An Empirical Comparison of Graph-based Dimensionality Reduction Algorithms on Facial Expression Recognition Tasks

Li He
Department of Computer Science
Fudan University, China
demonstrate@163.com

José M. Buenaposada
Dep. Ciencias de la Computación
Universidad Rey Juan Carlos

Luis Baumela
Dep. Inteligencia Artificial
Universidad Politécnica de Madrid
http://www.dia.fi.upm.es/~pcr

Abstract

Facial expression recognition is a topic of interest both in industry and academia. Recent approaches to facial expression recognition are based on mapping expressions to low dimensional manifolds. In this paper we revisit various dimensionality reduction algorithms using a graph-based paradigm. We compare eight dimensionality reduction algorithms on a facial expression recognition task. For this task, experimental results show that although Linear Discriminant Analysis (LDA) is the simplest and oldest supervised approach, its results are comparable to more flexible recent algorithms. LDA, on the other hand, is much simpler to tune, since it only depends on one parameter.

1. Introduction

One of the open problems of computer science is to make computers that interact with humans in a natural way. A key element in natural human computer interaction is the recognition of human facial expressions. Recently, much effort is being devoted within the computer vision research community to processing video sequences and modeling dynamic facial expressions [1, 9, 13]. One way to solve this problem is mapping facial expression to low dimensional manifolds exhibiting separable distributions for different expressions [1, 3, 9]. In this paper we compare the performance of eight graph-based dimensionality reduction algorithms on a facial expression recognition problem.

2. Face alignment and facial expression recognition

In our approach for facial expressions recognition, face images are located and tracked at video frame rates using an efficient face alignment procedure [5]. The tracker automatically crops the face and compensates illumination changes, as shown in Fig. 1, where the first row shows the original cropped images and the second row the corresponding illumination rectified ones (61×72 pixels images).

Figure 1. Illumination rectified images.

We aim to recognize Ekman's six prototypic facial expressions (joy, surprise, anger, sadness, fear, disgust). To do so we adopt a model-based approach for facial expression recognition. By tracking a set of 322 labeled image sequences of 92 subjects from the Cohn-Kanade data base [8], we build a user-and-illumination-independent global representation of all facial expressions (see [5] for details). In this model, a face image is represented with a point in a reduced dimensionality subspace (5 dimensions after PCA+LDA dimensionality reduction in our original formulation [5]). The variability of the classes of images associated to the prototypic facial expressions are represented by the Kohn-Kanade illumination rectified images projected onto a lower dimensional subspace embedded in the 90-dimensional PCA space of deformations, termed the *facial expression manifold*. In this paper we compare the performance of eight graph-based dimensionality reduction algorithms in the construction of the facial expression manifold.

Finally, images representing similar expressions are mapped to nearby points on the manifold. We use the nearest-neighbor probabilistic procedure introduced in [1], section 5, to combine the information provided

by the incoming image sequence with the information represented in the expression manifold to estimate the posterior probability of a facial expression.

3. Graph-based Dimensionality Reduction

3.1. The Basic Idea

On the whole the graph-based dimensionality reduction algorithms reviewed here are all built on the basis of a simple relationship (c.f. [4]):

$$\sum_{i,j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 W_{i,j} = 2\mathrm{tr}(X^\top L X), \qquad (1)$$

where $\boldsymbol{x}_i \in \mathbb{R}^n, i = 1, \ldots, N$, $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^\top$, W and L is the weight matrix and Laplacian matrix of a given graph respectively[1]. Equation (1) represents the scatterness of the given feature vectors $\boldsymbol{x}_i$ w.r.t. the given graph. For example, the early Locality Preserving Projection (LPP) algorithm [6] is unsupervised, which, via a linear projection P_{ULPP}, retains the neighborhood information obtained from high-dimensional data by choosing $W_{i,j} = 1$ when $\boldsymbol{x}_i \in \mathscr{N}(\boldsymbol{x}_j)$ or $\boldsymbol{x}_j \in \mathscr{N}(\boldsymbol{x}_i)$ ($\mathscr{N}(\boldsymbol{x})$ denotes the neighborhood of $\boldsymbol{x}$) and $W_{i,j} = 0$ otherwise. The desired P_{ULPP} minimizes

$$\min_P \mathrm{tr}(P^\top X^\top L_{\text{ULPP}} X P) \quad \text{s.t.} \quad P^\top X^\top D_{\text{ULPP}} X P = I,$$

where tr denotes the trace of a matrix. Later, a supervised version of LPP [7] is developed. The proposed graph G_{SLPP} has an edge between each pair of samples from different classes. Thus the desired projection will push samples of different classes away from each other and result in an increased Between Class Scatterness (BCS), i.e. P_{SLPP} solves the following optimization

$$\max_P \mathrm{tr}(P^\top X^\top L_{\text{SLPP}} X P) \quad \text{s.t.} \quad P^\top X^\top D_{\text{SLPP}} X P = I.$$

3.2. PCA and LDA

The well-known Principal Component Analysis (PCA) and LDA algorithms may also be described in terms of (1) using a graph-view of the common covariance matrix,

$$X^\top (I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top) X = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top = \sum_{i,j=1}^{N} W_{i,j} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top,$$

where $W_{i,j} = \frac{1}{N}$. Inspired by this formulation, Local Fisher Discriminant Criterion (LFDC) [10] rephrases LDA and adds locality to the classical LDA algorithm by modifying the original weights,

$$S_w = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} B_{i,j} W^w_{i,j} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top$$

$$S_b = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} B_{i,j} W^b_{i,j} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top$$

where

$$W^w_{i,j} = \begin{cases} \frac{1}{N_{y_i}} & y_i = y_j \\ 0 & \text{otherwise} \end{cases} \qquad W^b_{i,j} = \begin{cases} \frac{1}{N} - \frac{1}{N_{y_i}} & y_i = y_j \\ \frac{1}{N} & \text{otherwise} \end{cases}$$

are the original weights for the within class and between class graphs implicitly used in LDA, given that y_i and y_j are the class labels. LFDC imposes locality on these graphs by refraining the edges to only near samples, i.e. by defining B as a neighborhood matrix, that is, $B_{i,j} = 1$ iff $\boldsymbol{x}_i \in \mathscr{N}(\boldsymbol{x}_j)$ and $\boldsymbol{x}_j \in \mathscr{N}(\boldsymbol{x}_i)$, otherwise $B_{i,j} = 0$.

3.3. MFA, DNE and LSDA

Marginal Factor Analysis (MFA) [11], Discriminant Neighborhood Embedding (DNE) [12] and Locality Sensitive Discriminant Analysis (LSDA) [2] all build two graphs from neighborhood relationships, one for With-in Class Compactness (WCC), the other for BCS. MFA has one graph G^w_{MFA} whose edges are between each sample and its k_1-nearest neighbors in the same class, the other G^b_{MFA} whose edges are between each sample and its k_2-nearest neighbors in all other classes. It seeks the directions $\boldsymbol{v}_i, i = 1 \ldots, d$ that maximizes

$$\frac{\boldsymbol{v}_i^\top X^\top L^b_{\text{MFA}} X \boldsymbol{v}_i}{\boldsymbol{v}_i^\top X^\top L^w_{\text{MFA}} X \boldsymbol{v}_i} \quad \text{s.t.} \quad \boldsymbol{v}_i^\top X^\top L^w_{\text{MFA}} X \boldsymbol{v}_j = \delta_{i,j},$$

where $\delta_{i,j}$ is the Kronecker's delta and $j = 1, \ldots, i$. DNE, on the other hand, minimizes the difference,

$$\boldsymbol{v}_i^\top X^\top (L^w_{\text{MFA}} - L^b_{\text{MFA}}) X \boldsymbol{v}_i \quad \text{s.t.} \quad \boldsymbol{v}_i^\top \boldsymbol{v}_j = \delta_{i,j},$$

where $j = 1, \ldots, i$. In [12], Zhang and et al. paraphrase their idea with negative weights for the between class

[1] A graph G has several associated matrices, weight matrix W whose elements in ith row and jth column, $W_{i,j}$ is the weight of the edge between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and is zero when there's no edge between the two vertices, D for a diagonal matrix $\mathrm{diag}\{d_1, \ldots, d_N\}$ where $d_i = \sum_{j=1}^{N} W_{i,j}$, and the Laplacian matrix $L = D - W$. For different graphs, different sbscripts or supscripts are used.

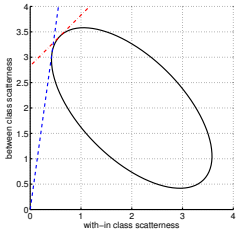

Figure 2. A Comparison of DNE, MFA and LSDA

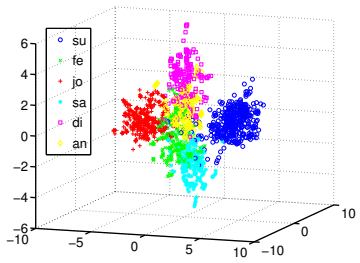

Figure 3. The expressions manifold in the PCA+LDA subspace (only the first 3 dimensions).

edges. It's easily seen that their Laplacian matrix for the graph with negative weights is almost $L^w_{\text{DNE}} - L^b_{\text{DNE}}$, except the edge rules—in [12] there will be edges between each sample and its k nearest samples, with positive weight if they are in the same class, otherwise negative ones. LSDA goes a little further by explicitly introducing a balancing parameter $\alpha \in [0, 1]$, resulting in

$$\begin{aligned} \max_{\boldsymbol{v}_i} \quad & \sum_{i=1}^{d} \boldsymbol{v}_i^\top X^\top (\alpha L^b_{\text{MFA}} - (1-\alpha) L^w_{\text{MFA}}) X \boldsymbol{v}_i \\ \text{s.t.} \quad & \boldsymbol{v}_i^\top X^\top W_w X \boldsymbol{v}_j = \delta_{i,j} \end{aligned}$$

or equivalently

$$\begin{aligned} \max_{\boldsymbol{v}_i} \quad & \sum_{i=1}^{d} \boldsymbol{v}_i^\top X^\top (\alpha L^b_{\text{MFA}} + (1-\alpha) W^w_{\text{MFA}}) X \boldsymbol{v}_i \\ \text{s.t.} \quad & \boldsymbol{v}_i^\top X^\top W_w X \boldsymbol{v}_j = \delta_{i,j} \end{aligned}$$

To have a clearer view of these algorithms, Fig. 2 shows a simple case: the ellipse-like curve shows the pair of WCC and BCS when a unit vector rotates in the plane. The MFA seeks for the line that intersects the ellipse with maximum slope (the blue dashed line). The DNE seeks for the line with slope 1 that intersects the ellipse and has largest intercept on y-axis (the red dash-dot line). Different choices of α for LSDA yield different lines that is just tangent to the upper half of the ellipse. If the embedding space is 1-dimensional, LSDA is best since with enough trial of α cross validation will ultimately picks a no worse projection than DNE and MFA. But it's not true for higher-dimensional embeddings, since α is constant.

4. Experiments

In this section we compare the dimensionality reduction algorithms described above for the facial expression recognition task introduced in section 2. For our comparison we used the Cohn-Kanade database, that was also used for building the expression manifold.

To estimate the recognition rate, we employ a leave-one-subject-out strategy for cross validation, in which sequences of each subject are tested against the model trained with all other sequences. Since all sequences in the database start with a neutral expression, we have verified that it is better to train the dimensionality reduction procedure with the last 6 images of each sequence. Hence during the training for each fold, there are more than 1500 images.

Cross validation also helps to obtain a model trained with the best suitable configuration parameters. There are several parameters that controls the behavior of the algorithms introduced in Sec. 3. Also the classifier has three parameters: a smoothing parameter h, a neighborhood size k and η to avoid the veto effect [1]. In the experiments, we search for best combination of the discriminant projection model and the classifier. For the classifier, η is manually set to 0.3, h takes values in $\{1/6, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1\}$ and k in $\{21, 23, \ldots, 59\}$.

In table 1, we display the results of the experiments conducted. Unsupervised algorithms such as PCA and ULPP lead to low recognition rates. In our feature space, images that are close to each other will not necessarily be of the same expression. On the contrary, they are more likely to be different expressions of the same subject. Thus, for building a good discriminating projection, it is not adequate to preserve a neighborhood based on pixel-level distance as ULPP does.

Surprisingly, LDA yields a competent recognition rate compared with other complicated algorithms, although it is the oldest and simplest supervised procedure. As shown in Fig. 3, the expression manifold contains one single cluster for each type of expression. Introducing locality into LDA, as LFDC does, slightly enhances the performance. The LFDC model in table 1 is trained with a neighborhood size 150, best in $\{30, 40, 50, 80, 100, 150, 200\}$.

Projection	PCA	LDA	SLPP	ULPP	LFDC	MFA	DNE	LSDA
Rate	76%	86%	85%	50%	86%	86%	84%	86%
Dimension	50	5	5	15	7	6	11	9
k	43	31	33	15	35	43	27	31
h	1/6	0.2	1/6	0.2	0.3	0.6	1/6	0.2

Table 1. Recognition rate for all eight dimensionality reduction approaches.

MFA is able to give result comparable to those of LDA when k_1 and k_2 are sufficiently large. We set $k_1 = k_2 = 100$ to obtain the reported result. Actually, a slightly worse result is obtained when setting $k_1 = k_2 = 50$ or $150, 200$. LSDA is more flexible than DNE since it is feasible to tune α and balance the BCS and WCC. The configuration for LSDA in table 1 is $k_1 = 100, k_2 = 100$ and $\alpha = 0$. It's interesting to note that the recognition rate actually decreases as α increases to 1, which means it's important to minimize WCC instead of maximizing BCS. The DNE projection in table 1 is trained with a neighborhood size 13.

These graph-based dimensionality reduction algorithms give us more room for tuning. Even though the data for facial expression recognition do not conform to the clustering assumption which might be necessary for designing them, they do work fine, perhaps even finer than the traditional algorithms. But on the other hand, with more choices of parameters, it takes much more time than LDA to find a proper setting. It is even possible to get a higher recognition rate given more time to search for other settings of LSDA. But anyway, now it might be close to the limit the linear methods could get to.

5. Conclusion

In this paper have revisited several dimensionality reduction algorithms and compared their performance on a facial expression recognition task. Unsupervised approaches like PCA and ULPP have the lowest recognition rates, since nearby images in our feature space are more likely to be different expressions of the same subject. Supervised approaches, on the other hand, achieve the best performance. LDA represents the best compromise between performance and complexity. For this problem, the WCC measure dominates BCS and consequently LSDA performs better than DNE. LSDA is the algorithm with the best recognition rate.

Also, these experiments show that, for appearance-based facial expression recognition tasks, we must build a large enough neighborhood for each sample, since the distance information in the feature space actually does not help in building discriminant projection and thus a small neighborhood would be misleading.

Acknowledgements

He Li was funded by NSFC 60635030, 863 Project (2007AA01Z176) and Universidad Politécnica de Madrid. José M. Buenaposada and Luis Baumela were funded by the Spanish *Ministerio de Educación y Ciencia*, under contract `TRA2005-08529-C02-02`.

References

[1] J. M. Buenaposada, E. Muñoz, and L. Baumela. Recognising facial expressions in video sequences. *Pattern Analysis and Applications*, 11(1):101–116, 2008.

[2] D. Cai, X. He, K. Zhou, J. Han, and H. Bao. Locality sensitive discriminant analysis. In *Proc. IJCAI*, pages 708–713, 2007.

[3] Y. Chang, C. Hu, and M. Turk. Probabilistic expression analysis on manifolds. In *Proc. CVPR*, volume 2, pages 520–527, 2004.

[4] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, Providence, Rhode Island, 1997.

[5] L. He, J. M. Buenaposada, and L. Baumela. Real-time facial expression recognition with illumination-corrected image sequences. In *Proc. of International Conference on Automatic Face and Gesture Recognition*, 2008.

[6] X. He and P. Niyogi. Locality preserving projections. Technical Report TR-2002-09, Department of Computer Science, University of Chicago, Oct 2002.

[7] X. He, S. Yan, Y. Hu, P. Niyogi, , and H.-J. Zhang. Face recognition using laplacianfaces. *IEEE Trans. on PAMI*, 27(3), 2005.

[8] T. Kanade, J. Cohn, and Y.-l. Tian. Comprehensive database for facial expression analysis. In *Proc. FG*, pages 46–53, 2000.

[9] C. Shan, S. Gong, and P. W. McOwan. Dynamic facial expression recognition using a bayesian temporal manifold model. In *Proc. BMVC*, volume 1, pages 297–306, 2006.

[10] M. Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In W. W. Cohen and A. Moore, editors, *ICML*, pages 905–912. ACM, 2006.

[11] S. Yan, D. Xu, B. Zhang, and H. Zhang. Graph embedding: A general framework for dimensionality reduction. In *Proc of CVPR*, pages 830–837, 2005.

[12] W. Zhang, X. Xue, Z. Sun, Y. Guo, and H. Lu. Optimal dimensionality of metric space for classification. In *Proc. of ICML*, pages 1135–1142, 2007.

[13] G. Zhao and M. Pietikäinen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. on PAMI*, 29(6):915–928, June 2007.