Robust Faces Manifold Modeling: Most Expressive Vs. Most Sparse Criterion

Xiaoyang Tan Lishan Qiao Wenjuan Gao Jun Liu
Department of Computer Science and Engineering
Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China
{x.tan, l.qiao, w.gao, j.liu}@nuaa.edu.cn

Abstract

Robust face image modeling under uncontrolled conditions is crucial for the current face recognition systems in practice. One approach is to seek a compact representation of the given image set which encodes the intrinsic lower dimensional manifold of them. Among others, Local Linear Embedding (LLE) is one of the most popular method for that purpose. However, it suffers from the following problems when used for face modeling: 1) it is not robust under uncontrolled conditions (e.g., the underlying images may contain large appearance distortions such as partial occlusion or extreme illumination variations); 2) a fixed neighborhood size is used for all the local patches without considering the actual distribution of samples in the input space; 3) the modeled local structures may not contain enough discriminative information which is essential to the later recognition stage. In this paper, we introduce the Sparse Locally Linear Embedding (SLLE) to address these issues. By replacing the most-expressive *type criterion in modeling local patches in LLE with a* most-sparse *one, SLLE essentially finds and models more discriminative patches. This gives higher model flexibility in the sense of less sensitiveness to incorrect model and higher robustness to outliers. The feasibility and effectiveness of the proposed method is verified with encouraging results on a publicly available face database.*

1. Introduction

In practice, it is usually necessary to observe and analyze the potential distribution of very high-dimensional data (e.g, face images or web pages). Dimensionality Reduction (DR) approach is frequently used for that purpose by mapping the high-dimensional data onto another (usually low-dimensional) space while preserving the essential information contained in the original space. Classical methods for this include PCA ,ICA, LDA, LPP[9], NPE[8] and so on. The transformations defined by these, however, are linear in nature, while other DR methods use nonlinear transforms such as SOM [11], LLE [17][19],HLLE[6],LTSA[25] and ISOMAP [21], giving more flexibility in data modeling.

In machine learning literatures, those methods whose transformations (either linearly or nonlinearly) have extra capability to preserve the spatial consistency between input space and output space are usually called manifold methods – most of the aforementioned nonlinear DR methods (e.g., SOM[11], LLE[17], ISOMAP[21]) belong to this category. Among others, the LLE (Locally Linear Embedding [17]) method, which provides a closed-form (hence efficient) solution to extract the intrinsic lower dimensional manifold for a given data set in high dimensional space, is chosen to demonstrate the idea of this paper. The LLE method starts by finding the local structures of high dimensional sample space in terms of local patches[1], each of which is then modeled with a linear combination of a set of neighborhood points. Note that this step is commonly found in many other manifold methods such as HLLE[6] and LPP [9] and is crucial for the performance of these algorithms. The local patches can be regarded as a way to characterize the local spatial distribution of sample space and hence contain essential regularity of it. By employing such regularity, one may find a way to obtain a compact representation of original data and it is at this point that different manifold methods begin to diverge. In LLE, this is done with a MDS[4]-like method which arranges the local patches in lower-dimensional space with global consistency. The major advantage of LLE lies in its conceptually simplicity and its efficiency and effectiveness for many practical applications such as face images modeling, an important task in face recognition[20].

However, there are also some drawbacks of LLE which are not well-addressed: 1), LLE uses a *most-expressive* criterion (i.e., least squares) to model local patches, assuming

[1]Note that here the term *patch* refers to a set of points close to each other, rather than a local region in a image.

that good local patches are available everywhere in the input space and can be found with a k-nearest neighbor classifier (kNN) with a fixed k value. Unfortunately, real world data is usually distributed in a sparse and non-uniform manner in the high dimensional space, hence the samples constituting a patch are unlikely distributed in a perfect small region; 2), the aforementioned mechanism of LLE to find and model the local structures also tends to be sensitive to outliers such as face images whose appearance is largely distorted by extreme illumination changes or partial occlusions. The existence of such unwelcome data would impose great challenges on both the kNN local patch finder and the least square regression-based patch encoder - both of them are key components of LLE but are not robust by themselves. Previously this problem was addressed by using some robust statistical technique to "filter out" the noise neighbors before graph embedding [3]; 3) last but importantly, the *most-expressive* type neighborhood relationship supposed to be preserved by LLE embedding is not very suitable for the task of classification since this may actually enlarge the within-class variations when the samples from different classes are overlapping (c.f., Fig. 1, bottom right). There are some work [16] trying to improve this but under a rather strong assumption that class labels of samples are given, preventing it from employing a large amount of cheap unlabeled data.

In this paper, we propose an improved LLE method named SLLE (Sparse Locally Linear Embedding) to address these issues. The idea of SLLE is to replacing the *most-expressive* type criterion of modeling local patches in LLE with a *most-sparse* one. The sparse representation is well known in the literatures of signal representation theory [1] and is regarded as an important way to encode the domain knowledge, thus being helpful to improve the generalization capability of a given model [23]. To obtain a sparse patch representation, we merge the procedure of patch finding and patch modeling into one single step, which is equivalent to solving a standard linear programming problem with $L1$ penalty. While this can be efficiently solved using many existing optimization packages, we show in this paper that this simple modification also leads to several benefits: 1) Due to the nature of sparseness constraints, local patches with higher discriminative capability are naturally found and compactly modeled (c.f., Fig. 1), which is essential for good recognition performance even when data labels are absent; 2) the modeling of local structures becomes more robust against outliers and more adaptive to complex nonlinear data; 3) since the tasks of patch finding and patch modeling is done in one single step instead of being treated separately, our algorithm is more conceptually clear than LLE; and 4) the idea of modeling *most-sparse* patches instead of *most-expressive* ones is quite general and can be naturally extended to other similar manifold methods like LPP[9], HLLE[6], and LTSA[25].

In a recent independent study, Yan and Wang [24] proposed a similar method to ours, where they treat the L1-minimization based graphic construction as a parameter-free method for semi-supervised learning. However, we note that the degree of sparsity may have significant influence on the performance of the algorithm (*c.f.* Fig. 6(a) left). More importantly, the L1 minimization based objective function may not always lead to correct solutions, and we propose a method with gaussian matrix transformation to address this problem.

The paper is organized as follows: in section 2, the original LLE algorithm is briefly reviewed and then we introduce our SLLE method in section 3. In section 4 we present the experimental results and conclude this paper in Section 5.

2. Local Linear Embedding

Suppose that there are N samples $y_i, i = 1, .., N, y_i \in R^D$, from a smooth manifold in the high dimensional space. LLE assumes that in the local sense, a point is always located on a hyper-plane, so that any point can be expressed by a combination of its neighbor points. In the low-dimensional space, LLE preserves the neighborhood relations learned in the high-dimensional space through keeping the above linear coefficients.

In particular, LLE first finds N local patches by computing k (a *fixed* value through all patches) neighbors for every sample. Then the objective function of reconstruction is

$$\varepsilon(W) = \sum_{i=1}^{N} \sum_{j=1}^{k} \|y_i - \sum_{j,j\neq i} W_{ij} y_j\|^2 \qquad (1)$$

i.e., the neighborhood reconstruction error $\varepsilon(W)$ is defined to be the sum of difference between each point and its neighbors over the training set. The weight W_{ij} represents the reconstruction contribution to the *i*-th point of the *j*-th point and if y_j is not among the k neighborhood of the y_i, $W_{ij} = 0$. In order to maintain the translation invariance, $\sum_j W_{ij} = 1$ should be met. The reconstruction matrix W are sought by minimizing the error function $\varepsilon(W)$, which is a standard least squares regression problem and can be solved with gradient descent method.

Now we need a mapping to transform all the high-dimensional data to a low-dimensional manifold in a nonlinear manner, with the requirement that the spatial consistency between the two spaces should be preserved. One simple way to do this is to align each local patch within another global coordinate system with lower dimensionality. Mathematically, the alignment can be achieved by seeking a nonlinear mapping with constraints that the local models learned in the high dimensional space are invariant to some particular affine transformations on points within each patch, such as translation, scaling and rotating. Actually

this is also the only hint LLE uses to find the locations of the high dimensional points in the output space. In particular, denote each image of y_i as $x_i \in R^d$, where d ($d < D$) is the dimensionality of the output space. Then the images $x_i, i = 1, ..., N$ can be obtained by minimizing the following loss function,

$$\varepsilon(x) = \sum_{i=1}^{N} \|x_i - \sum_j W_{ij} x_j\|^2 \quad (2)$$

where W is the reconstruction matrix learned in the input space. This boils down to solving a simple eigen problem for matrix $M = (I - W)(I - W)'$ (I is the identity matrix) [17],[19] and keeping the bottom d eigenvectors except the one with zero eigenvalue as the coordinates for the output space.

3. Sparse Local Linear Embedding

As mentioned before, due to the use of a fixed *k* value to model all the local areas, the traditional LLE becomes sensitive to noise and outliers when the input manifold is sparse and complicated. In this section, a local area modeling method based on sparse representation is introduced to overcome these problems. The idea is to adopt a *most-sparse* criterion instead of a *most-expressive* one to model the regions of interest, each of which consists of the smallest set of points (called support points here) best approximating the given prototype. One difference between SLLE and LLE is that to find the support points, we don't restrict the candidates in the range of k neighborhood but allow the algorithm to find them (and their weights) adaptively among the whole training set.

3.1. Sparse Representation

Suppose that the relationship between a face image $y \in R^D$ and a set of training images A could be modeled with a linear model $y = Aw$ [23], where A is a matrix consisting of N column-wise training vectors and w is the linear combination vector of interest. Since the dimensionality of face data is usually far greater than the number of samples (i.e., $D > N$), the linear equation $y = Aw$ has no exact solution. But an approximate solution can be sought by projecting y into the column space of A using the least-squares method:

$$(l^2) : \hat{w}_{l2} = \arg\min_w \|y - Aw\|^2 \quad (3)$$

The solution of the above equation is generally very dense. In LLE, this issue is addressed by manually restricting the range of A. Alternatively, one can incorporate prior knowledge in terms of regularization, by which the sparsity of w is explicitly imposed. For example, the l_0 pseudo-norm term encourages a solution with minimal number of non-zero elements, i.e.,

$$(l^0) : \hat{w}_{l0} = \arg\min_w \|y - Aw\|_2^2 + \lambda \|w\|_0 \quad (4)$$

where $\|w\|_0$ denotes l_0 pseudo-norm of the vector w, and λ is the regularization parameter. Note that once solved, (4) provides both the support points and the corresponding weights for y. Unfortunately, this is a difficult task as solving (4) is NP-hard [23].

According to the sparse representation theory [2],[5], if y can be represented sparsely enough by the samples in *A*, the above l_0 minimization problem is strictly equivalent to its l_1-norm counterpart,

$$(l^1) : \hat{w}_{l_1} = \arg\min_w \|y - Aw\|_2^2 + \lambda \|w\|_1 \quad (5)$$

where $\|w\|_1$ is usually called lasso penalty in the statistical literatures [18] and makes the solution nonlinear for y. Lots of packages are available (e.g., L1-Magic) to solve this quadratic programming problem. Moreover, from a Bayesian perspective, Eq.(5) encodes our prior belief w should be sparse in the basis of A and solving it gives a posterior belief for the values of w [18],[15],[22].

In experiments, however, we've found that the solution of (5) may not be stable - sometimes we simply cannot find the needed sparse solutions for certain images. This suggests that for real world data, the base matrix (A) may not always meet the sparse conditions[1, 2], which states that, to ensure that Eq.(5) has a sparse solution, A must conform to the so-called RIP (restricted isometry property,[2]) condition. Unfortunately, the computational complexity involved in verifying the RIP of any matrix is too high.

There are several methods to address this problem. For example, based on the observation that a Gaussian random matrix can always meet the conditions [1, 2], one can first transform the original image base A using a $D \times D$ Gaussian random matrix Φ to a new random matrix Θ before solving the optimization:

$$\begin{aligned} (l^{1*}) : \hat{w}_{l_1} &= \arg\min_w \|\Phi y - \Phi A w\|_2^2 + \lambda \|w\|_1 \\ &= \arg\min_w \|y' - \Theta w\|_2^2 + \lambda \|w\|_1 \end{aligned} \quad (6)$$

where $y' = \Phi y$ and $\Theta = \Phi A$. Since Φ is invertible, the solution $\hat{w}'_{l1}$ of (6) is an approximation of $\hat{w}_{l1}$ obtained from the original image base *A*. In particular, if $\Phi y = \Phi A w$, then $\Phi^{-1}\Phi y = \Phi^{-1}\Phi A w$, which gives $y = Aw$, i.e., the solution between (5) and (6) is exactly identical. When $\Phi y = \Phi A w + \varepsilon$, where ε is a approximating error vector with zero mean and fixed variance, we have $y = Aw + \Phi^{-1}\varepsilon$ and hence $\|y - Aw\|^2 = \|\Phi^{-1}\varepsilon\|^2$. In practice, we are actually more interested in the later case since face images are always distorted by some appearance changes ε.

Finally, we change the form of our objective function (6) as follows to facilitate the computation,

$$(l^{1**}) : \hat{w}_{l_1} = \arg\min_{w} \|y' - \Theta w\|_2^2 \quad (7)$$

$$s.t. \|w\|_1 \leq \tau \quad (8)$$

and after the w is solved, we scale them so that $1 = \mathbf{1}^T w$. The degree of sparseness of the solution is controlled by τ, which has a one to one correspondence with λ in (6) and can be set using cross validation technique [7]. To solve (7), we use a recently designed package named SLEP [13], which combines Nesterov's first-order black-box optimal method with an efficient Euclidean projection (in linear time) [12] for fast convergence.

3.2. Sparse Locally linear Embedding

By solving (7), we obtain a sparse representation of a given point y, i.e., most elements of w vector will be zeros. To exploit this advantage, we take the set of vectors in A whose corresponding weight in w is not zero as the support points of y, and naturally these corresponding nonzero weights model the contribution of each support point to the local model indexed by y. In other words, simply by solving (6), we get both the needed local regions *and* their models, which is contrary to the scheme of LLE, where local regions and their models are estimated separately.

In particular, for each face image $y_i \in R^D$, define $A_{/i} \triangleq [y_1, y_2, ..., y_{i-1}, y_{i+1}, ..., y_N]$. That is, the matrix $A_{/i}$ is composed of all of the face images except the vector y_i. Then for each face image y_i, we calculate its sparse representation with $A_{/i}$, and denote it as $\hat{W}_i = [W_{i,1}, W_{i,2}, ..., W_{i,(i-1)}, W_{i,(i+1)}, ..., W_{i,N}]^T$. Next, those points in $A_{/i}$ with non-zero weight are chosen to be the support points of y_i. Note that the patch defined in this way is adaptively emerged from our sparse constraints. Figure 1 gives an illustration of support points found in this way and those by LLE.

The following step is the same as that in LLE, i.e., projecting the points in high dimensional space into the lower dimensional manifold [2]. But before that we need to confirm ourselves that the weight matrix obtained in our method is invariant to certain affine transformation (translation, rotation and scale) as well: 1) denote the translation vector as $t \in R^{D\times 1}$, then for any point $y_i \in R^D$, we have $y_i + t \approx \sum_j w_j y_j + t = \sum_j w_j y_j + \sum_j w_j t = \sum_j (y_j + t)$, for $j \neq i$, and the constraint of $\sum_j w_j = 1$ is used; 2)similarly, for any rotation and scaling matrix *R*, we have: $Ry_i \approx R\sum_j w_j y_j = \sum_j w_j R y_j = \sum_j w_j (R y_j)$. In summary, if the reconstruction error of y_i by $\sum_j w_j y_j$ is relatively small, the weight matrix is approximately unchanged when we translate, zoom or rotate the samples in a patch. The complete SLLE algorithm is summarized in Table 1[3].

3.3. The 'Out-of-Sample' Problem

In order to generalize an unseen sample, say y_{test}, we first calculate its sparse representation $\hat{w}_{test}$ using the training set, then fix this weight vector and project the unseen sample into the d-dimensional space by solving:

$$\varepsilon(x_{test}) = \|x_{test} - X\hat{w}_{test}\|_2 \quad (9)$$

where X is the $d \times N$ matrix consisting of projections of samples in input space and x_{test} is corresponding coordinate in the output space for y_{test}. The same strategy has been adopted in [19].

4. Experiments

We evaluate the effectiveness of the proposed algorithm on the AR database[14], which contains over 4,000 color face images of 126 people's faces (70 males and 56 females), including frontal view faces with different facial expressions, illumination conditions, and occlusions (with sun glasses and scarf). There are 26 different images per person, taken in two sessions (separated by two weeks), each session consisting of 13 images.

4.1. Robust Face image Modeling

To verify the robustness of SLLE against large appearance changes such as partial occlusions, we simulate a set of faces gradually occluded by sunglasses using linear interpolation, as shown in Fig. 2, where the image size is scaled to be 66×48 pixels (thus D=3168). This results in 50 virtual images for a given subject and we also add another 5 random images as noise. We then model these face images using LLE (with k=6) and SLLE, respectively. The dimensionality of output space is set to be 3 in order to make it feasible for visual examination of the face manifold. Fig. 3 gives the result. The figure clearly reveals that SLLE is able to find good smooth manifold while leaving the outliers away from it. On the contrary, the manifold produced by the LLE is not smooth.

To verify the 'out-of-sample' ability of our method, we project ten more new samples onto the manifold learned by SLLE in the previous experiment, and check their locations in the low dimensional global coordinate (marked as '*' symbol in Fig. 4). As the figure shows, these unseen samples are perfectly embedded onto the manifold without being distracted by the noise images.

[2] In the implementation of this step, we regularize the M matrix (c.f., Section 2, also see [19]) by a small-valued constant matrix (i.e., ε I with $\varepsilon = 0.001$) before doing eigen decomposition, which is usually not necessary but we have found that this could generally stabilize the solution and is beneficial to the performance.

[3] A Matlab implementation of SLLE is available at: http://parnec.nuaa.edu.cn/xtan/paper/SLLEdemo.zip

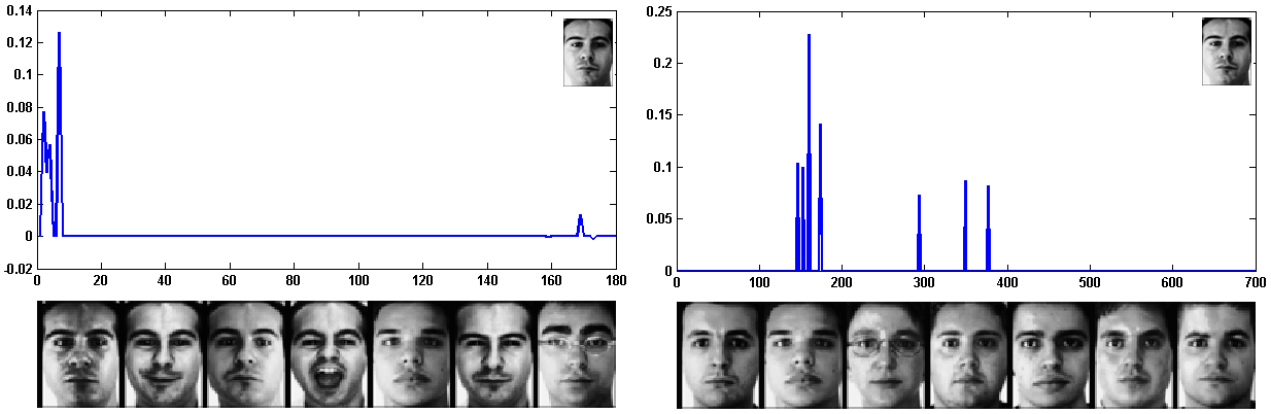


Figure 1. Illustration of seven support faces (bottom row) for a given face image (left top) and the distribution of corresponding sparse weights (upper row) using SLLE (left) and LLE (right), respectively. Note that the support faces found by SLLE with a *most sparse* criterion are more discriminative than those by LLE with a *most expressive* criterion - five of seven faces with the same identity as the prototype are correctly included in a SLLE patch.

Table 1. the SLLE algorithm(Sparse Linear Embedding)

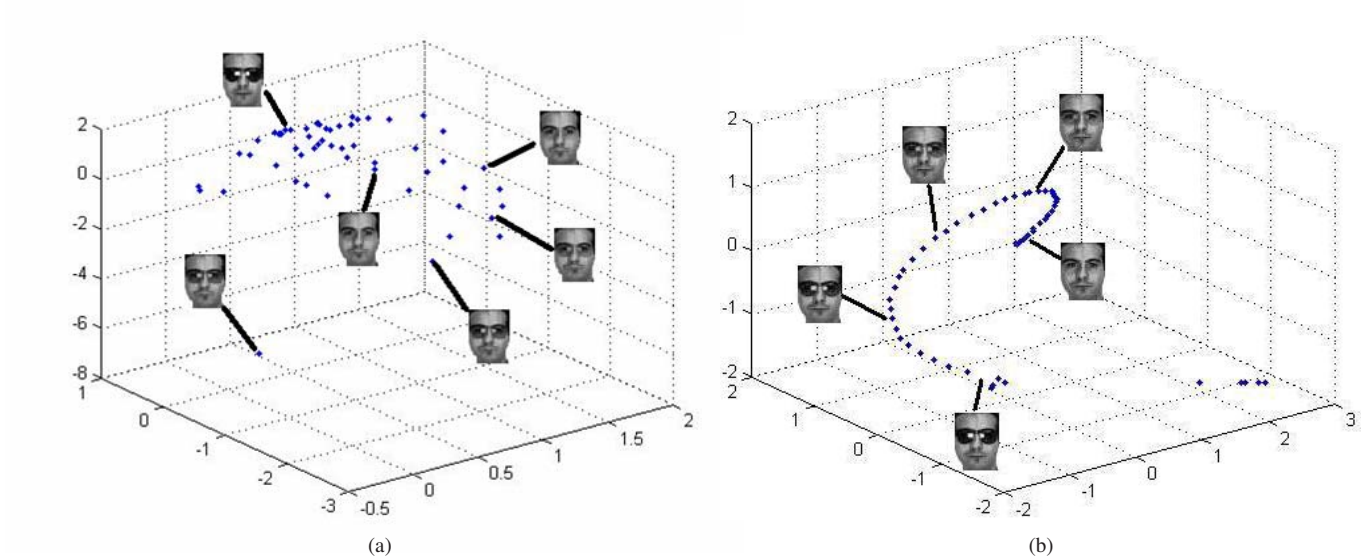
The SLLE algorithm (Sparse Locally Linear Embedding)
1: Input: N samples $A = [y_1, y_2, ..., y_N] \in R^{D\times N}$, error ε, and the dimensionality of output space d;
2: Normalize all the columns of A to have unit L_2 norm; (optional) Generate a Gaussian random array $\Phi \in R^{D\times D}$, and let $A' = \Phi A$, $y' = \Phi y$;
3:Let $\hat{A}_{/i} = [y'_1, ..., y'_{i-1}, y'_{i+1}, ..., y'_N] \in R^{D\times N}$;
3.1) Solve the minimizing L_1 norm problem Eq.(7) to obtain a weight vector w_i for each sample y'_i; ;
3.2) Set $w_i = [w_{i,1}, w_{i,2}, ..., w_{i,(i-1)}, 0, w_{i,(i+1)}, ..., w_{i,N}]$;
4:Repeat step 3 for each sample;
5:Solve $\varepsilon(X) = \sum_i \|x_i - \sum_j w_{ij}x_j\|_2^2$ to get the low dimensional representation of each face image in A.

Figure 2. An illustration of simulated training samples with sunglasses.

4.2. Recognition on the face Manifold

To illustrate the benefits of *most sparse* criterion in the task of pattern recognition with SLLE as an unsupervised feature extractor, we conduct another series of experiments on the AR dataset. In particular, the first 7 images in the first session of each subject are used for training and those in the second session for testing. Hence we have 1400 images covering various variations from expression, illustration and time changes (c.f., Fig. 5). Although many advanced classifiers such as SVM can be used, here we adopt a simple nearest neighbor classifier with cosine distance for recognition.

We first investigate the functional relationship between recognition rates and the degree of sparseness, which is done by testing the recognition performance using a validation set with various values of τ (c.f. Eq.(7)). A similar procedure was performed for LLE to choose the best k value. Note that although we need to experimentally set a suitable parameter value for τ (as for k in LLE), its meaning is quite different from the parameter k in LLE - even a fixed τ would generally lead to different patch size for every sample in the input space. To give a concrete correspondence between the value of τ and the degree of sparseness on this dataset, we also evaluate the degree of sparseness at



(a) (b)

Figure 3. The face manifold respectively modeled with LLE and SLLE on AR with partial occlusions when there exists noise: (a) LLE and (b) SLLE.

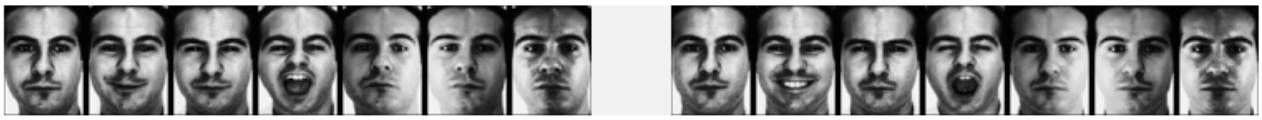


Figure 5. Some AR samples from one subject. Left: training images; Right: test images.

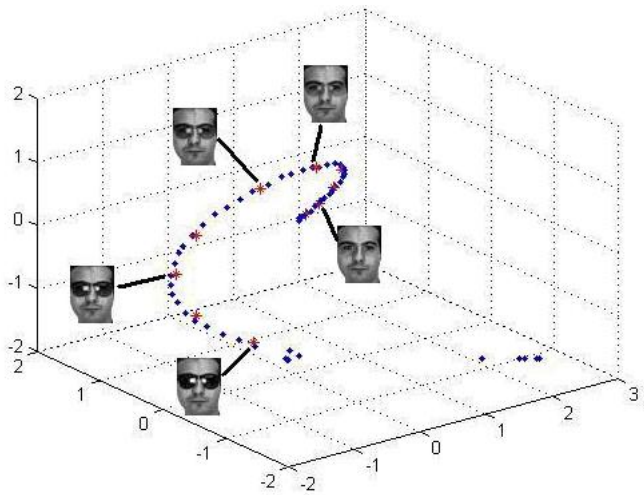


Figure 4. Illustration of the 'Out-of-Sample' capability of SLLE: unseen face images are well embedded into the previously learned manifold (blue points - training data, red * symbol denotes the location of the new face images)

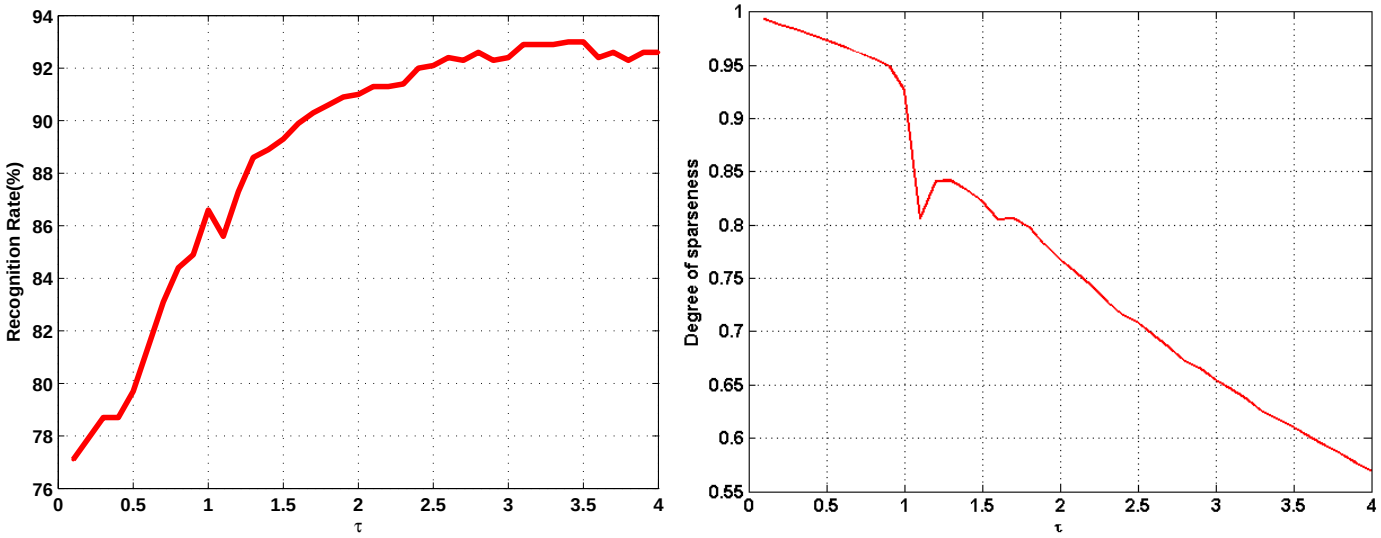
a grid of values of τ between [0,4], following [10],

$$sparseness(w) = \frac{\sqrt{N} - (\sum |w_i|)/(\sqrt{\sum w_i^2})}{\sqrt{N} - 1} \qquad (10)$$

where N is the dimensionality of w. This function equals unity if and only if w contains only a single non-zero component, and takes a value of zero if and only if all components are equal.

Fig. 6(a) gives the functional relationship between recognition performance and the value of τ and Fig. 6(b) gives the corresponding degree of sparseness for each value of τ. Although these figures reveal that a range of τ values higher than 2.5 is preferred, it remains unclear why setting a looser bound on $\|w\|_1$ (usually yielding larger patch size) is better than a tighter one in SLLE - for LLE, a k value higher than 10 has little influence on performance on this dataset.

Then we make a close comparison between our method and LLE, PCA under different feature dimension (settings: for SLLE, $\tau = 3.4$ (degree of sparseness = 0.6) and for LLE, $k = 10$). Fig. 7 shows the experimental results. Fig. 7 reveals that LLE only performs marginally better than PCA when the feature dimension is larger than 200 (corresponding to about 97% PCA energy), while its performance is much worse than PCA when the feature dimension is below 200. Considering that no spatial relationship between samples are preserved in PCA while local structures of sample space are explicitly modeled in LLE, these results are somewhat surprising. One possible explanation is that the *most-expressive* criterion actually *enlarge* the within-class variation in complicated situation - by inspecting Fig. 1 (bottom right), one can find that the resulting support faces of LLE for a given face have similar illumination variations to the prototype but their identities are quite different. By



(a) (b)

Figure 6. Experimental results on AR: (a) Recognition rates under different degree of sparseness value τ in SLLE; (b) The correspondence between the degree of sparseness and τ.

contrast, the support faces selected by SLLE using *most-sparse* criterion contain more images with the same identity as that of the face to be modeled (c.f., Fig. 1 bottom left). This leads to the remarkable performance improvement of SLLE - on average over 15% better than that of LLE. Furthermore, Fig. 7 reveals that the performance of SLLE begins to converge at low dimension of 67, compared to 100 for PCA and 200 for LLE, respectively. We also compare our method with other popular manifold methods besides a completely supervised method (FLDA [7]) and achieve favorable results (Table 2) consistently. From these, we conclude that the *most sparse* criterion is useful to extract discriminative features for an *unsupervised* learning algorithm such as LLE to improve its performance in a classification task.

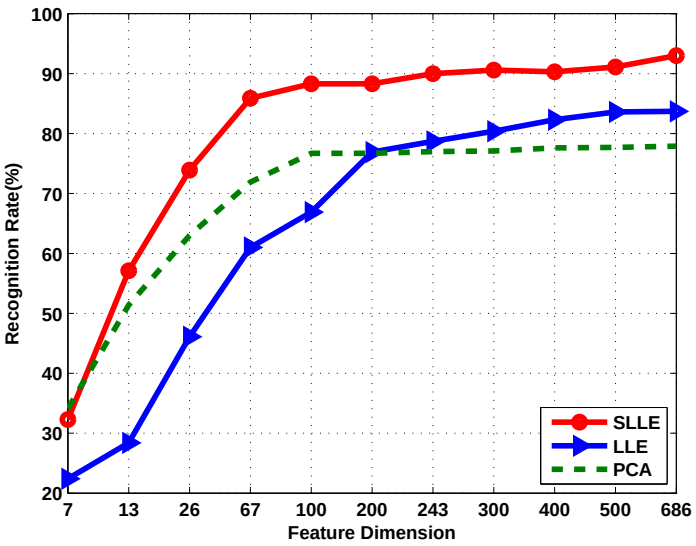


Figure 7. Comparative recognition rates with different feature dimension for various methods on the AR dataset.

5. Conclusion

In this paper, a robust face manifold embedding method named SLLE is proposed. The method adopts a *most-sparse* criterion rather than a *most-expressive* one to model the local spatial structures in high dimensional space. We show that this simple modification increases the discriminative power of traditional LLE algorithm when data labels are absent, making it behave more flexibly and more robustly when dealing with complex data such as face images. Note that since finding local patches is crucial and common in many manifold methods such as LPP[9], NPE[8], HLLE[6], and LTSA[25], the extension to these methods is straightforward.

Acknowledgments

This research was supported by the National Science Foundation of China (60773060), the Jiangsu Science Foundation (BK200922660) and the Project sponsored by SRF for ROCS, SEM.

References

[1] R. Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24(4):118–121, 2007.

Table 2. Comparison of best recognition rates (%) on AR over various methods.

Methods	PCA	FLDA	LPP[9]	NPE[8]	LLE[19]	SLLE
Accuracy	77.9	83.7	78.4	79.4	83.7	**93.0**

[2] E. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies. *IEEE Trans. Information Theory*, 52(12):5406–5425, 2006.

[3] H. Chang and D.-Y. Yeung. Robust locally linear embedding. *Pattern Recognition*, 39(6):1053–1065, 2006.

[4] T. Cox and M. Cox. *Multidimensional scaling*. Chapman and Hall, 2001.

[5] D.Donoho. For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution. *Comm. on Pure & Applied Math*, 59(6):797–829, 2006.

[6] D. Donoho and Grimes. Hessian eigenmaps locally linear embedding techniques for high-dimensional data. *Proc. the National Academy of Sciences*, 105(48):5591–5596, 2003.

[7] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2009.

[8] X. He, D. Cai, S. Yan, and H. Zhang. Neighborhood preserving embedding. In *ICCV*, pages 1208–1213, 2005.

[9] X. He, X. Yan, Y. Hu, P. Niyogi, and H. J. Zhang. Face recognition using Laplacianfaces. *IEEE TPMI*, 27(3):328–340, 2005.

[10] P. O. Hoyer and P. Dayan. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[11] T. Kohonen. *Self-Organizing Map*. Springer, Berlin, 2nd edition, 1997.

[12] J. Liu and J. Ye. Efficient euclidean projections in linear time. In *ICML*, 2009.

[13] J. Liu and J. Ye. Slep: Sparse learning with efficient projections. *Arizona State University*, 2009.

[14] M. Martinez and R. Benavente. The AR face database. Technical Report 24, CVC, 1998.

[15] M.Figueiredo. Adaptive sparseness for supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1150–1159, 2003.

[16] D. Ridder and R. Duin. Locally linear embedding for classification. Technical report, Pattern Recognition Group, Delft University of Technology.

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

[18] R.Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58(1):267–288, 1996.

[19] L. Saul and S. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4:119–155, 2003.

[20] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang. Face recognition from a single image per person: A survey. *Pattern Recognition*, 39(9):1725–1745, 2006.

[21] I. Tenenbaum, V. Silva, and J. Langford. A global geomentric framework for nonlinear dimensinality reduction. *Science*, 290(5500):2319–2322, 2000.

[22] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.

[23] J. Wright, A. Yang, and A. Ganesh. Robust face recognition via sparse representation. *IEEE TPAMI*, 31(2):210–227, 2009.

[24] S. Yan and H. Wang. Semi-supervised learning by sparse representation. In *SDM*, 2009.

[25] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *Scientific Computing*, 26(1):313–318, 2004.