Automatic Dimensionality Estimation for Manifold Learning through Optimal Feature Selection

Fadi Dornaika[1,2], Ammar Assoum[3], and Bogdan Raducanu[4]

[1] University of the Basque Country UPV/EHU, San Sebastian, Spain
[2] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
[3] LaMA Laboratory, Lebanese University, Tripoli, Lebanon
[4] Computer Vision Center, 08193 Bellaterra, Spain

Abstract. A very important aspect in manifold learning is represented by automatic estimation of the intrinsic dimensionality. Unfortunately, this problem has received few attention in the literature of manifold learning. In this paper, we argue that feature selection paradigm can be used to the problem of automatic dimensionality estimation. Besides this, it also leads to improved recognition rates. Our approach for optimal feature selection is based on a Genetic Algorithm. As a case study for manifold learning, we have considered Laplacian Eigenmaps (LE) and Locally Linear Embedding (LLE). The effectiveness of the proposed framework was tested on the face recognition problem. Extensive experiments carried out on ORL, UMIST, Yale, and Extended Yale face data sets confirmed our hypothesis.

1 Introduction

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) [1], Locally Linear Embedding (LLE) [2], Isomap [3], Supervised Isomap [4], Laplacian Eigenmaps (LE)[5,6]. This family of non-linear embedding techniques appeared as an alternative to their linear counterparts which suffer of severe limitation when dealing with real-world data: i) they assume the data lie in an Euclidean space, and ii) they may fail when the number of samples is too small. On the other hand, the non-linear dimensionality techniques are able to discover the intrinsic data structure by exploiting the local topology. In general, they attempt to optimally preserve the local geometry around each data sample while using the rest of the samples to preserve the global structure of the data. Most of existing works on non-linear manifold learning techniques are focused either on the graph design [7] or on the objective function that should be optimized. However, to the best of our knowledge, there is no work attempting to automatically estimate the dimensionality of the non-linear embedding. For classification tasks, the common way was to plot the performance over a validation (test) data set as a curve from which the optimal dimension can be estimated. This assumes that all dimensions below the found one will be considered as relevant and all dimensions beyond it should be irrelevant. This assumption seems to be very simplistic and does not take into account the effect of subsets of dimensions.

G.L. Gimel' farb et al. (Eds.): SSPR & SPR 2012, LNCS 7626, pp. 575–583, 2012.

For this reason, we address this problem in the current paper. The main contribution of our work is represented by a generic framework associated with manifold learning which allows the extraction and selection of optimal features (dimensions) in the embedded subspace (from the perspective of pattern classification). Our approach for feature selection is guided by a Genetic Algorithm (GA). The advantage of the proposed framework is twofold. First, by selecting the most relevant features (dimensions), the classification performance is enhanced (as proven by the experimental results). Second, by retaining only the most relevant dimensions, pattern classification task becomes much more efficient[1] (from the point of view of computational complexity).

The remainder of the paper is organized as follows. Section 2 reviews two non-linear manifold learning techniques. Section 3 briefly describes the feature selection paradigm. Section 4 provides some experimental results obtained with four public face data sets. Finally, section 5 contains our conclusions.

2 Non-linear Embedding Techniques

2.1 Laplacian Eigenmaps

Laplacian Eigenmaps is a recent non-linear dimensionality reduction technique that aims to preserve the local structure of data [5]. 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 $\{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^D$. Let's define a neighborhood graph on these samples, such as a K-nearest-neighbor or ϵ-ball graph, or a full mesh, and weigh each edge $\mathbf{x}_i \sim \mathbf{x}_j$ by a symmetric affinity function $W_{ij} = K(\mathbf{x}_i; \mathbf{x}_j)$, typically Gaussian $W_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\beta})$

where β is usually set to the average of squared distances between all pairs. LE seeks latent points $\{\mathbf{y}_i\}_{i=1}^N \subset \mathbb{R}^L$ that minimize $\frac{1}{2}\sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2 W_{ij}$, which discourages placing far apart latent points that correspond to similar observed points. If $\mathbf{W} \equiv W_{ij}$ denotes the symmetric affinity matrix and $\mathbf{D}$ is the diagonal weight matrix, whose entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$, then the Laplacian matrix is given $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The objective function can also be written as:

$$\frac{1}{2}\sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} = tr(\mathbf{Z}^T \mathbf{L}\, \mathbf{Z}) \tag{1}$$

where $\mathbf{Z}^T = \mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ is the $N \times L$ embedding matrix and $tr(.)$ denotes the trace of a matrix. The i^{th} row of the matrix $\mathbf{Z}$ provides the vector $\mathbf{y}_i$—the embedding coordinates of the sample $\mathbf{x}_i$.

The embedding matrix $\mathbf{Z}$ is the solution of the optimization problem:

$$\min_{\mathbf{Z}} tr(\mathbf{Z}^T \mathbf{L}\, \mathbf{Z}) \;\; s.t. \;\; \mathbf{Z}^T \mathbf{D}\, \mathbf{Z} = \mathbf{I}, \;\; \mathbf{Z}^T \mathbf{L}\, \mathbf{e} = \mathbf{0} \tag{2}$$

[1] This is a clear advantage for large data sets for which the dimensionality of the non-linear embedded space is equal to the size of the data set.

where $\mathbf{I}$ is the identity matrix and $\mathbf{e} = (1, \ldots, 1)^T$. The first constraint eliminates the trivial solution $\mathbf{Z} = \mathbf{0}$ (by setting an arbitrary scale) and the second constraint eliminates the trivial solution $\mathbf{e}$ (all samples are mapped to the same point). Standard methods show that the embedding matrix is provided by the matrix of eigenvectors corresponding to the smallest eigenvalues of the generalized eigenvector problem,

$$\mathbf{L}\,\mathbf{z} = \lambda\,\mathbf{D}\,\mathbf{z} \tag{3}$$

Let the column vectors $\mathbf{z}_0, \ldots, \mathbf{z}_{N-1}$ be the solutions of (3), ordered according to their eigenvalues, $\lambda_0 = 0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1}$. The eigenvector corresponding to eigenvalue 0 is left out and only the next eigenvectors for embedding are used. The embedding of the original samples is given by the row vectors of the matrix $\mathbf{Z}$, that is, $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] = \mathbf{Z}^T$.

2.2 Locally Linear Embedding

One important geometric intuition behind the LLE algorithm is that each data point and its neighbors lie on or are close to a locally linear patch of the manifold. LLE tries to characterize the geometry of the local patches by finding the linear coefficients that reconstruct each data point from its neighbors. In the first step, each sample is approximated by a weighted linear combination of its K nearest neighbors, making use of the assumption that neighboring samples will lie on a locally linear patch of the nonlinear manifold. To find the reconstruction weight matrix $\mathbf{W}$, where the entry W_{ij} contains the weight of neighbor j in the reconstruction of sample $\mathbf{x}_i$. The reconstruction error is minimized subject to the constraint that the rows of the weight matrix sum to one: $\sum_{j=1}^{N} w_{ij} = 1$.

Let $\mathbf{Y}$ be the non-linear embedding of the original data $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]$. Then $\mathbf{Y}$ can be computed by minimizing the following embedding cost function:

$$\sum_{i=1}^{N} ||\mathbf{y}_i - \sum_{j=1}^{N} w_{ij}\mathbf{y}_j||^2 = tr(\mathbf{Y}\,\mathbf{M}\,\mathbf{Y}^T) \tag{4}$$

where $\mathbf{M}$ is given by $\mathbf{M} = (\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W})^T$. The eigenvectors of the matrix $\mathbf{M}$ corresponding to the smallest eigenvalues then form the final embedding $\mathbf{Y}$.

3 Feature Selection

3.1 Overview

In many fields, including pattern recognition and machine learning, the input data are represented by a very large number of features, but only few of them are relevant for classification task. Many algorithms become computationally intractable when the dimensionality of the data is too high. On the other hand, once an optimal set of selected features has been chosen, even the basic classifiers (e.g., K-nearest neighbor) can achieve desirable performance. Therefore, the process of feature selection, i.e. the

task of choosing a small subset of features which is statistically relevant, can be critical to minimize the classification error. At the same time, feature selection also reduces training and inference time and leads to a better data visualization as well as to a reduction of measurement and storage requirements. Roughly speaking, feature selection algorithms have two key problems [8,9]: (i) search strategy and (ii) evaluation criterion. The first key problem refers to the strategy of the search in the space of all possible solutions. Roughly speaking, the search strategies can be optimal or heuristic. Regarding the second key problem, feature selection algorithms can be categorized into filter model and wrapper model. In the wrapper model, the feature selection method tries to directly optimize the performance of a specific predictor (classification or clustering algorithm). The main drawback of this method is its computational deficiency. In the filter model, the feature selection is done as a preprocessing, without trying to optimize the performance of any specific predictor directly [10,11,12]. A comprehensive discussion of feature selection methodologies can be found in [13].

3.2 Optimal Feature Subset Using a Genetic Algorithm

We adopt here a wrapper technique for feature selection. The adopted evaluation strategy will attempt to maximize the recognition accuracy over a given validation set.

The adopted search strategy will be carried out using a Genetic Algorithm (GA). Genetic Algorithms (GAs) are biologically motivated adaptive systems based on natural selection and genetic recombination [14] whose main application is for optimization problems. In the standard GA, candidate solutions are encoded as fixed length vectors–strings. We use a bit string representation whose length is determined by the number of eigenvectors obtained as a result of the embedding process. Thus, each eigenvector is associated with one bit in the string. If the i^{th} bit is 1, then the i^{th} eigenvector is selected. Otherwise, that component is discarded. Each string thus represents a different subset of eigenvectors. The initial population of solutions is chosen randomly. These candidate solutions are allowed to evolve over a certain number of generations. At each generation, the fitness of each string is set to the recognition rate over a fixed validation set.

4 Performance Study

To verify the effectiveness of our proposed framework, we applied it to the face recognition problem. Four public face data sets are considered.

4.1 Data Sets

1. The ORL face data set[2]. There are 10 images for each of the 40 human subjects, which were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20^o. Some samples are shown in figure 1.

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

2. The UMIST face data set[3]. The UMIST data set contains 575 gray images of 20 different people. The images depict variations in head pose. Some samples are show in figure 2.
3. The Yale face data set[4]. It contains 11 grayscale images for each of the 15 individuals. The images demonstrate variations in lighting condition (left-light, center-light, right-light), facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. Figure 3 shows some instances from this dataset.
4. The Extended Yale Face Database B[5]. It contains 16128 images of 28 human subjects under 9 poses and 64 illumination conditions. In our study, a subset of 1800 images has been used. Figure 4 shows some face samples in the extended Yale Face Database B.

Fig. 1. Some samples in ORL data set

4.2 Experimental Results

The experiments consisted of two stages. In the first stage, the selection paradigm was run over a fixed validation set. For every face data set, the validation set was randomly set to 40% of the whole data set. In the second stage, we evaluated the generalization capacity of the obtained features (generalization tests). For each face data set and for every 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. The partition of the data set was done randomly. In all our experiments the classification in the embedded spaces (selected or unselected) was carried out by the Nearest Neighbor Classifier. Figure 5 shows the results of the first stage,

[3] http://www.shef.ac.uk/eee/research/vie/research/face.html
[4] http://see.xidian.edu.cn/vipsl/database_Face.html
[5] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

Fig. 2. Some samples in UMIST data set

Fig. 3. Some samples in YALE data set

Fig. 4. Some samples in Extended Yale data set

where the plots depict the validation results before (blue line) and after feature selection (red line), for each data set.

In table 1, we summarize the face recognition performance using the Laplacian Eigenmaps embedding on the four data sets (output of second stage). For every data set and for every training percentage, three schemes were used: the original features/ dimensions (Orig.), the selected features using the GA (GA), and the sorted features using the Fisher Score[6] (FS). This table illustrates the average best recognition rate (%) over 5 random splits. The number in parenthesis is the mean recognition over the available dimensions. We can observe that: (i) the best recognition rate remains almost the same for all schemes, however, the GA scheme got these results with a fraction of the original features which varies between 30% and 40% of the total dimensions, and

[6] This scheme re-ranks the features according to their Fisher Score.

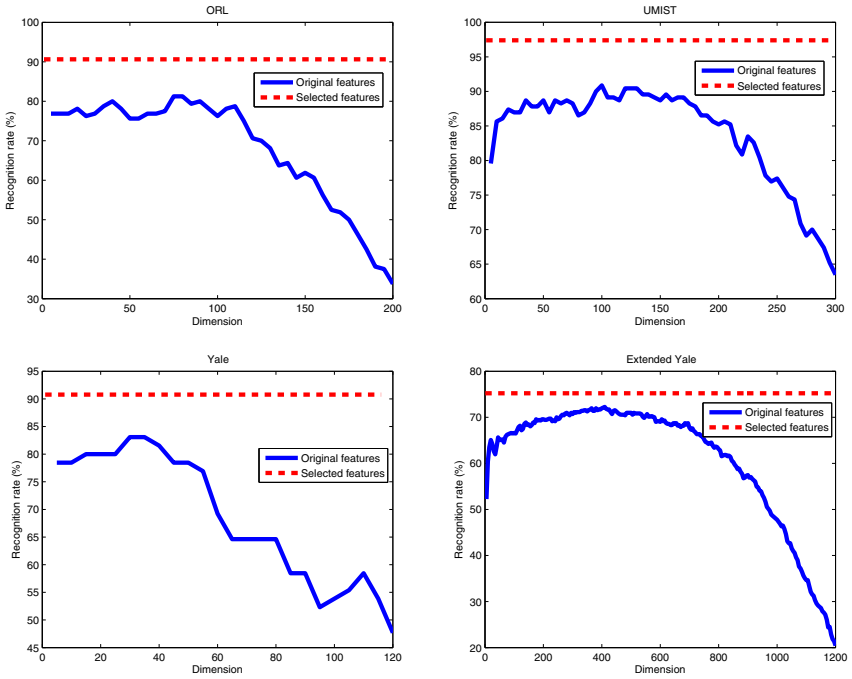

Fig. 5. Feature selection for LE embedding using a GA for four face data sets

Table 1. Comparison of recognition rates between the three schemes: maximum and average (in parenthesis)

30%	*ORL*	*UMIST*	*Yale*	*Ex.Yale*
Orig.	67.8 (52.5)	90.7 (73.8)	**75.9** (51.7)	67.8 (56.0)
GA	68.1 (**66.9**)	**91.3** (**83.8**)	74.1 (**61.7**)	68.2 (**65.2**)
FS	**71.8** (48.9)	89.7 (60.3)	73.3 (44.2)	**69.1** (54.7)
50%				
Orig.	73.5 (65.4)	94.1 (83.1)	**79.2** (57.7)	73.5 (63.2)
GA	74.0 (**79.8**)	**94.5** (**90.0**)	79.0 (**66.2**)	**74.0** (**70.8**)
FS	**84.0** (61.8)	94.0 (70.2)	77.5(49.5)	73.8 (61.4)
70%				
Orig.	87.1 (72.9)	**95.8** (88.9)	**78.8** (64.0)	75.5 (64.2)
GA	87.3 (**83.8**)	95.6 (**92.2**)	78.4 (**71.9**)	**76.0** (**72.2**)
FS	**88.1** (69.1)	94.9 (77.7)	76.7 (54.7)	75.5(60.4)

(ii) the GA method provided the most stable recognition rate as a function of the features used. Table 2 shows the face recognition performance using the LLE embedding on UMIST data set. Table 3 provides a comparison between the original dimensionality of the embedded space and the dimensionality discovered by our feature selection approach (in bold).

Table 2. Average best recognition rate (%) over 5 random splits using the LLE method

UMIST	*Train*30%	*Train*50%	*Train*70%
Orig.	**64.3** (47.4)	**73.7** (60.1)	**78.5** (66.4)
GA	61.6 (**54.5**)	73.1 (**64.6**)	77.7 (**68.8**)
FS	60.3 (45.5)	72.3 (58.2)	76.0 (63.9)

Table 3. The size of the selected features as obtained by the GA

	ORL	*UMIST*	*Yale*	*Ex.Yale*
LE	**68** (200)	**141** (300)	**54** (120)	**486** (1200)
LLE	**77** (200)	**82** (200)	**32** (80)	**243** (600)

5 Conclusion

In this paper, we proposed an automatic estimation of dimensionality for manifold learning through an optimal feature selection framework, based on a Genetic Algorithm. Experimental results show that the proposed approach can enhance the global performance for classification tasks, while reducing the computational complexity by removing the irrelevant features obtained by the non-linear embeddings.

Acknowledgment. This work was partially supported by the Spanish Government under the project TIN2010-18856 and the Lebanese National Council for Scientific Research (LCNRS) under the project 03-10-11.

References

1. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319 (1998)
2. Saul, L.K., Roweis, S.T., Singer, Y.: Think globally, fit locally: Unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research 4, 119–155 (2003)
3. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
4. Geng, X., Zhan, D., Zhou, Z.: Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 35, 1098–1107 (2005)
5. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6), 1373–1396 (2003)
6. Jia, P., Yin, J., Huang, X., Hu, D.: Incremental Laplacian Eigenmaps by preserving adjacent information between data points. Pattern Recognition Letters 30(16), 1457–1463 (2009)
7. Zhan, L., Qiao, L., Chen, S.: Graph-optimized locality preserving projections. Pattern Recognition 43, 1993–2002 (2010)
8. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. Journal of Machine Learning Research 5, 845–889 (2004)
9. Zhao, Z., Liu, H.: Spectral feature selection for supervised and unsupervised learning. In: Int. Conference on Machine Learning (2007)

10. Mitra, P., Murthy, C., Pal, S.: Unsupervised feature selection using feature similarity. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 301–312 (2002)
11. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: Advances in Neural Information Processing Systems 18 (2005)
12. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2010 (2010)
13. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowledge Data Engineering 17, 494–502 (2005)
14. Srinivas, M., Patnaik, L.: Genetic algorithms: a survey. IEEE Computer 27(6), 17–26 (1994)