FACIAL EXPRESSION RECOGNITION BASED ON GRAPH-PRESERVING SPARSE NON-NEGATIVE MATRIX FACTORIZATION

Ruicong Zhi [*], *Markus Flierl* [†]

Qiuqi Ruan [*], *Bastiaan Kleijn* [†]

[*]Institute of Information Science
Beijing Jiaotong University
Beijing 100044, P.R. China
{05120370, qqruan}@bjtu.edu.cn

[†]ACCESS Linnaeus Center
School of Electrical Engineering
KTH – Royal Institute of Technology, Stockholm
{ruicong, mflierl, bastiaan}@kth.se

ABSTRACT

In this paper, we present a novel algorithm for representing facial expressions. The algorithm is based on the non-negative matrix factorization (NMF) algorithm, which decomposes the original facial image matrix into two non-negative matrices, namely the coefficient matrix and the basis image matrix. We call the novel algorithm graph-preserving sparse non-negative matrix factorization (GSNMF). GSNMF utilizes both sparse and graph-preserving constraints to achieve a non-negative factorization. The graph-preserving criterion preserves the structure of the original facial images in the embedded subspace while considering the class information of the facial images. Therefore, GSNMF has more discriminant power than NMF. GSNMF is applied to facial images for the recognition of six basic facial expressions. Our experiments show that GSNMF achieves on average a recognition rate of 93.5% compared to that of discriminant NMF with 91.6%.

Index Terms— Facial expression recognition, non-negative matrix factorization, graph-preserving constraint, sparse representations

1. INTRODUCTION

Efficient methods for facial expression recognition are based on finding low-dimensional representations for facial images by extracting facial features. The features reflect the intrinsic structure of facial expressions and allow for efficient classification.

A popular class of feature extraction methods is formed by appearance-based algorithms such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Locality Preserving Projections (LPP) [1]. They utilize linear transformations to obtain low-dimensional facial subspaces that have different structures. PCA aims to find subspaces with maximum covariance. LDA aims to find subspaces that maximize the ratio of the between-class scatter and the within-class scatter of the facial images. LPP obtains subspaces which preserve the locality structure of the facial images. PCA is an unsupervised method, and LDA is a supervised method, while LPP can be conducted as either unsupervised or supervised. All these methods obtain ”holistic” representations for facial images. As there is no further constraint on the linear transformation, the low-dimensional representations of facial images contain both negative and non-negative values. Negative intensity values in images do not have physical meaning. This motivates non-negative decomposition methods for arrays of facial images. Efficient decompositions can be achieved with the Non-negative Matrix Factorization (NMF) algorithm [2].

The NMF algorithm decomposes a facial image into a linear combination of basis images. Both the coefficient values and basis images are non-negative. That means that NMF only allows the combination of basis images using addition rather than subtraction. The basis images are ”block components” of the facial images, so they are likely representations for facial parts. However, previous work revealed that NMF decompositions produce holistic image representations rather than sparse representations. Therefore, related work focused on finding sparse representations of facial images, e.g., non-negative sparse coding (SNMF) [3], local non-negative matrix factorization (LNMF) [4], etc. These methods add sparsity constraints to NMF to obtain improved facial representations.

For facial expression recognition, the most important goal is to achieve high recognition accuracy when classifying different facial expressions. It is well-known that prior class information is very helpful for improving classification performance. All NMF-based algorithms mentioned above are unsupervised. They do not consider the fact that different samples belong to different classes. Therefore, we introduce a graph-preserving constraint that considers class information for sparse NMF. The graph-preserving constraint maintains the neighborhood structure of the facial samples after embedding them into the low-dimensional subspace. Experimental

This work has been completed while the first author visited KTH. It has been supported by the Chinese National Program 973 (No. 2004CB318005), the National Natural Science Foundation of China (No. 60672062, No. 60472033) and the Technological Innovation Fund for Excellent Doctoral Candidates of Beijing Jiaotong University (48036).

results show that this graph-preserving sparse non-negative matrix factorization (GSNMF) algorithm performs better than sparse NMF (SNMF) and discriminant NMF (DNMF)[8]. It has more discriminant power to distinguish six basic facial expressions and achieves higher recognition accuracies than other tested NMF methods.

The paper is organized as follows: Section 2 reviews the NMF algorithm briefly; Section 3 introduces the graph-preserving sparse NMF; Section 4 outlines the utilized projected gradient method, and Section 5 reports our experiments on facial expression recognition based on the Cohn-Kanade database.

2. REVIEW OF NMF

Let the matrix $X = [x_1, x_2, \cdots, x_n]$ represent a set of n images, where each image $x_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,m}]^T$ is given by a m-dimensional vector. The $m \times n$ image matrix X represents the image database. It can be approximated by a linear combination of basis images $W = [W_1, W_2, \cdots, W_p]$ with $W_i = [w_{i,1}, w_{i,2}, \cdots, w_{i,m}]^T$. That is, each image vector x_i can be written as a linear combination of basis images, i.e. $x_i = Wh_i$, where h_i is a p-dimensional vector containing the linear decomposition coefficients. Extending the approximation to all images, we have $X \approx WH$. Both the basis image matrix W and coefficient matrix H contain non-negative elements. One of the common methods to measure the quality of the approximation is the square of the Euclidean distance

$$D(X\|WH) = ||X - WH||_F^2 = \sum_{ij}\left(x_{ij} - \sum_k w_{ik}h_{kj}\right)^2, \tag{1}$$

where $||\cdot||_F$ denotes the Frobenius norm. NMF aims to find the non-negative decomposition of X according to the following optimization problem:

$$\begin{aligned} &\min_{W,H} \quad D(X\|WH) \\ &s.\,t. \quad w_{i,k} \geq 0, h_{k,j} \geq 0, \quad \forall\, k, i, j \end{aligned} \tag{2}$$

3. GRAPH-PRESERVING SPARSE NON-NEGATIVE MATRIX FACTORIZATION

Shortcomings of the original NMF have been pointed out and extensions of the original NMF have been suggested. Li [4] found that the original NMF can only extract global features from some face databases and developed a local non-negative matrix factorization for enforcing parts-based representations. Hoyer [3] extended the original NMF to include the option to control the sparseness of the NMF explicitly. However, the structure of the image space is not considered in these works. Here, we propose a novel graph-preserving sparse non-negative matrix factorization algorithm that obtains a sparse representation of the images while preserving the neighborhood structure of the image data.

The degree of sparseness of the representation can be measured by the number of nonzero elements obtained in the decomposition matrix of basis images. The l^0-norm counts the number of nonzero entries in a matrix. Note that if the solution of l^0 is sparse enough, the solution of the l^0-norm minimization problem is equal to the solution of the l^1-norm minimization problem. Therefore, a sparse non-negative decomposition can be obtained by adding l^1-norm constraints. The l^1-norm of the basis matrix is defined as $||W||_1 = \sum_{k,j} |w_{k,j}|$.

In order to find a subspace that preserves the structure of the high dimensional image space, we use a graph-preserving constraint that is derived from graph embedding theory. Let $G = \{X, S\}$ be an undirected weighted graph. X is the vector set, $X = [x_1, x_2, \cdots, x_n] \in \mathcal{R}^{m\times n}$ (each column of the matrix X represents an image). $S \in \mathcal{R}^{n\times n}$ is the similarity matrix. Each element of the real-valued symmetric matrix S measures the similarity of a pair of vertices, which is assumed to be non-negative in our paper. Each original data point x_i is projected by $\tilde{x}_i = W^T x_i$. The projected data matrix $\tilde{X} = [\tilde{x}_1\tilde{x}_2, \cdots, \tilde{x}_n] \in \mathcal{R}^{p\times n}$ is used for classification. Our graph-preserving constraint is given by

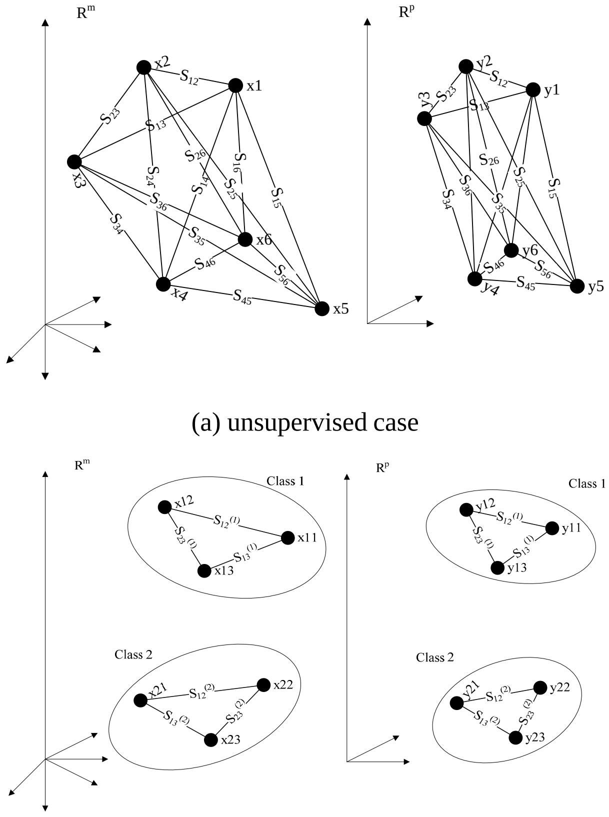
$$\sum_{i,j} ||\tilde{x}_i - \tilde{x}_j||_2^2 S_{ij}. \tag{3}$$

The graph-preserving constraint aims to preserve the neighborhood of the data samples. The graph-preserving character is illustrated in Fig.1. The graph-preserving constraint can be used unsupervised or supervised, depending on the construction method for the similarity matrix. If it is unsupervised, the weight matrix S is constructed according to the distances between pairs of samples. If it is supervised, the weight matrix is constructed utilizing the prior class information of the samples, and the weight values for sample pairs belonging to different classes are zero. The similarity value S_{ij} enforces similarity between the vectors as depicted in Fig. 1. For example, if the similarity between points x_i and x_j is large, the distance between the corresponding projected points $\tilde{x_i}$ and $\tilde{x_j}$ should be small.

We need to find a tradeoff between reconstruction error, graph-preserving threshold, and sparseness. Thus, the cost function of graph-preserving sparse non-negative matrix factorization (GSNMF) is defined as

$$\begin{aligned} &D_{GSNMF}(X\|WH) = \\ &||X - WH||_F^2 + \lambda \sum_{k,j} w_{k,j} + \eta \left(\sum_{i,j} ||\tilde{x}_i - \tilde{x}_j||_2^2 S_{ij}\right), \end{aligned} \tag{4}$$

where $||\cdot||_F$ is the Frobenius norm, λ is a positive multiplier which controls the sparseness, and η is a positive multiplier which controls the locality of the decomposition.



(a) unsupervised case

(b) supervised case

Fig. 1. Graphs for original data space and projected data space. y denotes projected points, $y = \tilde{x}_i = W^T x_i$.

4. PROJECTED GRADIENT FOR GSNMF

In order to calculate the update rules for coefficients and basis images, a projected gradient method is carried out that has been successfully used for the NMF algorithm [5]. The projected gradient method is better than the multiplicative update method as the former can guarantee the convergence of the optimization problem.

The desired decomposition is obtained by solving the following optimization problem:

$$\begin{aligned} \min_{W,H} \quad & D_{GSNMF}(X\|WH) \\ s.\,t. \quad & w_{i,k} \geq 0, h_{k,j} \geq 0, \quad \forall\, k,i,j \end{aligned} \tag{5}$$

By fixing W or H, we can get two functions $f_W^{GSNMF}(H)$ and $f_H^{GSNMF}(W)$, respectively. Then the optimization problem can be divided into two conditional problems. We utilize the projected gradient method to obtain the conditional problems. This method has been successfully used for NMF. According to the projected gradient algorithm [5], we need to calculate the first and second order gradients of the two functions $f_W^{GSNMF}(H)$ and $f_H^{GSNMF}(W)$. The GSNMF cost function contains three terms, and it can be written as $D_{GSNMF} = J_1 + \lambda J_2 + \eta J_3$, where $J_1 = ||X - WH||_F^2$, $J_2 = \sum_{k,j} w_{k,j}$ and $J_3 = \sum_{i,j} ||\tilde{x}_i - \tilde{x}_j||_2^2 S_{ij}$. $f_W^{GSNMF}(H)$ and $f_H^{GSNMF}(W)$ can be simplified as

$$\begin{aligned} f_W^{GSNMF}(H) &= J_1^H + C \qquad (C \text{ is constant}) \\ f_H^{GSNMF}(W) &= J_1^W + \lambda J_2^W + \eta J_3^W. \end{aligned} \tag{6}$$

Thus, the gradients of these two functions are

$$\begin{aligned} \nabla f_W^{GSNMF}(H) &= \nabla J_1^H \\ \nabla^2 f_W^{GSNMF}(H) &= \nabla^2 J_1^H \\ \nabla f_H^{GSNMF}(W) &= \nabla J_1^W + \lambda \nabla J_2^W + \eta \nabla J_3^W \\ \nabla^2 f_H^{GSNMF}(W) &= \nabla^2 J_1^W + \lambda \nabla^2 J_2^W + \eta \nabla^2 J_3^W. \end{aligned} \tag{7}$$

For the first conditional problem, we fix H and update W. The update rule is defined as

$$W^{(t+1)} = \left[W^{(t)} - \alpha_t \nabla f_H(W^{(t)}) \right]^+, \tag{8}$$

where $[\cdot]^+ = \max[\cdot, 0]$, t is the number of iterations, $\alpha_t = \beta^{\varphi_t}$ and φ_t is the first non-negative integer such that

$$\begin{aligned} &(1-\sigma)\left\langle \nabla f_H(W^{(t)}), W^{(t+1)} - W^{(t)} \right\rangle \\ &\quad + \tfrac{1}{2}\left\langle W^{(t+1)} - W^{(t)}, \left(W^{(t+1)} - W^{(t)}\right) \nabla^2 f_H(W^{(t+1)}) \right\rangle \leq 0, \end{aligned} \tag{9}$$

where $< \cdot, \cdot >$ is the Frobenius inner product. In this work, β and σ are chosen to be 0.1 and 0.01. The iterative procedure is set to stop when the solution is close to a stationary point. A common condition is

$$||\nabla^P f_H(W^{(t)})||_F \leq \varepsilon_W ||\nabla f_H(W^{(1)})||_F, \tag{10}$$

where ε_W is the predefined stopping tolerance. $\nabla^P f_H(W^{(t)})$ is the projected gradient. A similar procedure is conducted to update H by fixing W.

5. EXPERIMENTS FOR FACIAL EXPRESSION RECOGNITION

In this section, the proposed GSNMF algorithm is conducted for facial expression recognition on the Cohn-Kanade facial expression database [7]. The Cohn-Kanade database consists of subjects displaying distinct facial expressions, starting from neutral expression and ending with the peak of the expression. Each subject contains six basis facial expressions (anger, disgust, fear, happiness, sadness and surprise). For each expression of a subject, the last eight frames in the video are selected, and we treat these frames as static images for both training and testing. The original facial images are cropped to capture the faces only, and the size of each cropped image is 60×60. In addition, the nearest neighbor classifier is taken for classification. Some of the cropped image samples are shown in Fig.2.

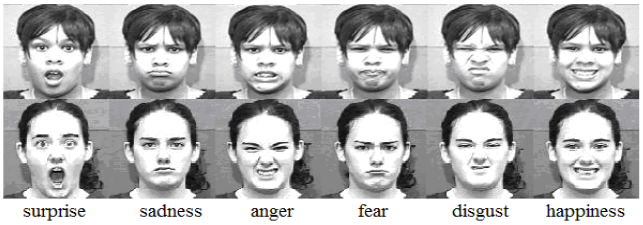


Fig. 2. Cropped face images from the Cohn-Kanade database.

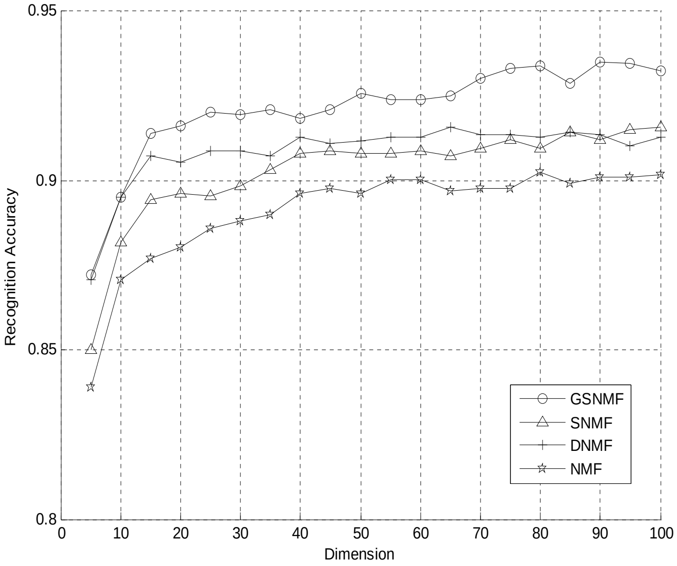


Fig. 3. Comparison of algorithms showing recognition accuracies versus the dimensionality of the projections on the Cohn-Kanade database.

In our experiments, we use the Gaussian kernel with Euclidean distance to construct the similarity matrix S. That is, if x_i and x_j belong to the same class, then $S_{ij} = \exp\left(-||x_i - x_j||_2^2/t\right)$ (t is an empirical positive parameter); otherwise, $S_{ij} = 0$. In our facial expression recognition experiments, a random subset with one randomly selected image of a person for each expression is taken to compose the training set, and the remaining images are used to form the testing set. Our proposed GSNMF algorithm is compared to three other NMF-based algorithms, namely SNMF, DNMF, and NMF. DNMF is the discriminant NMF as proposed in [8] and SNMF the non-negative sparse coding in [9]. The average facial expression recognition accuracies of the six facial expressions obtained by GSNMF, SNMF, DNMF and NMF are shown in Fig. 3. The best facial expression recognition accuracies obtained for GSNMF, SNMF, DNMF, and NMF are 93.5%, 91.6%, 91.6%, and 90.2%, respectively. The disgust images get most confused with sadness images. It can be seen that the proposed GSNMF algorithm achieves the best recognition accuracy compared to the other tested algorithms.

6. CONCLUSION

In this paper, we present a novel image representation approach that is applied to the facial expression recognition problem. GSNMF enhances the sparse and discriminant character of the NMF algorithm by using both the sparsity and graph-preserving constraints. The facial representation obtained by GSNMF aims to preserve the locality structure of the image space. For the optimization problem, we derive a projected gradient method. Our experiments show that GSNMF achieves higher recognition accuracy than NMF, SNMF, and DNMF. For example, GSNMF achieves a recognition rate of 93.5% compared to that of DNMF with 91.6%.

7. REFERENCES

[1] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, ”Face recognition using Laplacianfaces,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 3, pp. 328-340, 2005.

[2] D. D. Lee and H. S. Seung, ”Learning the parts of objects by non-negative matrix factorization,” Nature, vol. 401, no. 6755, pp. 788-791, 1999.

[3] P. O. Hoyer, ”Non-negative matrix factorization with sparseness constraints,” Journal of Machine Learning Research, vol. 5, pp. 1457-1469, 2004.

[4] S. Z. Li, X. W. Hou, H. J. Zhang and Q. S. Cheng, ”Learning spatially localized, parts-based representation,” in Proceeding of the IEEE International Conference on Computer Vision and Pattern Recognition, Hawaii, USA, pp. 207-212, 2001.

[5] C. J. Lin, ”Projected gradient methods for non-negative matrix factorization,” Neural Computation, vol. 17, no. 10, pp. 2756-2779, 2005.

[6] D. Donoho, ”For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution,” Communications On Pure and Applied Mathematics, vol. 59, no. 6, pp. 797-829, 2006.

[7] T. Kanade, J. Cohn, and Y. Tian, ”Comprehensive database for facial expression analysis,” in Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France, pp. 46-53, 2000.

[8] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, ”Exploiting discriminant information in nonnegative matrix factorization with application”, IEEE Transactions on Neural Networks, vol. 17, no. 3, pp. 683-695, 2006.

[9] P. O. Hoyer, Non-negative sparse coding, http:// www. cs. helsinki. fi/ u/ phoyer/ software.html