Out-of-Sample Embedding for Manifold Learning Applied to Face Recognition

F. Dornaika [1,2]

[1] University of the Basque Country UPV/EHU, San Sebastian, SPAIN
[2] IKERBASQUE, Basque Foundation for Science, Bilbao, SPAIN

B. Raducanu
Computer Vision Center, Barcelona, SPAIN

Abstract

Manifold learning techniques are affected by two critical aspects: (i) the design of the adjacency graphs, and (ii) the embedding of new test data—the out-of-sample problem. For the first aspect, the proposed schemes were heuristically driven. For the second aspect, the difficulty resides in finding an accurate mapping that transfers unseen data samples into an existing manifold. Past works addressing these two aspects were heavily parametric in the sense that the optimal performance is only reached for a suitable parameter choice that should be known in advance.

In this paper, we demonstrate that sparse coding theory not only serves for automatic graph reconstruction as shown in recent works, but also represents an accurate alternative for out-of-sample embedding. Considering for a case study the Laplacian Eigenmaps, we applied our method to the face recognition problem. To evaluate the effectiveness of the proposed out-of-sample embedding, experiments are conducted using the k-nearest neighbor (KNN) and Kernel Support Vector Machines (KSVM) classifiers on four public face databases. The experimental results show that the proposed model is able to achieve high categorization effectiveness as well as high consistency with non-linear embeddings/manifolds obtained in batch modes.

1. Introduction

Manifold learning refers to the problem of recovering the structure of a manifold from a set of unordered sample data. Manifold learning is often equated with dimensionality reduction, where the goal is to find an embedding or unrolling of the manifold into a lower dimensional space such that certain relationships between samples are preserved. Such embeddings are typically used for visualization. In recent years, a new family of non-linear dimensionality reduction techniques for manifold learning has emerged. The most

known ones are: Kernel Principal Component Analysis (KPCA) [22], Locally Linear Embedding (LLE) [19, 20], Isomap [23], Supervised Isomap [11], Laplacian Eigenmaps (LE)[1]. The non-linear methods such as Locally Linear Embedding (LLE), Laplacian Eigenmaps, Isomap, Hessian LLE (hLLE) [8] focus on preserving the local structure of data. LLE formulates the manifold learning problem as a neighborhood-preserving embedding, which learns the global structure by exploiting the local symmetries of linear reconstructions. Isomap extends the classical Multidimensional Scaling (MDS) [3] by computing the pairwise distances in the geodesic space of the manifold. Essentially, Isomap attempts to preserve geodesic distances when data are embedded in the new low dimensional space. Based on the spectral decomposition of the graph Laplacian, Laplacian Eigenmaps actually try to find Laplacian eigenfunction on the manifold.

The main issues of the non-linear methods are: (1) the quality of embedded space is very sensitive to the choice of free parameters used in the data graph construction [26], and (2) they do not provide an explicit mapping function between low and high dimensional spaces [18]. Such function is essential for ensuring continuity of low dimensional representation and projecting data between spaces. Many existing manifold learning techniques do not naturally contain an out-of-sample extension so research has been undertaken to find ways of extending manifold learning techniques to handle new samples. The out-of-sample extension problem has not received much attention by researchers since it was considered as a pure non-linear regression problem [10, 27]. Therefore, the out-of-sample problem has been addressed quite satisfactorily by applying Radial Basis Function network to approximate the optimal mapping function [10]. However, the quality of Radial Basis Function network relies on the careful selection of a few parameters which are chosen empirically [21]. In [7], the author presented an algorithm, Locally Smooth Manifold Learning, for learning the structure of a manifold in terms of tangent

vectors. Rather than pose manifold learning as the problem of recovering an embedding, they posed the problem in terms of learning a warping function for traversing the manifold using the learned tangent vectors. Smoothness assumptions on this warp allowed the method generalize to unseen data. In [2], the authors cast MDS, ISOMAP, LLE, and LE in a common framework, in which these methods are seen as learning eigenfunctions of a kernel. The authors try to generalize the dimensionality reduction results for the unseen data samples. They also show how the width of the Gaussian can be tuned to achieve extrapolation.

In this paper, we address the out-of-sample extension problem. We adopt the sparse representation approach as an optimal solution to the 'out-of-sample' problem. The sparse representation was recently used as an effective alternative to the parametric construction of the adjacency graph [26]. Without any loss of generality, we chose the Laplacian Eigenmaps as one of the non-linear dimensionality reduction techniques to test our method. We present a generalized out-of-sample extension solution using the recent findings in sparse coding theory. Unlike existing approaches we do not require information to be retained from the learning process, such as the pairwise distance matrix or the resultant eigenvectors, we simply learn the mapping from the original high-dimensional data and its low-dimensional counterpart. Although the proposed method integrates the locality preserving principle in its derivation, it is intended to be independent of any specific manifold learning algorithm.

The paper is structured as follows. In section 2, we briefly review the Laplacian Eigenmaps as well as the L_1 graph construction. In section 3, we introduce our proposed approach for the out-of-sample problem based on sparse representation. Section 4 contains the experimental results performed on four face data sets. We evaluate the performance of the proposed out-of-sample method for the face recognition problem. Finally, in section 5 we present our conclusions.

2. Background

2.1. Review of Laplacian Eigenmaps

Laplacian Eigenmaps is a recent non-linear dimensionality reduction technique that aims to preserve the local structure of data [1]. Using the notion of the Laplacian of the graph, this non-supervised algorithm computes a low-dimensional representation of the data set by optimally preserving local neighborhood information in a certain sense. We assume that we have a set of N samples $\{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^D$. LE seeks latent points $\{\mathbf{y}_i\}_{i=1}^N \subset \mathbb{R}^L$ that minimize $\frac{1}{2}\sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij}$, which discourages placing far apart latent points that correspond to similar observed points. It can be shown in [1] that the matrix of embedded data $\mathbf{Y}$ can be obtained by solving a generalized eigenvalues problem.

The maximum number of eigenvectors (i.e., the dimensionality of the embedded space) is given by the number of the training samples.

2.2. Review of L1 graph reconstruction

In traditional graph construction process, the graph adjacency structure and the graph weights are derived separately. In [26], the authors argue that the graph adjacency structure and the graph weights are interrelated and should not be separated. Thus it is desired to develop a procedure which can simultaneously completes these two tasks within one step. Indeed, many experiments show that the recognition rate in the embedded space of LE obtained with a traditional graph construction can be highly depending on the choice of the neighborhood size in the reconstructed graph [25, 15]. Choosing the ideal size in advance can be a very difficult task.

The basic idea of [26] is to simultaneously estimate the graph adjacency structure and graph weights. To this end, every sample image is coded as a sparse linear combination of the rest of the training samples [9, 24]. This is carried out by implementing an L_1 minimization process to obtain the sparse representation of that sample as linear combination of the remaining training samples. The coefficients for the L_1 reconstruction reflect the relation among samples [14]. The spareness of the obtained coefficients encodes the graph structure, and the absolute values of the coefficients are the weights.
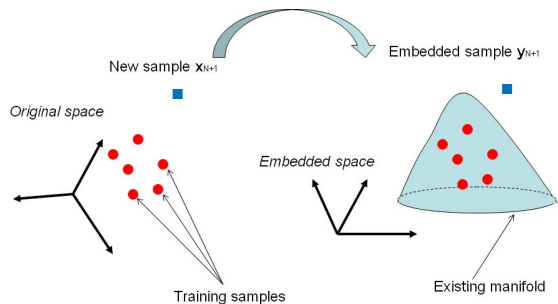

Figure 1. The out-of-sample problem consists in finding the embedding coordinate of an unseen sample.

3. Proposed out-of-sample embedding

In this section, we show that the theory of sparse representation (coding) can be used for solving the out-of-sample extension problem without relying on traditional heuristics that are usually parametric. For a case study, we use the Laplacian Eigenmaps for the non-linear embedding. The

reason of our choice is motivated by the fact that this transform is widely used by machine learning community for spectral clustering [16, 28].

3.1. Projection of new samples

Assume we have obtained an LE embedding $\mathbf{Y}_s = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$ of seen samples $\mathbf{X}_s = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ and consider unseen (out-of-sample) sample in observed space $\mathbf{x}_{N+1}$ (See Figure 1). The natural way to embed the new sample would be to recompute the whole embedding $(\mathbf{Y}_s, \mathbf{y}_{N+1})$ for $(\mathbf{X}_s, \mathbf{x}_{N+1})$. This is computationally costly and does not lead to defining a mapping for new samples; we seek a way of keeping the old embedding fixed and embed new sample based on that. Then, the next most natural way is to recompute the embedding but keeping the old embedded samples fixed and imposing that the embedding of the new sample (vector $\mathbf{y}_{N+1}$) should minimize the following target function:

$$\sum_{i=1}^{N} \|\mathbf{y}_{N+1} - \mathbf{y}_i\|^2 \, W_{(N+1)i} \tag{1}$$

$$= \sum_{i=1}^{N} (\mathbf{y}_{N+1} - \mathbf{y}_i)^T (\mathbf{y}_{N+1} - \mathbf{y}_i) W_{(N+1)i} \tag{2}$$

The above should correspond to a minimum, and thus the derivative with respect to $\mathbf{y}_{N+1}$ of the target function should vanish:

$$2 \sum_{i=1}^{N} (\mathbf{y}_{N+1} - \mathbf{y}_i) W_{(N+1)i} = 0 \tag{3}$$

From the above, we can conclude that the embedding $\mathbf{y}_{N+1}$ is given by:

$$\mathbf{y}_{N+1} = \frac{\sum_{i=1}^{N} W_{(N+1)i} \, \mathbf{y}_i}{\sum_{i=1}^{N} W_{(N+1)i}} \tag{4}$$

The above formula stipulates that the embedding of an unseen sample is simply the linear combination of all fixed embedded samples where the linear coefficients are set to the similarity between the unseen sample and the existing sample.

Whenever $W_{(N+1)i}$ is set to a Kernel function (i.e., $W_{(N+1)i} = K(\mathbf{x}_{N+1}, \mathbf{x}_i)$, Eq. (4) is equivalent to the Laplacian Eigenmaps Latent Variable Model (LELVM) introduced in [6].

3.2. Computation of the similarity coefficients via Sparse Representation

The problem of out-of-sample embedding boils down to the estimation of the similarities $W_{(N+1)i}, i = 1, \ldots, N$. In

[6], these $W_{(N+1)i}$ were computed using a K nearest neighbor and a Heat Kernel. However, it is well known that the neighborhood size as well as the Kernel parameter may affect the embedding process. We will bypass this limitation by using the coding provided by sparse representation.

We apply the sparse coding/representation principle for computing the set of coefficients $W_{(N+1)i}$ [14]. Let the vector $\mathbf{a} = (W_{(N+1)1}, W_{(N+1)2}, \ldots, W_{(N+1)N})^T$. Thus, the objective is to compute the vector $\mathbf{a}$ given the unseen sample and the training data. Based on sparse coding, the unseen sample $\mathbf{x}_{N+1}$ can be written as

$$\mathbf{x}_{N+1} = \sum_{i=1}^{N} a_i \, \mathbf{x}_i + \mathbf{e} = \mathbf{X} \, \mathbf{a} + \mathbf{e} \tag{5}$$

The goal is to minimize both the reconstruction error $\mathbf{e}$ and the L_1 norm of the vector $\mathbf{a}$:

$$\min_{\mathbf{a}, \mathbf{e}} \left(\|\mathbf{a}\|_{L_1} + \|\mathbf{e}\|_{L_1} \right) \quad s.t. \quad \mathbf{x}_{N+1} = \mathbf{X} \, \mathbf{a} + \mathbf{e} \tag{6}$$

Let $\mathbf{a}'$ denote the vector $\mathbf{a}' = (\mathbf{a}^T, \mathbf{e}^T)^T$ and $\mathbf{I}$ denote the $D \times D$ identity matrix, then the objective function (6) can be written as:

$$\min \|\mathbf{a}'\|_{L_1} \quad s.t. \quad [\mathbf{X} \; \mathbf{I}] \, \mathbf{a}' = \mathbf{x}_{N+1} \tag{7}$$

Although no sparse priors are imposed, the sparse property of the coefficient vector $\mathbf{a}$ is generated naturally by the L_1 optimization. The optimization of (7) is carried out using the matlab package provided by [5]. Once the vector $(\mathbf{a}^T, \mathbf{e}^T)^T$ is computed, the similarity coefficients $W_{(N+1)i}$ are set to:

$$W_{(N+1)i} = |a_i|, i = 1, \ldots, N$$

3.3. Advantages of the proposed out-of-sample embedding scheme

Although our proposed out-of-sample formula (Eq. (4)) is similar to that of the Latent Variable Model [6], it has the two following interesting differences and advantages:

1. For the LVM scheme, the neighborhood size must be set manually, and the optimal setting may be different for different data sets. In our scheme, the computation of similarity coefficients adapts to the dataset through the use of sparse coding. No parameter is required.

2. There have been many ways to compute the similarity coefficients and the most popular one among them is the typical Heat Kernel (Gaussian weighting function). However, the Gaussian aperture may affect the final classification results significantly, and how to optimally determine this parameter is still an open problem. Our scheme get rid of this since we exploit the

sparseness property of the deduced coefficients in order to express both adjacency structure and the associated weights without any predefined parameter.

4. Performance evaluation

To validate the effectiveness of our proposed approach, we applied it to the face recognition problem. The experimental results are reported in terms of recognition accuracy and a similarity measure of the embeddings ('out-of-sample' vs 'batch-mode').

4.1. Data sets

We considered in our experiments four public face data sets. All these databases are characterized by a large variation in face appearance. These databases are as follows: **ORL**[1], **Extended Yale - part B**[2], **PIE**[3], and **PF01**[4]. Figure 2 shows some face samples in the PF01 face Database.

Figure 2. Some samples in PF01 data set.

4.2. Recognition accuracy

To make the computation of the embedding process more efficient, the dimensionality of the original face samples was reduced by applying random projections [12]. It has a similar role to that of PCA yet with the obvious advantage that random projections do not need any training data.

We have compared our method with three other approaches. The first one is the Latent Variable Model (LVM), proposed in [6]. The second one is a linearization method of the existing mapping $\mathbf{X}_s \rightarrow \mathbf{Y}_s$. To this end, we use simple linear regression in order to infer a matrix transform $\mathbf{A}$ that best approximates the existing mapping through the linear equation $\mathbf{Y}_s = \mathbf{A}^T \mathbf{X}_s$. We stress the fact the linearization has not been thoroughly tested as an out-of-sample method. Instead, this linearization was used for spectral regression (e.g., [4]). The third one is a representation based on Radial-Basis Functions (RBF) [17, 10]. In our implementation of the RBF method, we used Gaussian kernels whose number is equal to the number of training samples. In other words,

---

[1] $http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html$

[2] $http://vision.ucsd.edu/\sim leekc/ExtYaleDatabase/ExtYaleB.html$

[3] $http://www.ri.cmu.edu/projects/project\_418.html$

[4] $https://sites.google.com/site/postechimlab2012/databases/face-database-2001$

we considered each training sample as a center. The aperture of the Gaussian kernels was set to the average squared distances between the pairs of training samples.

For each face data set and for every embedding method, we conducted three groups of experiments for which the percentage of training samples was set to 30%, 50% and 70% of the whole data set. The remaining data was used for testing. Here, the testing implies: (i) the out-of-sample embedding of the unseen observation (face) **(new observation embedding)**, and (ii) assigning it a class-label through the use of a classifier in the embedded space **(recognition)**.

We considered for comparison two classifiers: nearest neighbor (NN) and Support Vector Machines (SVM). For a given out-of-sample embedding method, the recognition rate was computed for several dimensions belonging to $[5, L_{max}]$, where L_{max} is the maximum dimensionality of LE embedding, which is equal to the number of training samples. Figures 3 and 4 illustrate the recognition rate as a function of the dimension of the embedded space for all the 4 out-of-sample embedding methods and for the four face datasets when the training percentage is set to 30%. In these two figures, the maximum dimensionality L_{max} is different for every face database since each database has a different size. The curves have been obtained by averaging the results over ten random splits. Figure 3 corresponds to the NN classifier (1 nearest neighbor). Figure 4 corresponds the SVM classifier based on a Gaussian kernel.

Tables 1 and 2 illustrate the best (average) performance obtained by each 'out-of-sample' method, based on 10 random splits using NN and SVM, respectively. Numbers marked in bold designate the best results. Each table depicts the results of two groups of experiments for which the training percentage was 50% and 70%, respectively. For the case of LVM method, the ϵ parameter corresponds to the number of neighbors used to approximate the unseen sample.

From the results, we can draw the following conclusions:

(i) For the case of NN, the above results confirm the superiority of our approach when compared with existing ones. We can observe that this superiority was obtained for all data sets and for all dimensions tested for the obtained embedding space. We can also observe that the linearization method provided the poorest results, which can be explained by the fact that the linear method is global and does not take into account the local adjacency information. We can also appreciate that, for the NN classifier, the performance of LVM and RBF depends on the dataset used. There is no general trend that shows that one method is better than the other.

(ii) For the case of SVM, the sparse representation does not guarantee always the best recognition accuracy rate, but it can be outperformed by the RBF method for some few cases. This could be explained by the fact that both RBF
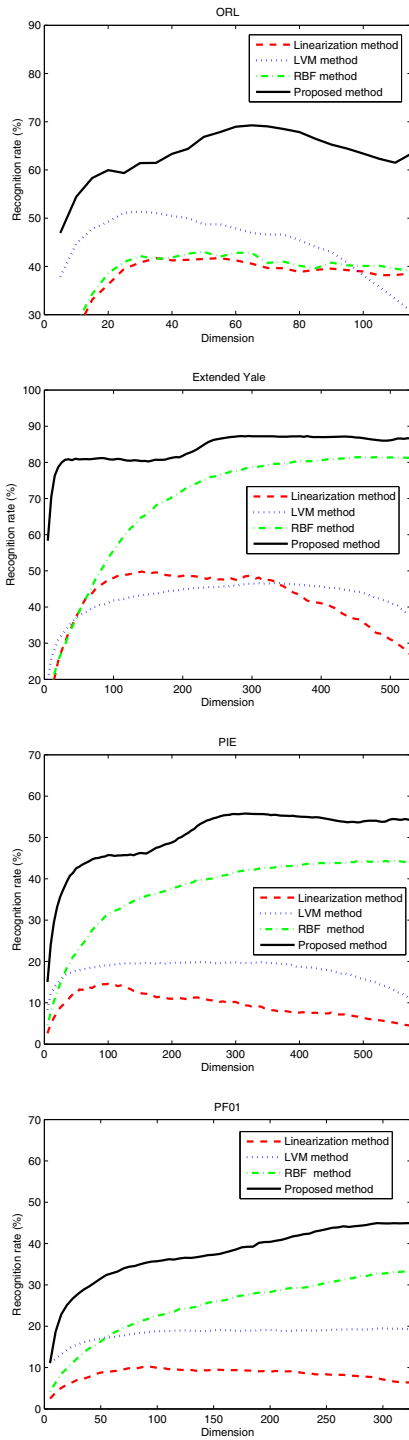
Figure 3. Experimental results on four face data sets. The used classifier was 1 NN.

Figure 4. Experimental results on four face data sets. The used classifier was SVM.

and SVM are highly non-linear techniques which can benefit each other well. For the SVM classifier, we can observe that the superiority of RBF was only obtained for few cases
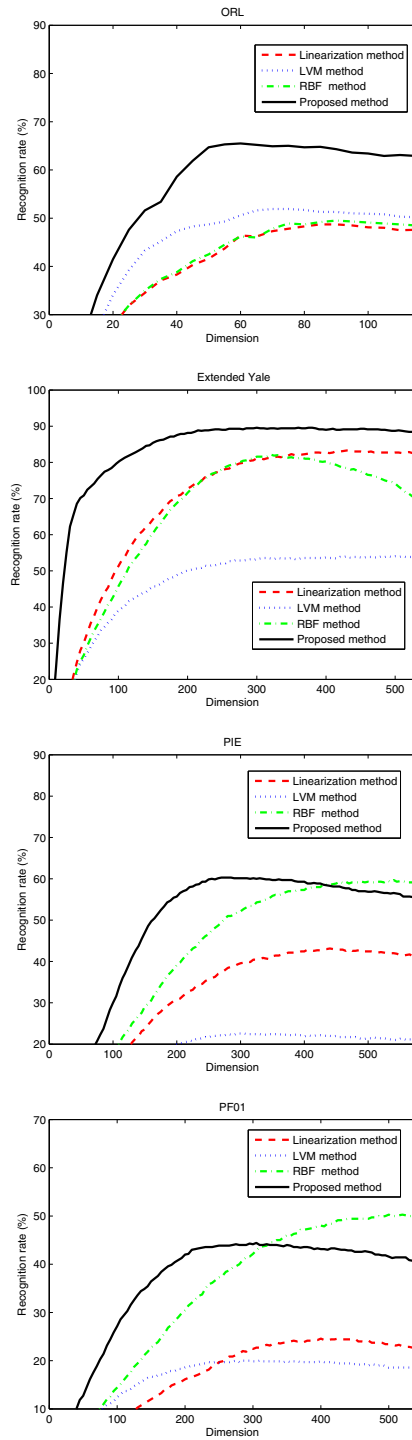
and for high dimensions (see PIE and PF01 datasets in Figure 4). In practice, one needs to find a trade-off between a high recognition rate and a compact representation with a

reduced number of dimensions. Thus, this requirement favors again our proposed sparse representation method since it has the best performance for low dimensions even when difficult face datasets (such as PIE) are considered. It is worth mentioning that this advantage is not shown in the Tables since the latter depict the best performances over the number of dimensions.

Dataset	Sparse	LVM		Lin.	RBF
50%-50%		$\epsilon = 3$	$\epsilon = 5$		
ORL	**82.5%**	72.0%	60.3%	46.4%	52.4%
Ext. Yale	**91.4%**	61.1%	46.8%	53.1%	89.6%
PIE	**66.2%**	27.5%	20.6%	12.2%	56.2%
PF01	**52.7%**	27.3%	20.4%	8.2%	42.1%
70%-30%		$\epsilon = 3$	$\epsilon = 5$		
ORL	**88.7%**	82.1%	73.6%	53.2%	68.6%
Ext. Yale	92.1%	70.9%	58.3%	57.1%	**92.5%**
PIE	**72.8%**	35.4%	26.8%	13.4%	64.0%
PF01	**54.4%**	34.0%	27.1%	8.6%	47.8%

Table 1. Maximum average recognition rate using the Nearest Neighbor classifier.

Dataset	Sparse	LVM		Lin.	RBF
50%-50%		$\epsilon = 3$	$\epsilon = 5$		
ORL	**82.6%**	75.4%	64.0%	53.7%	62.6%
Ext. Yale	**94.1%**	68.8%	63.0%	88.3%	88.0%
PIE	73.28%	31.6%	23.5%	38.5%	**74.3%**
PF01	53.42%	28.7%	23.0%	20.1%	**62.3%**
70%-30%		$\epsilon = 3$	$\epsilon = 5$		
ORL	**90.5%**	85.6%	77.0%	62.8%	75.7%
Ext. Yale	**95.7%**	78.5%	75.0%	90.5%	91.0%
PIE	**80.7%**	40.5%	30.1%	36.7%	**80.7%**
PF01	60.3%	35.8%	29.1%	15.7%	**72.1%**

Table 2. Maximum average recognition rate using the SVM classifier.

4.3. Assessing manifold reconstruction accuracy

In the previous section, we have evaluated the recognition performance of the proposed out-of-sample embedding method. However, the main role of the out-of-sample embedding method is to complete the reconstruction of the manifold in the embedded space (i.e., by adding the new observations to the embedded space). To this end, we can compare the coordinates of the new embedded observations with their coordinates computed in the batch mode. The batch mode assumes that the whole data set is used in order to get the non-linear manifold learning.

In order to quantify the accuracy of the out-of-sample embedding methods, we use the following error measure:

	Sparse	LVM	Linearization	RBF
50%-50%				
ORL	**0.3597**	0.4058	0.9614	0.7167
Ext. Yale	**0.3564**	0.3892	0.7528	0.4958
PIE	**0.4248**	0.4514	0.7758	0.4462
PF01	**0.4040**	0.4746	0.7746	0.4433

Table 3. Alignment error between batch-mode manifold and the out-of-sample computed manifold (See text for details.)

$$e = \frac{dist(\mathbf{Y}, \hat{\mathbf{Y}})}{\|\hat{\mathbf{Y}}\|_F}$$

where $dist(,)$ denotes the Procrustes distance [13], $\|\mathbf{A}\|_F$ denotes the Frobenius norm of the matrix $\mathbf{A}$, and $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ are the test data that are provided by the out-of-sample method and the associated batch one, respectively. The above error can quantify the dissimilarity between the batch mode geometric configuration and the out-of-sample geometric configuration related to the test observations.

In table 3, we show some results based on this definition for all out-of-sample methods. Numbers marked in bold represent the best alignment between 'out-of-sample' and 'batch-mode' embedding. The smaller the number, the better the alignment. We could conclude that sparse representation offers the best similarity with the batch mode embedding. In the above experiment, the percentage of training is set to 50%.
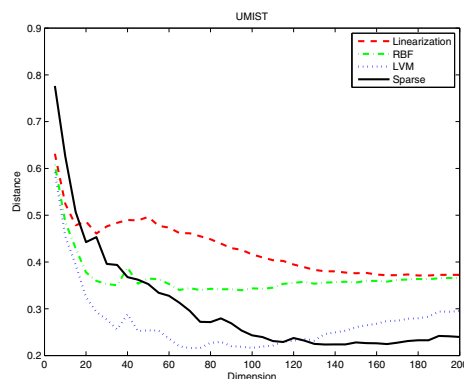

Figure 5. Alignment error as a function of the embedded space dimensionality.

Figure 5 shows, for the UMIST dataset[5] (training percentage is 70%), the evolution of the dissimilarity, e, obtained by the out-of sample methods as function of dimensionality of the embedded space. We could appreciate that, for all out-of-sample methods, the dissimilarity distance is decreasing with the increase of the embedding dimensional-

---

[5]$http : //www.shef.ac.uk/eee/research/vie/research/$
$face.html$

ity. However, after a certain threshold, the Sparse representation, again, guarantees the highest similarity. We can also observe that the alignment obtained by the LVM and proposed methods was much better than that of the linearization and RBF methods.

5. Conclusion

In this paper, we demonstrated that sparse representation can serve as an accurate alternative for out-of-sample embedding. Considering for a case study the Laplacian Eigenmaps, we applied our method to the face recognition problem. Indeed, the proposed out-of-sample embedding in general provided the best classification accuracy as well as the best alignment between out-of-sample mode and batch mode. The experimental results demonstrate that our algorithm can maintain an accurate low-dimensional representation of the data without any parameter tuning. A natural extension of our approach is its application to online learning and incremental embedding.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. 1, 2

[2] Y. Bengio, J. Paiement, and P. Vincent. Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps and spectral clustering. In *Advances in Neural Information Processing*, 2004. 2

[3] I. Borg and P. Groenen. *Modern Multidimensional Scaling: theory and applications*. Springer-Verlag New York, 2005. 1

[4] D. Cai, X. He, and J. Han. Spectral regression for efficient regularized subspace learning. In *Proc. Int. Conf. Computer Vision (ICCV'07)*, 2007. 4

[5] E. Candes and J. Romberg. *l1-magic: recovery of sparse signals via convex programming*. CALTECH, 2005. http://www.acm.caltech.edu/l1magic/. 3

[6] M. A. Carreira-Perpinan and Z. Lu. The Laplacian Eigenmaps latent variable model. *Journal of Machine Learning Research*, 2:59–66, 2007. 3, 4

[7] P. Dollar, V. Rabaud, and S. Belongie. Learning to traverse image manifolds. In *NIPS*, 2006. 1

[8] D. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. In *Proc. of the National Academy of Arts and Sciences*, 2003. 1

[9] D. L. Donoho, M. Elad, and V. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Information Theory*, 52(1):6–18, 2006. 2

[10] A. Elgammal and C. Lee. Non-linear manifold learning for dynamic shape and dynamic appearance. *Computer Vision and Image Understanding*, 106(1):31–46, 2007. 1, 4

[11] X. Geng, D. Zhan, and Z. Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on systems, man, and cybernetics-part B: cybernetics*, 35:1098–1107, 2005. 1

[12] N. Goel, G. Bebis, and A. Nefian. Face recognition experiments with random projections. In *SPIE Conference on Biometric Technology for Human Identification*, 2005. 4

[13] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 1996. 6

[14] J.-B. Huang and M.-H. Yang. Fast sparse representation with prototypes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3618–3625, june 2010. 2, 3

[15] J. Liu. Face recognition on Riemannian manifolds. Master's thesis, Autonomous University of Barcelona, 2011. 2

[16] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS 14*, 2002. 3

[17] C. Piret. *Analytical and Numerical Advances in Radial Basis Functions*. PhD thesis, University of Colorado at Boulder, 2007. 4

[18] B. Raducanu and F. Dornaika. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 45:2432–2444, 2012. 1

[19] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 1

[20] L. K. Saul, S. T. Roweis, and Y. Singer. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003. 1

[21] M. Scheuerer. An alternative procedure for selecting a good value for the parameter c in rbf-interpolation. *Advances in Computational Mathematics*, 34(1):105–126, 2011. 1

[22] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. 1

[23] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 1

[24] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009. 2

[25] Y. Xu, A. Zhong, J. Yang, and D. Zhang. LPP solution schemes for use with face recognition. *Pattern Recognition*, 43:4165–4176, 2010. 2

[26] S. Yan and H. Wang. Semi-supervised learning by sparse representation. In *SIAM International Conference on Data Mining*, 2009. 1, 2

[27] Y. Yang, F. Nie, S. Xiang, Y. Zhuang, and W. Wang. Local and global regressive mapping for manifold learning with out-of-sample extrapolation. In *American Association for Artificial Intelligence Conference*, 2010. 1

[28] S. X. Yu and J. Shi. Multiclass spectral clustering. In *IEEE International Conference on Computer Vision*, 2003. 3