf{f}) = \sum_{j=1}^{n}(f_j - \sum_{l\in\mathcal{N}_j} B_{jl}f_l)^2 = \|\mathbf{f} - B\mathbf{f}\|^2 = \mathbf{f}^T Z\mathbf{f} \quad (8)$$

From (8), we can see that the coherence error (CE) evaluates two aspects : 1) The extent of match between the feature and the local structure of the image manifold; 2)The fitness between the feature and the subspace assignments. The smaller CE is, the more relevant to the subspace structures the feature is.

To understand the later property further, we expand it in a deeper way(given any feature vector $\mathbf{f}$ in the output data $\widetilde{X}$):

$$\begin{aligned}\frac{CE(\mathbf{f})}{\|\mathbf{f}\|^2} &= 1 - \frac{\mathbf{f}^T(I-Z)\mathbf{f}}{\mathbf{f}^T\mathbf{f}} = 1 - \frac{\mathbf{f}^T[\sum_{i=1}^{n}(1-\lambda_i)\mathbf{g}_i\mathbf{g}_i^T)]\mathbf{f}}{\|\mathbf{f}\|^2} \\ &= 1 - \sum_{i=1}^{n}(1-\lambda_i)\cos^2(\Theta_{<\mathbf{g}_i,\mathbf{f}>}) \\ &\approx 1 - \sum_{i=1}^{C}(1-\lambda_i)\cos^2(\Theta_{<\mathbf{g}_i,\mathbf{f}>}) \quad (9)\end{aligned}$$

where λ_i is the eigenvalues of the matrix Z and $\Theta_{<\mathbf{g}_i,\mathbf{f}>}$ is angle between the feature $\mathbf{f}$ and the eigenvector $\mathbf{g}_i$. Recall that the optimal solution of the Scaled Partition Matrix is the first C normalized eigenvectors corresponding to the first C smallest eigenvalues. Therefore the smaller the CE, the larger the fitness between $\mathbf{f}$ and the Scaled Partition Matrix $\mathbf{G}$.

Therefore, one objective can be defined as the coherence error for the whole denoised data $\widetilde{X} \in \mathbb{R}^{n\times m}$ as follows:

$$\begin{aligned}\mathcal{L}_{CE} &= \sum_{j=1}^{m} CE(\widetilde{X}_j) \\ &= \sum_{j=1}^{m} \|\widetilde{X}_j - B\widetilde{X}_j\|^2 = Trace(Z\widetilde{X}^T\widetilde{X}) \quad (10)\end{aligned}$$

3.3.3 Subspace Reconstruction Error

Another direction of denosing is to make sure the new data points are not far away from the initial data manifolds which are inferred by traditional data reconstruction technique PCA. We look at each data point x_i and its neighbors $\mathcal{N}_i$, and assume them to be random samples from a linear subspace, approximating a local manifold around x_i (To be exact, it can be guaranteed that neighbours inferred by sparse subspace clustering(1) share the same subspace [9]). Let X_i denotes the points in $x_i \bigcup \mathcal{N}_i$, we can approximate it with PCA:

$$R_i = U_iU_i^T X_i(I - \mathbf{1}\mathbf{1}^T/(1+k_i)) + X_i\mathbf{1}\mathbf{1}^T/(1+k_i) \quad (11)$$

where $U_i \in \mathbb{R}^{m\times d_i}$ denote the d_i principal components from PCA and $k_i = |\mathcal{N}_i|$. The number of the principal component d_i can be estimated adaptively. We estimate d_i by setting the variance of the chosen d_i principal components account for at least 90% the whole variance.

So for the preferred denoised data $\widetilde{X}$, we seek to minimize the sum of discrepancies to all neighbors. Specifically, we minimize the following reconstruction error term:

$$\mathcal{L}_{RE} = \sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i} \|\widetilde{x}_i - R_i(j)\| \quad (12)$$

where $R_i(j)$ stands for the local reconstructed data for the j-th neighbor for x_i in (11). This loss function can be expressed in a more compact way with data selection matrices. Let $S = [S_1, S_2, \ldots, S_n]$ be a $1 \times n$ block matrix where each block $S_i \in \mathbb{R}^{n\times(1+k_i)}$ corresponds to the neighborhood indicator matrix. That is, S_i is a binary matrix and its element $s_{jk} = 1$ if and only if x_j is the k-th neighbor of x_i. Therefore we can express the reconstruction error in matrix form:

$$\mathcal{L}_{RE} = \|\widetilde{X}S - R\|_F \quad (13)$$

where $R = [R_1, R_2, \ldots, R_n]$ and $\|\ldots\|_F$ denotes Frobenious norm.

Hence the overall error can be expressed as a combination of subspace coherence error and subspace reconstruction error:

$$\mathcal{L}(\lambda) = \mathcal{L}_{CE} + \lambda\mathcal{L}_{RE} = Trace(Z\widetilde{X}^T\widetilde{X}) + \lambda\|\widetilde{X}S - R\|_F \quad (14)$$

Empirically, the coefficient $\lambda \geq 0$ is chosen with validation sets.

Fortunately, we have a closed-form of the optimal solution of (14):

$$\widetilde{X}^*(\lambda) = RS^T(SS^T + \frac{1}{\lambda}Z)^{-1} \quad (15)$$

Note that our algorithm depends on the neighborhood graphs, so our algorithm is potential to be generalized into an iterative way. After computing the denoised output $\widetilde{X}^*(\lambda)$ in (15), the sparse subspace matrix can be refined. A summary of the proposed algorithm is presented in Algorithm.(1).

3.4. Analysis

In this section, we provide some preliminary analysis of our method and its relationship with some existing denoising algorithms. First, we consider two extreme cases: 1) if $\lambda = 0$, our algorithm degenerates to over-smooth case in which the solution $\widetilde{X}$ becomes trivially constant; 2) if $\lambda = \infty$, our algorithm takes the form of $\widetilde{X}^*(\infty) = RS^T(SS^T)^{-1}$. An easy calculation shows that this is the average of all local neighbors. This is very similar with [6] in which mean-shift deblurring technique is used.

Algorithm 1: Summary of Sparse Subspace Denoising (SSD)

Input: Noisy data X, λ
Output: Clean data $\widetilde{X}$
Step1: Construct Sparse Coefficient matrix A by eqn. (1);
Step2: Compute subspace coherence error by constructing Z from eqn. (5);
Step3: Compute reconstruct error with R and S given A;
Step4: Obtain the results by eqn. (15);
Step5: (optional) Iterate from Step1 using the results from Step4 in replace of X.

Another line of denoising algorithms tends to use Laplacian-Bertrami operator on smooth manifolds [13, 11]. Iterative Laplacian regularization is used to approximate the global gradient of the data manifolds. They often use Normalized Laplacian $\mathcal{L}^+ = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ where D is the degree matrix of the similarity matrix W. The key difference from ours is that, Laplacian regularization only considers the global structure while our algorithm focus on local subspace structure. As pointed by [11], pure Laplacian denoising tends to be over smooth and local structure is more helpful to reduce the effect of noise for applications such as clustering and classification afterwards.

4. Experiements

4.1. Toy Data

In this section, we test our algorithm on two toy datasets. The first one consists of two moon-like clusters, and the second one contains two concentric circles. We add random noise $\epsilon \sim \mathcal{N}(0, \sigma)$, where σ is the standard deviation. Each dataset consists of 1000 data points for two classes.

First, we set $\sigma = 1.5$ and run our algorithm iteratively for three rounds. We show the denoised results on Fig.(2). We can see that, our SSD can successfully remove the added noise and therefore facilitate the applications afterwards. We also test the sensitivity of SSD with respect to the parameter setting of λ. we vary λ from 0.1 to 20, and calculate the classification error(here we use Local and Global Consistency (LGC) [29] as the basic classifier). Note that, we use 10 randomly chosen labeled points and the rest as test. For each λ, we report the average value of errors from 10 independent runs. The result is shown in Fig.(3)(A). It is observed that, our results are not sensitive to the parameters, although large values of λ tend to perform a little worse. Hence, in all our experiments, we set $\lambda = 2$.

In addition, we vary the scales of noise (i.e., σ) and test the classification performance(see Fig.(3)(B-C)). We can see that, when noise is small, one round of SSD is enough to clean the data, while iterative SSD can obtain cleaner data with large noise. Therefore, in all our experiments later, we run SSD for three rounds iteratively.

4.2. Real Data

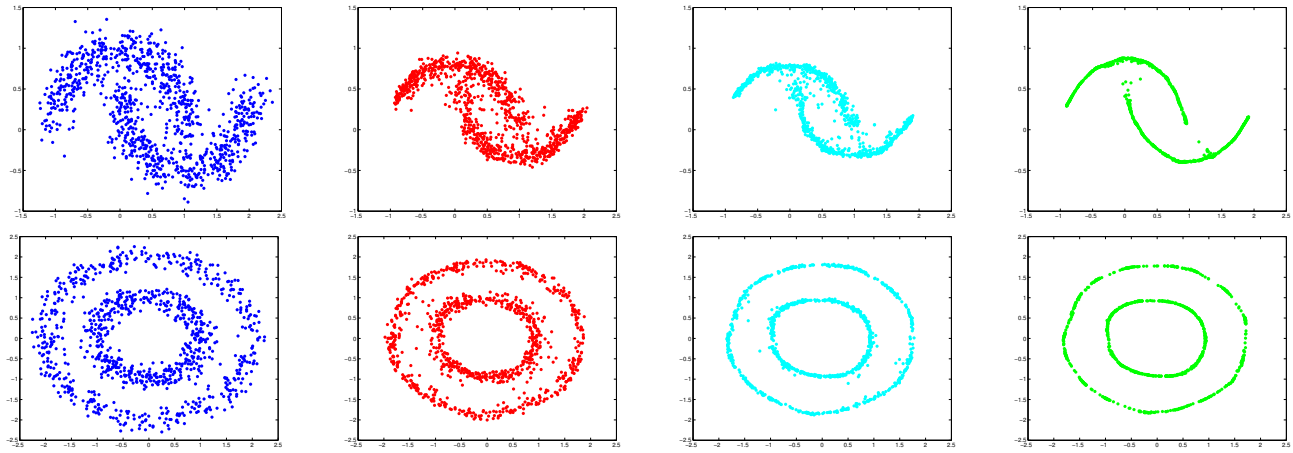
In this section, we use our method as a pre-processing tool to three common applications: clustering, semi-supervised learning(SSL) and supervised classification. And we compare our method to a few existing manifold denoising algorithms : 1) Generalized Blurring Mean Shift (GBMS) [5]; 2) Manifold Blurring Mean Shift (MBMS) [28]; 3) Manifold Denoising (MD) [13]; 4) Locally Linear Denoising (LLD) [11]. We used the codes of GBMS and MBMS provided by the authors[1]. Also, the code of MD is also provided by the author[2]. And we implemented the codes of LLD and set the parameters as suggested in [11]. For the proposed method, we set $\lambda = 2$ in (15), and $\beta = 1$ in (1) for all the experiments.

4.2.1 Clustering

In this section, we evaluate the proposed methods on three well-known face datasets: ORL [17], Yale [2], YaleB [10]. The ORL database consists of samples from 40 individuals, each of which owns 10 different images. These images were taken with a tolerance for tilting and rotation up to 20°. The facial expressions are various: open or closed eyes, smiling or non-smiling and even occlusion of glasses. All images are grayscale and normalized to a resolution of 112×92 pixels. This dataset is mainly designed to test the performance under scale and rotation variations. The Yale database contains images from 15 subjects with 11 different samples for each individual. The images differ from lighting conditions (left-light, center-light, right-light), facial expressions (happy, sad, surprised, and so on), and occlusion (with/without glasses). We use the cropped images of size 32×32. This database is used to test the performance under expression and illumination variations. The extended YaleB dataset is a much more complicated one than Yale. It consists of 38 subjects each of which has around 64 different images.

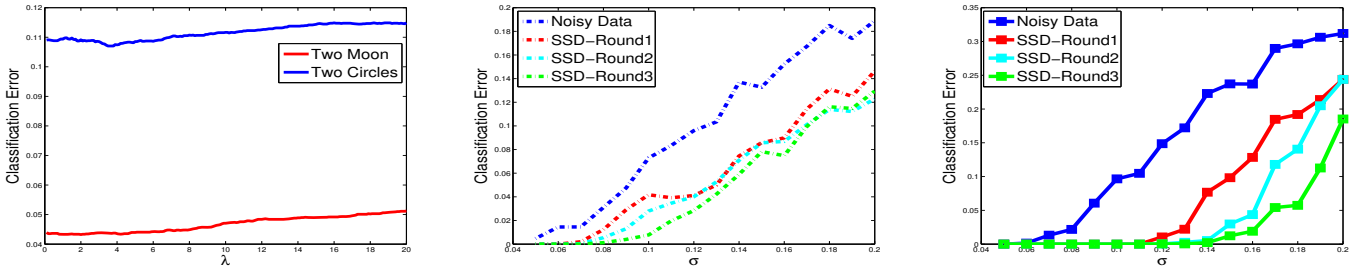
Firstly, we perform clustering on three benchmark face recognition datasets (ORL, Yale, Yale B). We use Normalized Mutual Information (NMI) [19] to evaluate the performance of clustering results. Higher NMI indicates better concordance with the ground-truth labeling. We report the comparison in Tab(1). We can see that, our algorithm outperforms other algorithms consistently. Note that, GBMS and MBMS sometimes even worsen the initial features due to the tendency of over-smooth. Our method does not have this problem.

[1] https://eng.ucmerced.edu/people/wwang5
[2] http://people.kyb.tuebingen.mpg.de/mmaier/ManifoldDenoising.html



(Noised Data) Round 1 Round 2 Round 3

Figure 2. Denoised outputs by SSD for three rounds



(A) (B) (C)

Figure 3. Classification Results. (A) shows the classification errors versus different values of λ, (B)-(C) shows classification error versus different scales of noises on the datasets of two moons and two circles, respectively.

	Baseline	GBMS	MBMS	MD	LLD	SSD
ORL	0.767	0.769	0.793	0.803	0.811	**0.901**
Yale	0.129	0.137	0.127	0.134	0.146	**0.207**
Yale B	0.466	0.453	0.492	0.502	0.483	**0.533**

Table 1. Comparison on Face Clustering.

We also show some illustrative of the denoised images from YaleB in Fig.(4). It is noticeable that SSD is capable to adjust the face orientations and illuminations to make the face image clearer and discriminative.

Figure 4. Examples of denoised images. The upper panel is the initial faces while the lower panel is the denoised output. It is observed that SSD can change the luminance to make the faces easier to identify, e.g., SSD slightly lightened the 4th face and also reduced the pixelation effect around the eyes.

4.2.2 Semi-supervised Learning

We test our method on the benchmarks of Chapelle's book[3]. An extensive review of the performance of existing algorithms are available in [7]. All the databases have 12 splits each of which has 10 labeled and 1400 unlabeled instances. We use Local and Global Consistency (LGC) [29] as a basic semi-supervised learning algorithm. The comparison results are shown in Tab.(2).

We can see that, SSD can improve the baselines by reducing the noise in the local structure. Note that, we don't make use of the label information. Instead, our algorithm can infer the cluster distribution since we aim to minimize the subspace coherence error. Again, we observe that, the other existing methods suffer from the possibility of over smooth thus diminishing the results. However, our method reserves the basic topology of the subspace distribution due to the objective of minimizing the reconstruction error.

[3]The data sets are available at http://www.kyb.tuebingen.mpg.de/ssl-book/benchmarks.html

Table 2. A quantitative comparison on the benchmark datasets. Each split has 10 labeled data.

Methods/Dataset	digit1	USPS	BCI	g241c	COIL	gc241n	text
LGC	15.20	18.51	50.01	48.27	66.23	48.28	42.52
LGC+GBMS	15.34	19.77	50.39	45.24	66.01	44.39	40.13
LGC+MBMS	14.28	19.67	50.46	44.29	65.24	44.77	37.45
LGC+MD	13.06	20.58	49.82	**38.98**	66.59	40.93	34.58
LGC+LLS	14.02	18.68	49.97	42.56	65.68	43.11	35.67
LGC+SSD	**11.02**	**15.33**	**43.14**	40.76	**59.97**	**40.79**	**31.96**

4.2.3 Classification

In this section, we test the utilities of the proposed method as a pre-processing tool for classification. We selected three different datasets: Alphadigits[4], COIL100 [15], and Caltech 256 [12]. Alphadigits is a popular dataset which consists images of ten handwritten digits (0 to 9) and 26 capital letters of the alphabet (A to Z) , with 28×16 pixels. COIL100 is a famous color 3D shape dataset which consists of 100 objects (72 images per object). We used the cropped images of size 32×32. The last dataset, Caltech256, is a well-known natural image database. We selected 41 classes(such as dog, hamburger, motorbikes, and so on) each of which contains various number of objects. And we download well-established SIFT feature [5].

The experimental setting is as follows: we randomly selected different numbers of training data and the rest as test data. For each specific number of training samples, we perform 10 independent runs and report the average accuracy. We use two different basic classifier: 1NN classifier and linear SVM. With denoising algorithms, the classification accuracy can be improved. We report the relative improvement with respect to the baseline. The results are shown in Fig.(5). We can see that, when the number of training samples is very small, our algorithm can still boost up the performance to a great extent. Furthermore, in natural image dataset (e.g., Caltech 256), cluster information is severely corrupted by the inherent noise. In this case, other denoising algorithms failed to improve the data features. However, our SSD is still capable to boost up the recognition performance.

In addition, we show several examples of denoised images compared with the original ones in Fig.(6). The most obvious change is that the denoised ones look smoother and easier to identify. It is noticeable that our algorithm can reduce the pixel noise and even adjust the contents to make the object more salient, while reserving the distinctive style aspect.

5. Conclusion

In this paper, we have introduced a general manifold learning algorithm which takes the advantages of subspace structures and the intrinsic data graph. Our algorithm requires nearly no parameter tuning across different datasets and we show significant improvement over the competing methods on various applications such as clustering and classification. Future work could include a deep theoretically analysis of the convergence property of iterative SSD and an adaptive stopping criteria.

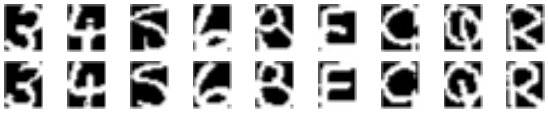


Figure 6. Examples of denoised images. The upper panel is the initial faces while the lower panel is the denoised output. It is observed that SSD reserves the distinctive style of each digit or letter while smoothing a few minor outliers, e.g., digit 5 looks easier to identify after denoting by SSD.

Acknowledgements

This research paper is supported by NSF CAREER award IIS- 0844566, NSF award IIS-1216528, and NIH R01 MH094343.

References

[1] F. R. Bach and M. I. Jordan. Learning spectral clustering. In *Advances in Neural Information Processing Systems 16*. 2004.

[2] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, 1997.

[3] M. Belkin and P. Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. *J. of Comp. and Sys. Sci.*, 74(8):1289–1308, 2008.

[4] E. Candes and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies. *IEEE Trans. Inform. Theory*, 52(2):5406–5425, 2005.

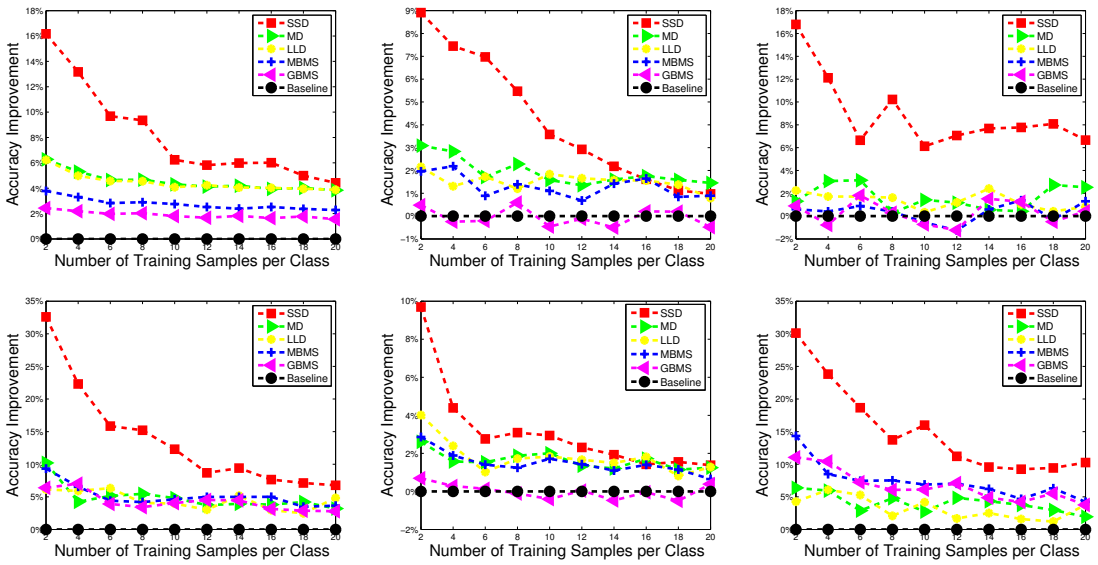
[5] M. Á. Carreira-Perpiñán. Generalised blurring mean-shift algorithms for nonparametric clustering. In *CVPR*, 2008.

[6] M. . Carreira-Perpin. Generalised blurring mean-shift algorithms for nonparametric clustering. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[7] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[4] We downloaded the data from http://www.cs.nyu.edu/ roweis/data.html

[5] http://www.vision.ee.ethz.ch/ pgehler/projects/iccv09/



(A) (B) (C)

Figure 5. Accuracy improvement comparison on three datasets (A-C refers to Alphadigits, COIL100, and Caltech 256 respectively). The upper panels are the results using 1NN classifier. The lower panel are the results using SVM with linear kernels. The baseline refers the results on raw features without any denoising algorithm.

[8] R. Coifman and S. Lafon. Diffusion maps. *Applied and Comp. Harmonic Ana.*, 2006.

[9] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, pages 2790–2797, 2009.

[10] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:643–660, 2001.

[11] D. Gong, F. Sha, and G. G. Medioni. Locally linear denoising on image manifolds. *Journal of Machine Learning Research - Proceedings Track*, 9:265–272, 2010.

[12] G. Griffin, A. Holub, and P. Perona. The Caltech-256. Technical report, California Institute of Technology, 2007.

[13] M. Hein and M. Maier. Manifold denoising. In *NIPS*, pages 561–568, 2006.

[14] J. Jiang, B. Wang, and Z. Tu. Self-smoothing operator for retrieval, clustering, and segmentation. In *Proc. of ICCV*, 2011.

[15] S. Nene, S. Nayar, and H. Murase. Columbia Object Image Library (COIL-100). Technical report, Columbia University, 1996.

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

[17] F. S. Samaria, F. S. S. *t, A. Harter, and O. A. Site. Parameterisation of a stochastic model for human face identification. 1994.

[18] H. S. Seung and D. D. Lee. The manifold ways of perception. *Science*, 290(5500):2268–2269, 2000.

[19] A. Strehl, J. Ghosh, and C. Cardie. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[20] T. Takahashi and T. Kurita. Robust Denoising by Kernel PCA. In *Int. Conference on Artificial Neural Networks*, pages 739–744, 2002.

[21] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2000.

[22] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443482, 2012.

[23] R. Unnikrishnan and M. Hebert. Denoising manifold and non-manifold point clouds. In *BMVC*, 2007.

[24] R. Vidal, Y. Ma, and S. Sastry. Generalized principle component analysis (gpca). *IEEE Trans. Patt. Ana. Mach. Intel.*, 27(12):1–15, 2005.

[25] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu. Unsupervised metric fusion by cross diffusion. In *CVPR*, 2012.

[26] B. Wang and Z. Tu. Affinity learning via self-diffusion for image segmentation and clustering. In *CVPR*, 2012.

[27] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[28] W. Wang and M. A. Carreira-Perpinn. Manifold blurring mean shift algorithms for manifold denoising. In *Proc. of CVPR*, 2010.

[29] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schlkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.

[30] X. Zhu. Semi-supervised learning literature survey. *Computer Science TR 1530, University of Wisconsin-Madison*, 2008.