A Unified Neighbor Reconstruction Method for Embeddings

Zhiling Ye [†], Zhihong Zhang[*†], Lu Bai[‡], Guosheng Hu[§], Zheng-Jian Bai[†], Yiqun Hu[*†], Edwin R. Hancock[¶]
[†] Xiamen University, Xiamen, China
[‡] Central University of Finance and Economics, Beijing, China
[§] Anyvision company, Belfast, UK
[¶]Department of Computer Science, University of York, York, UK
[*]Corresponding author: zhihong@xmu.edu.cn, hyq0826@yahoo.com.

***Abstract*—In this work we propose a novel and compact Neighbor Reconstruction Method (NRM) which is a unified preprocessing method for graph-based sparse spectral algorithms. This method is conducted by vector operations on a central point and its corresponding neighbor points. NRM generates new neighbor points which can capture the local space structure of the central point more appropriately than original neighbor points. With NRM, a large number of sparse spectral based nonlinear feature extraction and selection algorithms gain significant improvement. Specifically, we embedded NRM to several classical algorithms, Local Linear Embedding (LLE) [1], Laplacian Eigenmaps (LE) [2] and Unsupervised Feature Selection for Multi-cluster Data (MCFS) [3], with accuracy improvement of up to 7%, 2.6 %, 2.4 % on ORL, CIFAR 10, and MINST data sets respectively. We also apply NRM to a Super Resolution algorithm, A+ [5], and obtain 0.12dB improvement than original method.**

I. Introduction

In the real world, data such as speech signals and digital figures always have high dimensionality. To process these data adequately, dimensionality reduction is required. In machine learning and statistics, dimensionality reduction is the process of reducing the number of random variables under consideration, and obtaining a set of principal variables [1]. Dimensionality reduction is widely used for feature selection and extraction [2]. Feature selection can be viewed as the search for feature subsets from the full data set, along with an evaluation metric which scores the different feature subsets. Feature extraction transfers the entire original data set into a more informative and non-redundant representation, leading to better human interpretations. The most obvious distinction between feature selection and extraction is that feature selection does not create new features from the original data set, while the latter transfers the entire data set into a new coordinate space. In the following two paragraphs we will detail feature selection and extraction respectively.

Based on different evaluation metrics, feature selection methods can mainly be grouped into three categories: wrappers, filters and embedded methods. Wrapper methods employ a predictive model to score different selected subsets [9], [10], [12]. Wrapper methods can usually achieve best performance for a particular model, but a large amount of computation is needed. Filter models use a proxy metric instead of the error rate used in wrapper to score feature subsets [8], [3], [13], [14]. With less computation, a filter model can provide more general but lower prediction performance than that provided by wrappers. Embedded methods are a catch-all group of techniques which perform feature selection as part of the model construction process [11], [16].

Based on the constructed mapping, feature extraction methods can roughly be classified into linear and nonlinear methods. Linear feature extraction algorithms construct a linear mapping to transfer the original data set into a feature set with designed properties, such as Principal Components Analysis (PCA) [17], Linear Discriminant Analysis (LDA) [19] and Locality Preserving Projection (LPP) [18]. These algorithms perform well on linearly separable data sets. However, these linear techniques cannot adequately handle complex nonlinear data. By constructing a nonlinear mapping from original data set to a feature set with designed properties, nonlinear feature extraction algorithms, such as, Isomap [20], Locally linear Embedding [1], Laplacian Eigenmaps [2], multilayer autoencoders, have the ability to deal with complex nonlinear data.

In the last decade, a manifold based view of data processing has emerges, such as locally Linear Embedding (LLE) [1], Laplacian Eigenmaps (LE) [2], Hessian LLE [21], Local Tangent Space Analysis (LTSA) [22], Unsupervised feature selection for Multi-Cluster data (MCFS) [3], Unsupervised Feature Selection with Structured Graph Optimization (SOGFS) [4]. These algorithms view input data as an undirected weighted graph, which is used to approximate the manifolds. This global graph is constructed by connecting smaller local graphs in a admissible way. These small graphs are generated by connecting data points with their corresponding neighbors. The global graph is represented by a sparse matrix. Based on this sparse matrix, locally linear algorithms solve a sparse eigenvalue problem to generate the designed features. We refer to these sparse spectral dimensionality reduction techniques as graph based sparse spectral algorithms.

Here we propose a novel neighbor reconstruction method which focuses on generating better small local graphs on manifolds. Accordingly, data are viewed as laying on a manifold. Specifically, we present the new method which generates a much closer neighbor by aggregating the central point and its corresponding neighbor points together and dividing the result

by a scalar (detailed in Fig 2). Briefly, our contributions are:

1) The neighbor reconstruction method can generate a closer neighbor for an assigned central point leading to a significant improvement for embedding methods.
2) The neighbor reconstruction method is a unified compact preprocessing method for graph based sparse spectral algorithms. Namely, it can be applied to many existing feature selection and extraction algorithms.
3) We also analyze theoretically the mechanism of neighbor reconstruction method, guaranteeing its convergence to manifolds. And give out the optimised adjustable parameter c.

II. Related work

The problem of graph based sparse spectral feature extraction and feature selection can be defined respectively as follows. Assume we have a data set represented as a $D \times n$ matrix $\mathbf{X}$ consisting of n sample vectors $\mathbf{x}_j (j \in \{1, 2, 3, ..., n\})$ with D dimensionality. Assume that this data set has intrinsic dimensionality d, where $d < D$ and often $d \ll D$. In mathematical terms intrinsic dimensionality means the sample points in data set $\mathbf{X}$ are lying on or near a manifold with dimensionality of d which is embedded in the D-dimensional Euclidean space. In feature extraction, researchers construct some nonlinear mappings to make the high dimensional feature space mapped into a low dimensional one. On the other hand feature selection can be viewed as the search technique for proposing new feature subsets from original data set, along with an evaluation measure which scores the different feature subsets. In the following subsections we analyze three discriminative methods stretching over two research domains. We now review in detail two important sparse spectral dimensionality reduction techniques, Local Linear Embedding (LLE) [1] and Laplacian Eigenmaps (LE) [2].

Local Linear Embedding Firstly, LLE constructs a graph to represent the data set $\mathbf{X}$. In LLE, to describe the local properties of the manifold around a data point $\mathbf{x}_j$, which is written as a linear combination $\mathbf{w}_j$ (the reconstruction weights) of its k nearest neighbors $\mathbf{x}_{j_i} (i \in \{1, 2, 3, ..., k\})$. LLE assumes the reconstruction weights $\mathbf{w}_j$ is shared between high- and low-dimensional space. Hence, LLE utilizes reconstruction weights to obtain corresponding low-dimensional representation.

Laplacian Eigenmaps LE finds a low-dimensional data representation by preserving local properties of the manifold. In LE, the local properties are based on the pairwise distances between near neighbors. Like LLE, LE firstly construct a sparse adjacency matrix $\mathbf{W}_{le}$ in a different way. LE generates a low-dimensional representation of the data in which the distances between a data point and its k nearest neighbors are minimized.

Multi-Cluster Feature Selection (MCFS) Like LE, Unsupervised Feature Selection for Multi-cluster Data (MCFS) [3] also constructs a sparse adjacency distance matrix $\mathbf{W}_{mcfs}$. Based on $\mathbf{W}_{mcfs}$, MCFS employs graph spectral embedding method to generate a low-dimensional representation of original data.

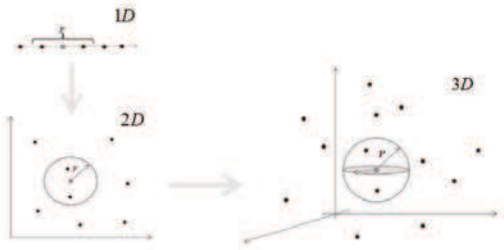

Figure 1: To capture two neighbors within radius r, the number of sampled data exponentially grows with increasing dimension number.

All of these methods can be viewed as different graph-based sparse spectral algorithms. Our analysis is based on a basic property of the manifold: if a local intrinsic manifold subspace is small enough then it can be well described by its corresponding embedded Euclidean subspace. To model the intrinsic manifold, these methods construct a sparse adjacency matrix $\mathbf{W}$. In graph based sparse spectral algorithms, although $\mathbf{W}$ is constructed in different ways, the k nearest neighbors $\mathbf{x}_{j_i}$ are usually used to represent the local subspace. If the neighbors are close enough to the central point $\mathbf{x}_j$, according to this property, the generated sparse adjacency matrix $\mathbf{W}$ can model intrinsic the manifold in a better way than those generated by relatively distant neighbors. This is an explanation why a method can perform better than its original version when a larger training set is used. Clearly a larger data set means that closer neighbors can be found. Due to the high dimension of the data, numerous data would be required to find a sufficiently close neighbor for a central point. Fig 1 illustrate this problem.

Clearly large data set is important for finding a sufficiently close neighbor. However it is massively expensive in both computation and memory. This observation motivates our work in this paper.

III. Neighbor reconstruction method(NRM)

The first step of a graph based sparse spectral algorithm is to select k nearest neighbors for every data point and then to take subsequent processing steps. We add a unified pre-processing step between the first step and the subsequent processing steps to comes for regular procedures. Inspired by Euclid's theorem in plane space, namely the parallelogram axiom of vectors, we designed a new neighbor reconstruction method. More details are shown in Fig 2. Denote the neighbors of $\mathbf{x}_j$ as the set of vectors $[\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, ..., \mathbf{x}_{j_k}] \in \mathbb{R}^{D \times k}$. We concatenate the central point and its corresponding neighbors together as column in the matrix $\bar{\mathbf{X}} = [\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, ..., \mathbf{x}_{j_k}, \mathbf{x}_j]$. We induce a reconstruction operator,

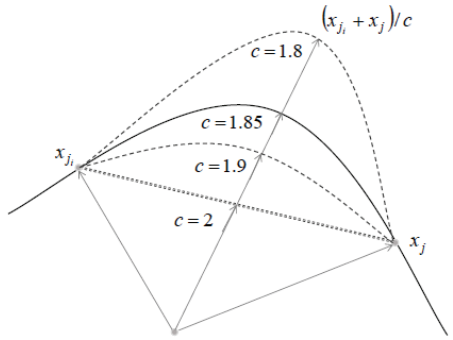

Figure 2: Geometric interpretation of neighborhood reconstruction. The figure shows how to create a cosine similarity closer point $(x_{j_i}+x_j)/c$ by using x_j and its neighbor x_{j_i}. c is an adjustable parameter to make $(x_{j_i}+x_j)/c$ be close to the intrinsic manifold, namely the solid line. In this figure, when $c=1.85$, $(x_{j_i}+x_j)/c$ can fall on the intrinsic manifold.

$$\mathbf{R}=\begin{bmatrix} \frac{1}{c} & 0 & \dots & 0 & 0 \\ 0 & \frac{1}{c} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \frac{1}{c} & 0 \\ \frac{1}{c} & \frac{1}{c} & \frac{1}{c} & \frac{1}{c} & 1 \end{bmatrix} \in \mathbb{R}^{(k+1)\times(k+1)} \tag{1}$$

where $c(>1)$ is an adjustable parameter. For the jth$(1 \le j < k+1)$ column R_j, it can generate the jth reconstructed neighbor $\frac{1}{c}x_{j_i}+\frac{1}{c}x_j$ by the right multiplication $\bar{\mathbf{X}}R_j$. For the $(k+1)$th column, it is used to preserve central point x_j for the next iteration. In NRM, reconstruction manipulation is achieved in parallel by right multiplying $\mathbf{R}$ by $\bar{\mathbf{X}}$. This manipulation can be done achieved iteratively. $\bar{\mathbf{X}}^{(r)}=\bar{\mathbf{X}}\mathbf{R}^r (r\in\{1,2,3,...,s\})$ where s is a truncation number. By mathematical induction, we can obtain a closed form of $\bar{\mathbf{X}}^{(r)}$,

$$\begin{aligned}\bar{\mathbf{X}}^{(r)} &= \bar{\mathbf{X}}\mathbf{R}^r \\ &= \left[\tfrac{1}{c^r}\mathbf{x}_{j_1}+S(r,c)\mathbf{x}_j,...,\tfrac{1}{c^r}\mathbf{x}_{j_k}+S(r,c)\mathbf{x}_j,\mathbf{x}_j\right]\end{aligned} \tag{2}$$

where $S(r,c)=\frac{1}{c^r}\frac{c^r-1}{c-1}$. After operating on $\bar{\mathbf{X}}$ s times, NRM collects $\bar{\mathbf{X}}^{(r)}$ as a large set $\mathfrak{X}=\{\bar{\mathbf{X}}^{(r)}\}_{r=1}^{s}$. The final step in NRM is to select k the nearest points for $\mathbf{x}_j$ from $\mathfrak{X}$ to replace the original neighbor set. The complete NRM algorithm is summarized in Alg. 1

Algorithm 1 NRM

Require:
Central point $\mathbf{x}_j$;
The corresponding neighbors $[\mathbf{x}_{j_1},\mathbf{x}_{j_2},...,\mathbf{x}_{j_k}]$;
Truncation number s;
Adjustable parameter c.
Ensure:
Reconstructed neighbor set $\mathbf{N}$;
1: **for** r=1,2, ..., s **do**
2: Put central point and its corresponding neighbors together $\bar{\mathbf{X}}=[\mathbf{x}_{j_1},\mathbf{x}_{j_2},...,\mathbf{x}_{j_k},\mathbf{x}_j]$;
3: Do the manipulation $\bar{\mathbf{X}}^{(r)}=\bar{\mathbf{X}}\mathbf{R}^r$;
4: Collect $\bar{\mathbf{X}}^{(r)}$ into $\mathfrak{X}=[\bar{\mathbf{X}}^{(1)},\bar{\mathbf{X}}^{(2)},...,\bar{\mathbf{X}}^{(r-1)}]\in\mathbb{R}^{D\times(k+1)r}$;
5: **end for**
6: In $\mathfrak{X}$ select another k nearest neighbors except x_j to be $\mathbf{N}$;
7: **return** $\mathbf{N}$;

Algorithm 2 LLP-NRM

Require:
Central point $\mathbf{x}_j$;
The corresponding neighbors $[\mathbf{x}_{j_1},\mathbf{x}_{j_2},...,\mathbf{x}_{j_k}]$;
Truncation number s;
Shrinkage parameter $\tilde{c}$.
Ensure:
Reconstructed adjacency matrix $\bar{\mathbf{W}}$;
1: Construct sparse adjacency distance matrix $\mathbf{W}=\{w_{ij}\}_{i,j=1}^{n}$, where

$$w_{ij}=\begin{cases} m(x_i,x_j), & j\in\Gamma_i \\ 0, & j\notin\Gamma_i \end{cases}$$

$m(\cdot,\cdot)$ is an assigned metric, Γ_i is a index set which contains column indices of kth smallest values in ith row.

2: **for** r=1,2, ..., s **do**
3: Shrink neighbors: $\mathbf{W}^{(r)}=\frac{1}{\tilde{c}^r}.*\mathbf{W}$;
4: Collect $\mathbf{W}^{(r)}$ into a big matrix $\mathfrak{W}=\left[\mathbf{W},\mathbf{W}^{(1)},...\mathbf{W}^{(r-1)}\right]\in\mathbb{R}^{n\times nr}$
5: **end for**
6: Select k smallest values from $\mathfrak{W}$ to replace non-zero values in $\mathbf{W}$ row by row;
7: **return** $\bar{\mathbf{W}}$;

A. Local Linearly preserving NRM (LLP-NRM)

In many applications the local manifold curvature is small, so we also propose a local linearly preserving NRM which is a very simple form of NRM detailed in Alg. 2. Small curvature leads to a flat data manifold, so the local manifold can be approximated well by a hyperplane which implies the length of the geodesic is almost equal to the Euclidean distance of the sampled points. Namely,

$$l(\mathbf{x}_j,\mathbf{x}_{j_i})\approx\triangle\mathbf{x} \tag{3}$$

Here $l(\mathbf{x}_j,\mathbf{x}_{j_i})$ is the geodesic between $\mathbf{x}_j$ and $\mathbf{x}_{j_i}$, and $\triangle\mathbf{x}=\|\mathbf{x}_j-\mathbf{x}_{j_i}\|_2$. Hence, to obtain closer points, we can directly shrink the neighbors to the center point alone with the vector $\mathbf{x}_j-\mathbf{x}_{j_i}$. This variant method is suitable for those algorithms requiring sparse adjacency distance matrix.

B. Convergence

In this subsection, we analyze $\frac{1}{c^r}\mathbf{x}_{j_i}+S(r,c)\mathbf{x}_j$ to demonstrate the convergence of NRM to the around of the intrinsic

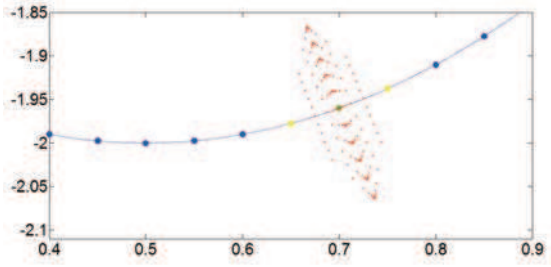

Figure 3: The distribution of NRM generated neighbors. The blue dashed curve is intrinsic manifold. These points on the curve are sampled data points. The green one of the two blue points is center point and others are its corresponding neighbors. Red small points mark out the distribution of reconstructed point when c and s vary.

manifold. And we give out the optimised adjustable parameter c. In Fig 3, performing NRM on a toy example to support our analysis in the following paragraph. For a fixed $c(>1)$, We investigate whether it is useful to iterate. For precise descriptions, we introduce some mathematical deduction.

1) Euclidean metric convergence: Let D be the Euclidean distance between $\frac{1}{c^r}\mathbf{x}_{j_i} + S(r,c)\mathbf{x}_j$ and $\mathbf{x}_j$. Namely,

$$D(\frac{1}{c^r}\mathbf{x}_{j_i} + S(r,c)\mathbf{x}_j, \mathbf{x}_j)^2 = \| \frac{1}{c^r}\mathbf{x}_{j_i} + S(r,c)\mathbf{x}_j - \mathbf{x}_j \|_2^2 \quad (4)$$

We now explore the relationship between r and D, To do this rewrite Eq.(4) as,

$$\begin{aligned} D =& \| \frac{1}{c^r}\mathbf{x}_{j_i} + (S-1)\mathbf{x}_j \|_2 \quad (5) \\ =& < \frac{1}{c^r}\mathbf{x}_{j_i}, \frac{1}{c^r}\mathbf{x}_{j_i} > + 2 < (S-1)\mathbf{x}_j, \frac{1}{c^r}\mathbf{x}_{j_i} > \\ & + < (S-1)\mathbf{x}_j, (S-1)\mathbf{x}_j > \quad (6) \end{aligned}$$

Taking consideration of that $\mathbf{x}_{j_i}$ is a neighbor of $\mathbf{x}_j$, we use a min term $\triangle\mathbf{x}$ to connect them as $\mathbf{x}_j + \triangle\mathbf{x} = \mathbf{x}_{j_i}$. In Eq.(5) replace $\mathbf{x}_{j_i}$ with $\mathbf{x}_j + \triangle\mathbf{x} = \mathbf{x}_{j_i}$ we can obtain a easier equation,

$$\begin{aligned} D =& < \frac{1}{c^r}\mathbf{x}_{j_i}, \frac{1}{c^r}\mathbf{x}_{j_i} > + 2 < (S-1)\mathbf{x}_j, \frac{1}{c^r}\mathbf{x}_{j_i} > \quad (7) \\ =& (\frac{1}{c^r} + S - 1)^2 \| \mathbf{x}_j \|_2^2 + O(\triangle\mathbf{x}) \end{aligned}$$

Firstly, we focus on the first term $(\frac{1}{c^r} + S - 1)^2 \| \mathbf{x}_j \|_2^2$. We denote $g(r,c) = \frac{1}{c^r} + S - 1$. For arbitrary given $r \geq 1$, we have,

$$\begin{aligned} \frac{\partial g(r,c)}{\partial c} &= \frac{\partial(\frac{2}{c^r} + \frac{1}{c^{r-1}} +, ..., + \frac{1}{c^1} - 1)}{\partial c} \quad (8) \\ &= (\frac{-2r}{c^{r+1}} + \frac{-(r-1)}{c^r} +, ..., + \frac{-1}{c^2}) < 0 \end{aligned}$$

which means that $g(r,c)$ monotone decrease for c. On the other hand, its lucky to see that with arbitrary $r \geq 1$, when $c = 1$,

$$\begin{aligned} g(r,c) &= \frac{1}{c^r} + \frac{1}{c^r} + \frac{1}{c^{r-1}} +, ..., + \frac{1}{c^1} - 1 \quad (9) \\ &= \frac{1}{c^r} + \frac{1}{c^r}(1 + c +, ..., + c^r) - 1 \\ &= 1 + \underbrace{(1 + 1 +, ..., + 1)}_{r-1} - 1 \\ &= r - 1 \geq 0 \end{aligned}$$

when $c = 2$,it's easy to see,$g(r,c) = 0$. And when $c \geq 2$, because $g(r,c)$ monotone decrease for c and $g(r,2) = 0$, we can obtain that $g(r,c) \leq 0$. And,

$$\begin{aligned} \lim_{c \to +\infty} g(r,c) &= \lim_{c \to +\infty} \frac{1}{c-1}(1 + \frac{1}{c^{r-1}} - \frac{2}{c^r}) - 1 \quad (10) \\ &= -1 \end{aligned}$$

In summary, with arbitrary $r \geq 1$, $g(r,1) > 0$, $g(r,2) = 0$, and $-1 < g(r,c) < 0, c \in (2,\infty)$. In this paragraph, we will do some analysis on r for $g(r,c)$. Given arbitrary $c \geq 1$, we want to explore the relation between $g(r,c)$ and r. Starting from the definition of monotonicity, we give out the following conclusion,

$$g(r,c) = \begin{cases} \text{monotone decrease for } r & c > 2, \\ \text{monotone increase for } r & 2 \geq c > 1 \ . \end{cases}$$

In consideration of the former analysis on the relation between g and r, c, it is obvious to find that when $c \in [2,1)$, $g(r,c)$ is non negative and monotone increase for r. And when $c \in (2,\infty)$, $g(r,c)$ is negative and monotone decrease for r. Hence $(g(r,c))^2$ is monotone increase for r, namely $D(\frac{1}{c^r}\mathbf{x}_{j_i} + S(r,c)\mathbf{x}_j, \mathbf{x}_j)^2$ is monotone increase for r. Now we return to Eq.(7) and focus on the min term $O(\triangle\mathbf{x})$. It obvious to see that when $g \neq 0$, the min term can be justifiably removed. While when $g = 0$, namely $c = 2$, $O(\triangle\mathbf{x})$ it is non-negligible. we need to do some analysis on $O(\triangle\mathbf{x})$.

$$\begin{aligned} O(\triangle\mathbf{x}) =& < \triangle\mathbf{x}, 2(\frac{1}{c^r})^2\mathbf{x}_j + (\frac{1}{c^r})^2\triangle\mathbf{x} + \frac{(S-1)}{c^r}\mathbf{x}_j > \quad (11) \\ =& ((\frac{1}{2^r})^2 + \frac{1}{2^r})\triangle\mathbf{x}^T\mathbf{x}_j + (\frac{1}{2^r})^2 \| \triangle\mathbf{x} \|_2^2 \end{aligned}$$

Here, $\lim_{r \to +\infty} O(\triangle\mathbf{x}) = 0$. So we can make a conclusion that when $C = 0$, D is monotone decrease for r. While when $c \neq 0$, namely $g \neq 0$, the min term $O(\triangle\mathbf{x})$ can be justifiably removed and D is monotone increase for r.

2) Cosine metric convergence: Let C be the cosine distance between $\frac{1}{c^r}\mathbf{x}_{j_i} + S(r,c)\mathbf{x}_j$ and $\mathbf{x}_j$.

$$C = \mathbf{arccos}\langle \frac{1}{c^r}\mathbf{x}_{j_i} + S(r,c)\mathbf{x}_j, \mathbf{x}_j \rangle \quad (12)$$

where $\langle \cdot, \cdot \rangle$ represents the angle between a pair of vectors. Before we analyse the relationship between C and r, a theorem should be given out,

Theorem 1. *Given vectors* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, *which satisfy* $< \mathbf{a}, \mathbf{b} >\geq 0$, *then we have* $\frac{<\mathbf{a},\mathbf{b}>}{\|\mathbf{a}\|\|\mathbf{b}\|} \leq \frac{<\mathbf{a}+\mathbf{b},\mathbf{b}>}{\|\mathbf{a}+\mathbf{b}\|\|\mathbf{b}\|}$, *namely,* $\mathbf{cos}\langle \mathbf{a}, \mathbf{b} \rangle \leq \mathbf{cos}\langle \mathbf{a}, \mathbf{a}+\mathbf{b} \rangle$.

Following with theorem 1, we give out the convergence of NRM on cosine metric. Firstly we have,

$$\begin{aligned} &\mathbf{cos}\langle \frac{1}{c}\left(\frac{1}{c^{n-1}}\mathbf{x}_{j_i} + S(c, n-1)\mathbf{x}_j\right) + \mathbf{x}_j, \mathbf{x}_j \rangle \qquad (13) \\ &\geq \mathbf{cos}\langle \frac{1}{c^{n-1}}\mathbf{x}_{j_i} + S(c, n-1)\mathbf{x}_j), \mathbf{x}_j \rangle \end{aligned}$$

With recurrent trick,

$$\begin{aligned} &\mathbf{cos}\langle \left(\frac{1}{c^{r}}\mathbf{x}_{j_i} + S(c, r)\mathbf{x}_j\right), \mathbf{x}_j \rangle \qquad (14) \\ &= \mathbf{cos}\langle \frac{1}{c}\left(\frac{1}{c^{r-1}}\mathbf{x}_{j_i} + S(c, r-1)\mathbf{x}_j) + \mathbf{x}_j\right), \mathbf{x}_j \rangle \\ &\geq \mathbf{cos}\langle \frac{1}{c^{r-1}}\mathbf{x}_{j_i} + S(c, r-1)\mathbf{x}_j), \mathbf{x}_j \rangle \\ &\quad ..... \\ &\geq \mathbf{cos}\langle \mathbf{x}_{j_i}, \mathbf{x}_j \rangle \end{aligned}$$

So like D, C is also monotone decrease for r and this property can be preserved for arbitrary c.

Based on the former analysis, we can conclude that when $c = 2$ and vary r, the neighbors generated by NRM is simultaneously convergent to central point on Euclidean metric and cosine metric. This conclusion is suitable for every point and its neighbors.

IV. Experiments

In this section to validate the benefits of NRM, we conduct experiments on two tasks, single image super resolution and image classification.

A. Image super resolution (SR)

In this part we apply NRM on a classical example-based SR method A+ [5]. A+ is a typical neighbor embedding method. By applying NRM on A+ we can construct a better neighborhood than original method leading to better performance measured by quantitative PSNR and (structural similarity) SSIM results. We validate NRM on Set5, Set14, and B100 detailed in Table I and visualize some results in Fig 4 and obtain 0.12dB improvement.

B. Image classification

In this section we apply NRM and LLP-NRM on graph based sparse spectral methods. Firstly, we employ different methods and their reinforced variants to extract or select features from the raw data. Then we randomly and equally split these feature sets into disjoint training and testing sets. We train classifier on the training data. Finally, we compare the performance of classifiers on the test data.

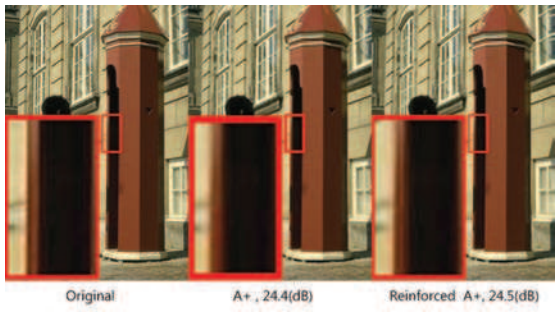

Figure 4: Visualization of NRM applied on SR

data set	s	A+ [5] PSNR	SSIM	Time	Reinforced A+ PSNR	SSIM	Time
Set5	2	36.55	0.9611	0.8	**36.65**	**0.9614**	1.6
	3	32.59	0.9139	0.5	**32.67**	**0.9202**	0.8
	4	30.28	0.8737	0.3	**30.40**	**0.8760**	0.5
Set14	2	32.28	0.9649	1.6	**32.39**	**0.9649**	3.6
	3	29.13	0.8940	0.9	**29.20**	**0.8946**	1.7
	4	27.32	0.8281	0.6	**27.42**	**0.8300**	1.1
B100	2	30.77	**0.8773**	1.1	**30.83**	0.8772	2.3
	3	28.18	0.7791	0.6	**28.23**	**0.7820**	1.1
	4	26.77	0.7085	0.4	**26.83**	**0.7105**	0.7

Table I: Performance of x2, x3,and x4 magnification in terms of averaged PSNR (dB),SSIM and execution time (s) on data set Set5, Set14 and B100.

1) Settings: The ORL[1] face data set contains facial 64×64 images of 40 distinct subjects each of which has 10 images. The MNIST[2] database (Mixed National Institute of Standards and Technology database) is a large database of handwritten

[1]http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
[2]http://yann.lecun.com/exdb/mnist/

methods	2.5K samples	5.0K samples	10.0K samples
LLE [1]	81.34(0.69)	84.46(0.27)	88.48(0.21)
RLLE	**81.69**(0.42)	**86.84**(0.23)	**89.72**(0.20)
LE [2]	84.88(0.39)	88.60(0.27)	92.11(0.13)
RLE	**85.36**(0.50)	**88.74**(0.13)	**92.72**(0.12)
MCFS [3]	**85.50**(0.56)	86.70(0.32)	87.35(0.25)
RMCFS	85.34(0.29)	**86.73**(0.38)	**89.44**(0.24)

Table II: Comparisons on for MINST handwritten digits data (mean ± std). Fix the number of reduced dimensionality as the optimal value and vary the number of samples.

methods	4K samples	6K samples	8K samples
LLE [1]	24.77(0.57)	**26.16**(0.40)	26.13(0.42)
RLLE	**24.97**(0.55)	26.08(0.39)	**26.46**(0.50)
LE [2]	19.29(0.37)	22.12(0.41)	22.38(0.35)
RLE	**21.03**(0.44)	**23.90**(0.30)	**25.00**(0.29)
MCFS	27.30(0.48)	29.13(0.63)	29.52(0.49)
RMCFS [3]	**27.82**(0.31)	**29.80**(0.44)	**29.59**(0.47)

Table III: Comparisons on CIFAR 10 data (classification rate mean ± std). Fix the number of reduced dimensionality as the optimal value and vary the number of samples.

methods	All samples
LLE [1]	92.00(0.77)
RLLE	**92.30**(0.82)
LE [2]	79.10(1.22)
RLE	**79.75**(1.72)
MCFS [3]	88.00(1.03)
RMCFS	**95.00**(1.31)

Table IV: Comparisons on ORL data (classification rate mean $\pm$ std). The number of reduced dimensionality as the optimal.

digits widely used for training and testing in the field of machine learning. For MINST database our training data set is derived from the original MNIST database with 10 classes having 1000 20$\times$20 examples. The CIFAR 10 [3] dataset consists of 60000 32x32 colour images belonging 10 classes, with 6000 images per class. We employ a random forests (RF) as our classifier. In this paper, the hyper-parameters of classifier is are not varied. To eliminate the stochastic effects of RF, we average the accuracy over 10 trials. We employ six different methods to do validations. These are a) Local linearly embedding (LLE), b) Laplacian Eigenmaps (LE), c) Unsupervised Feature Selection for Multi-cluster Data (MCFS) and their corresponding NRM variant, denoted as RLLE, RLE, and RMCFS. And RLLE is LLE reinforced with Alg 1, RLE and RMCFS are reinforced with Alg 2.

2) Results: **Face recognition:** we conduct face recognition on the ORL face database and achieve an improvement of up to 7% as detailed in Table IV

Digit recognition: we perform classification on the MINST digital database. For each method, we vary the number of samples leading to three groups for comparison. Finally we achieve an improvement of up to 2.4%, as detailed in Table II

Object recognition: we perform classification task on the CIFAR 10 object database. For each method, we vary the number of samples leading to three groups for comparison. Finally we obtain an improvement of up to 2.6%, as detailed in Table III

V. Conclusion

In this paper we present a novel and compact neighbor reconstruction method(NRM) for graph based sparse spectral algorithms. Through manipulations on anchored points and corresponding neighborhoods, NRM can reconstruct new points which are closer to central point on the assumed manifold. Though a theoretical analysis and experiments on different tasks, we validate the benefits of NRM. In this paper NRM use a unify truncation number c which is easy for fine-tuning but not favorable for our theoretical analysis. In future, we wish to learning different c values in reconstruction operator R by some machine learning technologies.

Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant No.61402389,No.11671337), the Fundamental Research Funds for the Central Universities in China (no. 20720160073), the Natural Science Foundation of Fujian Province of China (No. 2016J01035) and Health joint fund of the Provincial Department of science and technology (No.2015J01534).

[3]http://www.cs.toronto.edu/ kriz/cifar.html

References

[1] Roweis, Sam T. and Saul, Lawrence K. *Nonlinear Dimensionality Reduction by Locally Linear Embedding*, Science, 2000.

[2] Belkin, Mikhail and Niyogi, Partha. *Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering*, Advances in Neural Information Processing Systems, 2002.

[3] Cai, Deng and Zhang, Chiyuan and He, Xiaofei. *Unsupervised feature selection for Multi-Cluster data*,ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010.

[4] Nie, Feiping and Zhu, Wei and Li, Xuelong. *Unsupervised feature selection with structured graph optimization*, Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[5] Timofte, Radu and De Smet, Vincent and Van Gool, Luc. *A+: Adjusted anchored neighborhood regression for fast super-resolution*, Asian Conference on Computer Vision, 2014.

[6] Webb, Andrew R and Copsey, Keith D. *Introduction to Statistical Pattern Recognition*, John Wiley&Sons, Ltd, 1990.

[7] Pudil, Pavel and Novovicova, Jana. *Novel Methods for Feature Subset Selection with Respect to Problem Knowledge*, IEEE Intelligent Systems, 1998.

[8] Tu, Minh Phuong and Lin, Z and Altman, R. B. *Choosing SNPs using feature selection*, Computational Systems Bioinformatics Conference, 2005.

[9] Broadhurst, David and Goodacre, Royston and Jones, Alun and Rowland, Jem J and Kell, Douglas B. *Genetic algorithms as a method for variable selection in multiple linear regression and partial least squares regression, with applications to pyrolysis mass spectrometry*, Analytica Chimica Acta, 1997.

[10] Zhang, Yudong and Wang, Shuihua and Phillips, Preetha and Ji, Genlin. *Binary PSO with mutation operator for feature selection using decision tree applied to spam detection*, Knowledge-Based Systems, 2014.

[11] Duval and Hao, Jin Kao and Hernandez, Jose Crispin Hernandez. *A memetic algorithm for gene selection and molecular classification of cancer*, Genetic and Evolutionary Computation Conference, 2009.

[12] Jirapech-Umpai, Thanyaluk and Aitken, Stuart. *Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes*, BMC Bioinformatics, 2005.

[13] Zhang, Yudong and Dong, Zhengchao and Phillips, Preetha and Wang, Shuihua and Ji, Genlin and Yang, Jiquan and Yuan, Ti Fei. *Detection of subjects and brain regions related to Alzheimer's disease using 3D MRI scans based on eigenbrain and machine learning*, Frontiers in Computational Neuroscience, 2015.

[14] Roffo, Giorgio and Cristani, Marco. *Infinite Feature Selection*, IEEE International Conference on Computer Vision, 2015.

[15] Roffo, Giorgio and Melzi, Simone. *Feature Selection via Eigenvector Centrality*, New Frontiers in Mining Complex Patterns in Conjunction with Ecml, 2016.

[16] Hernandez, Jose Crispin Hernandez and Duval, Batrice and Hao, Jin Kao. *A Genetic Embedded Approach for Gene Selection and Classification of Microarray Data*, European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, 2007.

[17] Turk, M. A and Pentland, A. P. *Face recognition using eigenfaces*, Computer Vision and Pattern Recognition, 1991.

[18] He, Xiaofei and Yan, Shuicheng and Hu, Yuxiao and Niyogi, Partha and Zhang, Hong Jiang. *Face recognition using laplacianfaces*, 2005.

[19] Eigenfacesvs, By. *Fisherfaces: Recognition Using Class Specific Linear Projection*, IEEETrans. on PAMI, 2013.

[20] Tenenbaum, J. B. and De, Silva V and Langford, J. C. *A global geometric framework for nonlinear dimensionality reduction*, Science, 2000.

[21] Donoho, David L. and Grimes, Carrie. *Hessian Eigenmaps: Locally Linear Embedding Techniques for High-Dimensional Data*, Science, 2000.

[22] Zhang, Zhen Yue and Zha, Hong Yuan. *Principal manifolds and nonlinear dimensionality reduction via tangent space alignment*, Journal of Shanghai University (English Edition), 2004.