Joint Spatial Geometric and Max-margin Classifier Constraints for Facial Expression Recognition Using Nonnegative Matrix Factorization

Mai Lam and Phan Trong-Thanh

College of Information Technology, The University of Da Nang, Da Nang, Viet Nam
{mlam, ptthanh}@cit.udn.vn

Abstract. In this paper, we propose a new approach to facial expression recognition based on the constrained non-negative matrix factorization algorithm. Our proposed method incorporated two tasks in an automatic expression analysis system: facial feature extraction and classification into expressions. To obtain local and geometric structure information in the data as much as possible, we unite max-margin classification into the constrained NMF optimization, resulting in a multiplicative updating algorithm is also proposed for solving optimization problem. Experimental results on JAFFE dataset demonstrate that the effectiveness of the proposed method with improved performances over the conventional dimension reduction methods.

Keywords: facial expressions; nonnegative matrix factorization; classification; graph regularization; spatial constraints

1 Introduction

Facial expression recognition (FER) has been increased the attention from psychologists anthropologists, and computer scientists [1, 2, 3]. The computer researchers attempt to create complex human-computer interfaces that are able to automatically recognizing and classifying human expressions or emotions. Fasel et al. [1] define facial expressions as temporally deformed facial features such as eyelids, eyebrows, nose, lips and skin texture created by compressions of facial muscles. They observed common changes of muscular activities to be brief, “lasting for a few seconds, but rarely more than five seconds or less than 250 ms”. They additionally bring up the essential fact that felt emotions are only single source of facial expressions besides others like verbal and non-verbal correspondence or physiological activities.

In spite of facial expressions are not to equate with feelings (and the terms are commonly wrongly exchanged), in the PC vision group, the expression " facial expression recognition" frequently refers to the characterization of facial components in one of the six alleged essential feelings: happiness, sadness, fear, disgust, surprise and anger, as presented by Ekman in 1971 [2]. This endeavor of an elucidation depends on the suspicion that the appearances of feelings are widespread crosswise over people and also human ethnics and societies.

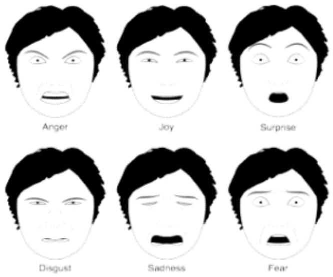

Fig. 1. Six universal emotions

Two tasks are necessary for an automatic expression analysis system [4]: facial feature extraction and classification into expressions. In facial feature extraction, there are mainly two kinds of approaches: geometric feature-based and appearance-based methods. After localizing the face, as much information as possible about the displayed facial expression has to be extracted. In facial expression recognition, most automatic expression analysis systems attempt to recognize a small set of prototypic expressions (i.e. joy, surprise, anger, sadness, fear, and disgust).

Table 1. Universal emotion identification

Universal emotion identification		
Emotion	***Definition***	***Motion of facial part***
Anger	Anger shows the most dangerous emotion, it may be very harmful, humans are trying to avoid this emotion.	Eyebrows pulled down, Open eye, teeth shut and lips tightened, upper and lower lids pulled up.
Fear	Fear is the emotion of danger. It may be physical or psychological harms.	Outer eyebrow down, inner eyebrow up, mouth open, jaw dropped.
Happiness	Happiness is most desired expression by human.	Open Eyes, mouth edge up, open mouth, lip corner pulled up, cheeks raised, and wrinkles around eyes.
Sadness	Sadness is opposite emotion of Happiness.	Outer eyebrow down, inner corner of eyebrows raised, mouth edge down, closed eye, lip corner pulled down.
Surprise	This emotion comes when unexpected things happens.	Eyebrows up, open eye, mouth open, jaw dropped.
Disgust	Disgust is a feeling of dislike such as taste, smell, sound or tough.	Lip corner depressor, nose wrinkle, lower lip depressor, Eyebrows pulled down

2 Facial expression recognition

2.1 Feature Extraction Using Nonnegative Matrix Factorization

In this section, we describe the most important step in a facial expression recognition system which is feature extraction step that it can be analyzed in terms of facial action occurrence after the face has been located in the image or video frames. Over the past several decades, massive efforts have been made and remarkable achievements are obtained in FER. One key step in FER is to form or extract expression features from the original face images. Several wide used feature extraction methods such as Principal component analysis (PCA) [5], Eigen-face [6], Singular value decomposition (SVD) [7] Nonnegative matrix factorization (NMF) [8].

In NMF algorithm, it was described as follow: Given a nonnegative $m \times n$ matrix $\boldsymbol{X} = (x_0, x_1, \dots, x_{m-1}) \in R^{m \times n}$ is exactly the facial data which is going to analyzed, we have to find nonnegative matrix factor $\boldsymbol{U}$ $(m \times k)$ and matrix factor $\boldsymbol{V} = (k \times n)$ such that $\boldsymbol{X} \approx \boldsymbol{UV}$ where k is a smaller number compared to m and n.

A column vector in original matrix $\boldsymbol{X}$ can be considered as the weighted summation of all vectors in left matrix $\boldsymbol{U}$, while the weight coefficients are the elements of the corresponding column vector in the right matrix $\boldsymbol{V}$. The non-negativity constrains of $\boldsymbol{U}$ and $\boldsymbol{V}$ compatible with the intuitive notion of combining parts to form a whole, which is how NMF learns a part-based representation.

$$\min_{\mathbf{U},\mathbf{V}} f(\mathbf{U},\mathbf{V}) = \frac{1}{2}\|\mathbf{X} - \mathbf{UV}\|_F^2, s.t. \mathbf{U} \geq 0, \mathbf{V} \geq 0 \qquad (1)$$

where $\|.\|_F$ is Frobenius norm of a matrix, and the product $\mathbf{UV^T}$is the non-negative matrix factorization approximation of $\mathbf{X}$ of rank at most k. The non-negativity constraints on $\mathbf{U}$ and $\mathbf{V}$enables only additive (non-subtractive) combination of parts to construct the whole data.

In facial expression recognition application, the grey value of each facial image is nonnegative and stored in the computer as a form of matrix $\boldsymbol{X} = [X_1, X_2, \dots, X_n]$, where $\boldsymbol{X_j}$ is an m dimension column vector, which is made up the nonnegative grey facial expression image. The matrix $\boldsymbol{X}$ can be disassembled into the product of a nonnegative matrix $\boldsymbol{U}$ which represents the NMF basic images and a nonnegative weight coefficient matrix $\boldsymbol{V}$, NMF decomposition makes the reconstruction of expression images in a non-subtractive way and much similar to the process of forming unity from parts.

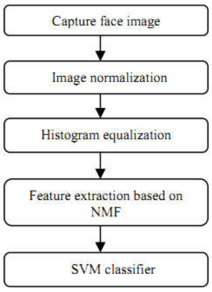

Fig. 2. Basic structure of facial expression analysis system [4]

2.2 Classification by SVM Classifier

The last part of the FER system is based on machine learning theory; exactly it is the classification task. The input to the classifier is a set of features which were retrieved from face region in the previous stage. The set of features is formed to describe the facial expression. Classification requires supervised training, so the training set should consist of labeled data.

Support vector machine (SVM) is a kind of pattern classification method based on the statistical learning theory, and designed in the principle of minimizing on construction risk. As for the binary linearly separable classification problem, SVM not only distinguish both classes with no errors, but also find the best separation line to make the largest margin between two classes. In higher dimension space, SVM becomes finding a best classification hyperplane for the high dimensional data. To resolve the multiclass problem, we use many sub-classifiers such as binary classifiers, and two regular strategies are one-against-one and one-against-all, as a result we find a multiclass classifier from a series of binary classifiers. For the C-class problem, one-against-one have to create $C(C-1)/2$ binary classifiers while one-against-all only have to create C binary classifiers.

2.3 Adaptive Feature Extraction and Classification Method

There is few existed works that use constrains the aim at increasing the discriminative power of the extracted features. Several variants of NMF with discriminant constraints imposed were proposed in [9, 10]. Kumar et al [11] introduced an adaptive feature extraction and classification method which proposed soft max-margin constraints to the objective function of NMF to obtain a bases matrix that maximized the classification margin using the features that are extracted using those bases. Inspired by this, they aim at finding a set of basis vectors that maximizes the margin of a SVM classifier.

Let $\{x_i, y_i\}_{i=1}^{L}$denote a set of data vectors and their corresponding labels, where $x_i \in R^m, y_i \in \{-1,1\}$. Our aim is to determine a bases matrix U that can be used to extract features that are optimal under a max-margin classification criterion. This is accomplished by imposing constraints on the feature vectors derived from U. In this work, the features that are extracted from a data example x are given by $\mathbf{U^TX}$. That is, they are the projections of the data example x on the bases vectors stored in U. Then, the optimization problem is given by

$$\min_{\mathbf{U},\mathbf{V},w,b,\varepsilon_i} \lambda\|\mathbf{X}-\mathbf{UV}\|_F^2 + \frac{1}{2}w^T w + C\sum_{i=1}^{L}\varepsilon_i \tag{2}$$

$$s.\mathrm{t.\,y_i(w^T}\mathbf{U}^\mathrm{T}\mathrm{x_i + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0{,}1 \leq i \leq L,}\mathbf{V} \geq 0$$

where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_i \ldots \varepsilon_L)$ is the lack variable vector, λ is a scalar that controls the relative importance for the NMF cost and C a scalar that controls the relative importance of the penalty imposed for the training examples that are either too close to the separating hyper-plane or misclassified.

3 Methodology

In this section, we introduce the unified objective function of proposed model which archives the upper objectives by combining the benefit of max-margin classifiers and NMF constraints together, through adding the pixel dispersion penalty and manifold regularization into the objective function. Following, we drive a multiplicative update rules using optimized gradient method and describe how the systems use this algorithm to perform the classification task we expect it to do.

3.1 Max-margin Nonnegative Matrix Factorization via Spatial Constraints and Graph Regularization

The unified objective function is constructed by jointing the data reconstruction objective function:

$$\min_{\mathbf{U},\mathbf{V},\mathrm{w},\mathrm{b},\varepsilon} ||\mathbf{X}-\mathbf{UV}||_F^2 + \lambda_1(\frac{1}{2}||w||_2^2 + C\sum_{i=1}^{n}\varepsilon_i) + \lambda_2 Tr(\mathbf{VLV^T}) + \lambda_3 Tr(\mathbf{UEU^T}) \qquad (3)$$

$$s.t.\,\mathbf{U}\geq 0, \mathbf{V}\geq 0, \varepsilon_i \geq 0, y_i(w^T v_i + b) \geq 1-\varepsilon_i, i = 1..n$$

All variables are divided into three terms: the coefficient matrix (V), the basis matrix (U) and variables about max-margin projection (w, b, ε). Where $||.||_F$ is Frobenius norm of a matrix, and the product **UV** is the non-negative matrix factorization approximation of **X** of rank at most k; $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_i \ldots \varepsilon_L)$ is the lack variable vector, λ_1 and C are scalars; the regularization parameter $\lambda_2 \geq 0$ controls the smoothness of the new representation; $\lambda_3 \geq 0, L \ll d$ and c_0 is a simple positive constant bound parameter; L is called graph Laplacian, E is called the dispersion kernel matrix.

Multiplicative Update Rules.

Update the Projection Vector and Slack Variables.

When the coefficient matrix and the basis matrix are fixed, MMNMF_MR optimization problem changes into the standard binary soft-margin SVM classification.

$$\min_{w,b,\varepsilon} \lambda_1(\frac{1}{2}||w||_2^2 + C\sum_{i=1}^{n}\varepsilon_i), s.t.\,\varepsilon_i \geq 0, y_i(w^T v_i + b) \geq 1-\varepsilon_i, i = 1..n \qquad (4)$$

The hyper-plane parameters w, b and slack variable vector ε are obtained using an off-the-shelf SVM classifier.

Update the Coefficient Matrix.

When other variables are fixed, the optimization of the coefficient matrix is transformed to quadratic programming:

$$\min_{\mathbf{V}} ||\mathbf{X}-\mathbf{UV}||_F^2 + \lambda_2 Tr(\mathbf{VLV^T}) \qquad (5)$$

$$s.t.\mathbf{V} \geq 0, y_i(w^T v_i + b) \geq 1 - \varepsilon_i, i = 1..n$$

The Lagrangian of above objective function is

$$L(V,\alpha,\beta) = Tr(\mathbf{X} - \mathbf{UV})(\mathbf{X} - \mathbf{UV})^T + \lambda_2 Tr(\mathbf{VLV^T}) - \alpha^T\mathbf{V} \\ - \sum_{i=1}^{n} \beta_i[y_i(w^T v_i + b) - 1 + \varepsilon_i]$$

where α, β are Lagrangian multipliers, specifically α is Lagrangian multipliers vector. Under the Karush-Kuhn-Tucker (KKT) conditions, we get

$$\begin{cases} 2\mathbf{U^T UV} - 2\mathbf{U^T}X + 2\lambda_2\mathbf{VL^T} - \alpha - \beta y w = 0 \\ 1^T V = 0 \\ y(w^T V - b) - 1 + \varepsilon = 0 \end{cases}$$

Transform the equation into a matrix for

$$\begin{pmatrix} 2\mathbf{U^T U} + 2\lambda_2\mathbf{L^T} & -1^T & -yw \\ 1^T & 0 & 0 \\ yw^T & 0 & 0 \end{pmatrix} \times \begin{pmatrix} \mathbf{V} \\ \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 2\mathbf{U^T X} \\ 0 \\ yb + 1 - \varepsilon \end{pmatrix}$$

where **1** is a unit vector whose size is the same as **v, 0** is the zero vector. We can derive **v** by solving this equation.

Update the Basis Matrix.

When other variables are fixed, the model is transformed to a non-negative matrix factorization:

$$O_3 = \min_U \|\mathbf{X} - \mathbf{UV}\|_F^2 + \lambda_3 Tr(\mathbf{UEU^T}), s.t.\mathbf{U} \geq 0 \qquad (6)$$

Because of the non-negative constraints, we use gradient descent methods to solve this problem. The gradient of equation (6) is

$$\nabla = 2\mathbf{UVV^T} - 2\mathbf{X^T V^T} + 2\lambda_3\mathbf{EU}$$

Classification.

During testing, the input test vector x_{test} is projected onto the basis matrix U to obtain the feature vector, $f_{test} = \mathbf{U}^T\mathbf{x}_{test}$. The feature vector is used by the max-margin classifier which predicts the class $\mathbf{y}_{test} = sign(\mathbf{U}^T f_{test} + b)$ where *w, b,* **U** are computed during training.

Fig. 3.

```
Algorithm for MNMF SGR
Input: Matrix X, rank k, maxIter; positive constants $\lambda_1,\lambda_2,\lambda_3$
Output: U, V, w, b
Begin
Initial the basis matrix U0 and the coefficient V0, let t=0
Let s=1, U=Us, V=Vs
Repeat
Fix U and Vs to find ws+1, bs+1 via equation (4)
Fix V, ws and bs+1 to find Us+1 via equation (5)
s=s+1
Until reaches the maximal iteration number;
Let t=t+1, Vt=Vs, wt=ws,bt=bs
Learning the new basis matrix Ut via minimizing equation (3.1)
End
```

Algorithm for Max-margin Nonnegative Matrix Factorization via Spatial Constrainst and Graph Regularization

4 Experiments

In this subsection, proposed MNMF_SGR method would be experimented and compared against several popular subspace learning algorithms, specifically the unsupervised methods (NMF [8], Spatial NMF [9] and Graph NMF [10]). We also compared with the supervised algorithm Semi-NMF [12] and Max-margin NMF [11].

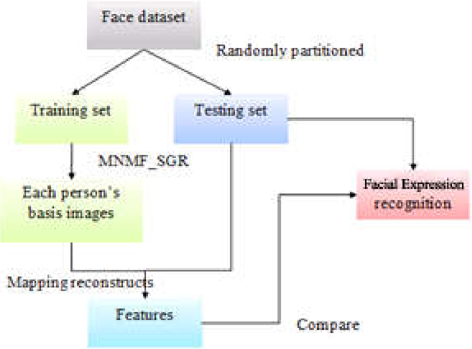

Fig. 4. The flow chart of MNMF_SGR based image reconstruction for facial expression recognition

4.1 Datasets

Japanese Female Facial Expression (JAFFE) database [13]: The database contains 213 images of 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models. Each image has been rated on 6 emotion adjectives by 60 Japanese subjects. The database was planned and assembled by Michael Lyons, Miyuki

Kamachi, and Jiro Gyoba. We thank Reiko Kubota for her help as a research assistant. The photos were taken at the Psychology Department in Kyushu University.

4.2 Preprocessing

Due to the background is larger than face image; firstly we apply the Viola-Jones algorithm to find the faces. For eyes, nose and mouth detection we applied cascaded object detector with region set on already detected frontal faces. Actually it uses Viola-Jones Algorithm as an underlying system. This preprocessing step is critical in achieving good classifier performance. Each original image from both databases is cropped and down-sampled in a such way that the final image size is 16 × 16 pixels.

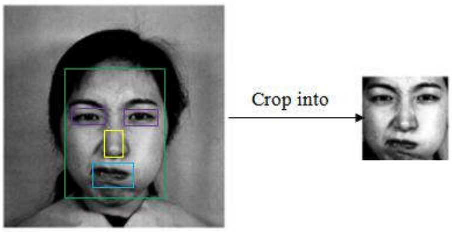

Fig. 5. Face and facial parts detection

All algorithms were initialized with 20 random U and V matrices, each of them was trained for 20 iterations and the one with the minimum objective function value was further trained for 1000 iterations.

4.3 Parameter Settings

For training and testing splits, we repeated the following procedure for ten times. Each time we randomly selected two-thirds of number of image per individual and labeled them. All the other images were unlabeled and used as the testing set.

In MNMF_SGR, λ_1 was tested for the following values {0.01, 1, 10} and λ_2 was tested for {1, 100} and λ_3 was tested for {10^{-5}, 10^{-4},..., 10^{2}}. Firstly, the dimensionality reduction process with NMF, SpaNMF, GNMF and Semi-NMF algorithms, the trained coefficient matrix is ready to be used for classifying a testing face image. Then we use SVM algorithm for the classifiers in the face recognition.

With MNMF and MNMF_SGR, after training process we compute the feature vector from the input test vector which is projected onto the basis matrix. After that, this feature vector is used in predicting class of face recognition. All algorithms were initialized with 20 random U and V matrices, each of them was trained for 20 iterations and the one with the minimum objective function value was further trained for 1000 iterations.

4.4 Classification Results.

The results of facial expression recognition for JAFFE dataset shown in Figure 6. Semi-supervised algorithms outperform all un-supervised ones. MNMF_SGR has highest accuracy, and then followed by MNMF_FA, MNMF, SpaNMF, GNMF, SemiNMF and standard NMF. MNMF_SGR outperforms NMF by 21.87%. The highest classification accuracy of 86.94% is achieved with k=30.

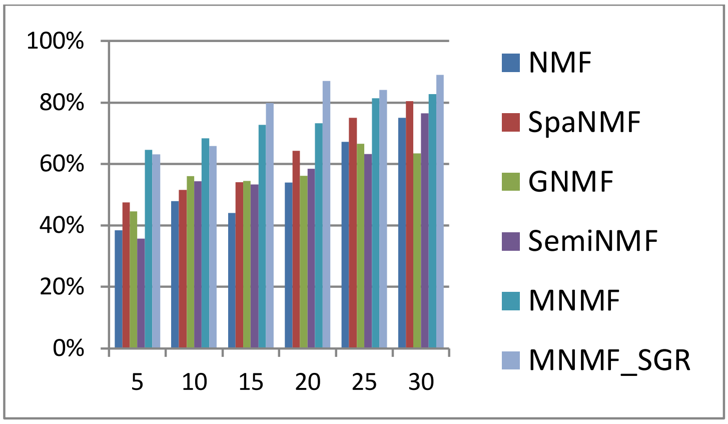

Fig. 6. JAFFE dataset facial expression recognition average accuracies (%) of different algorithms of five iterations

The confusion matrix of facial expression recognition shown in Table II using proposed method with 30 number of feature vectors (k=30). Some of the sad and happiness facial expression are confused with each other. The difference of happiness and sad failed because these expressions had a similar motion of mouth.

Table 2. Confusion matrix of 7-class facial expression recognition using proposed method MNMF_SGR on JAFFE

An	Di	Fe	Ha	Sa	Su	Ne
0.6268	0.1813	0.7285	0	0.0812	0.0322	0.0333
0.1234	**0.4510**	0.1009	0.0426	0.1268	0	0.0224
0	0.1191	**0.4825**	0.0918	0.0810	0	0.0421
0	0.1289	0.0199	**0.7490**	0.0211	0.0635	0.1289
0.1713	0.1270	0.1197	0.0423	**0.5343**	0.0524	0.1756
0	0	0	0.0203	0.0417	**0.7417**	0.0722
0	0.0000	0.0198	0.0486	0.1222	0.0711	**0.5221**

5 Conclusion

In this paper, constrained NMF approach had been introduced in the context of facial expression recognition. The proposed MNMF_SGR performs well in facial expression recognition task. This demonstrates the effectiveness of our model. To summarize, more constraints enable to build more effective models especially on high dimensional, sparse and noisy datasets. For future work more sophisticated and efficient way to tune kernel functions will be explored. We will also apply the proposed method to problems in other fields, such as bioinformatics and computer vision. Studying the convergence rate for MNMF_SGR and increasing the efficiency, they should be all in consideration.

References

1. B. Fasel, J. Luettin, "Automatic facial expression analysis: a survey", Pattern Recognition, 36 (1) (2003), pp. 259-275.
2. Ekman. P, Friesen W.V, "Constants across cultures in the face and emotion", Journal of Personality and Social Psychology, 17: 124–129, 1971.
3. L. Mai, "Joint Support Vector Machine with Constrained Nonnegative Matrix Factorization and Its Applications", Master Thesis, National Central University, July 2017.
4. Ying Zilu, Zhang Guoyi, "Facial Expression Recognition Based on NMF and SVM", IEEE: International Forum on Information Technology and Applications, pp. 612-615, 2009.
5. H. Moon and P. J. Phillips, "Computational and performance aspects of PCAbased face-recognition algorithms.," Perception, vol. 30, no. 3. pp. 303–21, Jan- 2001.
6. H. Wechsler, "Enhanced Fisher linear discriminant models for face recognition," Proceedings. Fourteenth Int. Conf. Pattern Recognit. (Cat. No.98EX170), vol. 2, pp. 1368–1372.
7. L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," Pattern Recognit., vol. 33, no. 10, pp. 1713–1726, Oct. 2000.
8. D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization", Advances in Neural Information Processing System, 2000.
9. W. S. Zheng, J. Lai, S. Liao, R. He, "Extracting non-negative basis images using pixel dispersion penalty", Pattern Recognition, pp. 2912-2926, 2012.
10. D. Cai, X. He, J. Han and T. Huang, "Graph regularized nonnegative matrix factorization for data representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33(8), pp. 1548–1560, 2011.
11. B. G. Kumar, I. Kotsia and I. Patras, "Max-margin nonnegative matrix factorization", Image Vision Computing, pp. 279–291, 2012.
12. C. Ding, T. Li, and M. Jordan, "Convex and semi-nonnegative matrix factorizations for clustering and low-dimension representation", Technical Report LBNL-60428, Lawrence Berkeley National Laboratory, University of California, Berkeley, 2006.
13. Evidence and a computational explanation of cultural differences in facial expression recognition. Matthew N Dailey, Carrie Joyce, Michael J Lyons, Miyuki Kamachi, Hanae Ishi, Jiro Gyoba, & Garrison W Cottrell Emotion, Vol 10(6), Dec 2010, 874-893.