A comparative study of NMF, DNMF, and LNMF algorithms applied for face recognition

Ioan Buciu, Nikos Nikolaidis and Ioannis Pitas

***Abstract*— Three techniques called non-negative matrix factorization (NMF), local non-negative matrix factorization (LNMF), and discriminant non-negative matrix factorization (DNMF), have been recently developed for decomposing a data matrix into non-negative factors named basis images and decomposition coefficients. Although these techniques are closely related to each other since they impose certain common non-negative constraints, the decomposition process of each algorithm involves a different objective function. While NMF approximates in the best possible way the data matrix by the product of its decomposition factors imposing only non-negative constraints, LNMF adds more constraints on the basis images to reduce the redundant information between them and to enlarge their sparseness degree. DNMF imposes more constraints on the coefficients in order to take into account class information. In this paper these methods are used in the context of face recognition to extract features from two image databases (YALE and ORL). Extracted features are further classified by two metric-based classifiers, namely Maximum Correlation Classifier (MCC) and Cosine Similarity Measure (CSM). Besides, Support Vector Machines (SVMs) are also used for classification. Experiments show that when these algorithms are applied along with the aforementioned classifiers, to face recognition task they lead to quite different results, their performance being data dependent.**

I. Introduction

Due to its wide range of applications that includes biometrics, information security, law enforcement and surveillance, etc., face recognition is a research topic heavily investigated by many researchers working on the field of computer vision. A survey on face recognition can be found in [1]. Although a variety of techniques have been proposed to cope with this task, their performance differs from one test image database to another. Face recognition is not an easy task since facial images can be recorded under various conditions such as non-ideal illumination conditions, the presence of occlusions, or different facial expressions. In this paper we compared three subspace methods named non-negative matrix factorization (NMF) [2], local non-negative matrix factorization (LNMF) [3], and discriminant non-negative matrix factorization (DNMF) [4] on the task of feature extraction from two image databases (YALE [5], and ORL [6]). The resulting features are further used to recognize faces from the aforementioned databases by employing two metrics-based classifiers: Maximum Correlation Classifier (MCC), Cosine Similarity Measure (CSM). Support Vector Machines (SVMs) are also used for classification. NMF has been tested in the context of face recognition in [7] along with the MCC. In this paper NMF is employed with CSM and SVM. The face recognition performance of LNMF and NMF has been compared in the ORL database in [3], and LNMF showed superior recognition performance over NMF. In this paper we investigate the behavior of these algorithms in the case of a another database namely the YALE face database. The recently introduced DNMF algorithm that was successfully applied to recognize facial expression [4] is also investigated for its potential to recognize faces.

II. Feature extraction

Suppose that each image from a database is stored in a m - dimensional column vector whose elements are the pixel values obtained by lexicographically scanning an image, i.e. $\mathbf{x} = [x_1, x_2, \ldots, \mathbf{x}_m]^T$, where m represents the total number of image pixels. Given n images $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, we can store them in a matrix $\mathbf{X}$ of size $m \times n$. Since pixel values correspond to grayscale levels the matrix $\mathbf{X}$ is non-negative. Non-negative matrix factorization (NMF) evaluates two non-negative factors, namely $\mathbf{W}$ (basis images) and $\mathbf{H}$ (coefficients) such that each image $\mathbf{x}_j$, for $j = 1, 2, \ldots, n$ is a linear combination of the basis images, i.e. $\mathbf{x}_j \approx \mathbf{W}\mathbf{h}_j$. Here $\mathbf{W}$ is a matrix $m \times p$ (whose columns store the basis images), and $\mathbf{h}$ is a $p \times 1$ vector comprising the linear decomposition coefficients. Due to the non-negativity constraint imposed in the decomposition process the reconstructed image is formed additive composition of basis images. The quality of approximation depends on the cost function associated to the decomposition. Two cost function have been proposed in [2]: Kullback-Leibler divergence $KL(\mathbf{x}||\mathbf{Wh})_{NMF} = \sum_i \left(x_i \log \frac{x_i}{\sum_k W_{ik} h_k} - x_i + \sum_k W_{ik} h_k \right)$ and squared Euclidean distance $D(\mathbf{x}||\mathbf{Wh})_{NMF} = \sum_i \|x_i - \sum_k W_{ik} h_k\|^2$ between $\mathbf{x}$ and its decomposition $\mathbf{Wh}$. When the first cost function is chosen to be minimized, the following multiplicative updating rules, found by applying a Expectation - Maximization (EM) approach, guarantee a non-increasing behavior of the cost function [2]:

$$h_{kj} = h_{kj} \frac{\sum_i w_{ki} \frac{x_{ij}}{\sum_k w_{ik} h_{kj}}}{\sum_i w_{ik}}. \tag{1}$$

$$w_{ik} = w_{ik} \frac{\sum_j \frac{x_{ij}}{\sum_k w_{ik} h_{kj}} h_{jk}}{\sum_j h_{kj}}. \tag{2}$$

Local non-negative matrix factorization (LNMF) has been developed by Li et al [3] in order to increase the basis images sparseness. Also, by appropriate modification of the cost function by imposing basis orthogonality the redundant

I. Buciu, N. Nikolaidis and I. Pitas are with the Department of Informatics, Aristotle University of Thessaloniki, GR-54124, Thessaloniki, Box 451, Greece {nelu,nikolaid,pitas}@aiia.csd.auth.gr

information captured by the basis images is minimized. The new cost function becomes:

$$D(\mathbf{X}||\mathbf{WH})_{LNMF} = KL(\mathbf{X}||\mathbf{WH})_{NMF} + \\ + \alpha \sum_{ik} u_{ik} - \beta \sum_{k} v_{kk} \tag{3}$$

where $[u_{jk}] = \mathbf{U} = \mathbf{W}^T\mathbf{W}$, $[v_{jk}] = \mathbf{V} = \mathbf{HH}^T$ and $\alpha,\ \beta > 0$ are constants. Therefore, the new function has to be minimized subject to three additional constraints: 1) $min \sum_j u_{jj}$ (to generate more localized features), 2) $min \sum_{j \neq k} u_{jk}$ (to minimize the redundancy between image bases) and 3) $max \sum_j v_{jj}$ (to maximize the total "activity"). The following update rules for the basis images and the coefficients, that are applied sequentially, are provided in [3]:

$$h_{kj} = \sqrt{h_{kj} \sum_i w_{ki} \frac{x_{ij}}{\sum_k w_{ik} h_{kj}}}. \tag{4}$$

$$w_{ik} = \frac{w_{ik} \sum_j \frac{x_{ij}}{\sum_k w_{ik} h_{kj}} h_{jk}}{\sum_j h_{kj}}. \tag{5}$$

$$w_{ik} = \frac{w_{ik}}{\sum_i w_{ik}}, \quad \text{for all } k. \tag{6}$$

Both NMF and LNMF consider the database as a whole and treat each image in the same way. There is no class information integrated into the cost function. An extension of LNMF algorithm called Discriminant Non-negative Matrix Factorization (DNMF) which takes into account class information has been proposed in [4]. If we have $\mathcal{Q}$ distinctive image classes and we denote by n_c the number of samples in class c, $c = 1, \ldots, \mathcal{Q}$, then each image from the image database (corresponding to one column of matrix $\mathbf{X}$) belongs to one of these classes. Therefore, each column of the $p \times n$ matrix $\mathbf{H}$ can be expressed as the image representation coefficients vector $\mathbf{h}_{cl}$, where $c = 1, \ldots, \mathcal{Q}$ and $l = 1, \ldots, n_c$. The total number of coefficient vectors is $n = \sum_{c=1}^{\mathcal{Q}} n_c$. We denote the mean coefficient vector of class c by $\boldsymbol{\mu}_c = \frac{1}{n_c} \sum_{l=1}^{n_c} \mathbf{h}_{cl}$ and the global mean coefficient vector by $\boldsymbol{\mu} = \frac{1}{n} \sum_{c=1}^{\mathcal{Q}} \sum_{l=1}^{n_c} \mathbf{h}_{cl}$. If we express the within-class scatter matrix by $\mathbf{S}_w = \sum_{c=1}^{\mathcal{Q}} \sum_{l=1}^{n_c} (\mathbf{h}_{cl} - \boldsymbol{\mu}_c)(\mathbf{h}_{cl} - \boldsymbol{\mu}_c)^T$ and the between-class scatter matrix by $\mathbf{S}_b = \sum_{c=1}^{\mathcal{Q}} (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T$, then, the cost function associated with DNMF algorithm is written as [4]:

$$D(\mathbf{X}||\mathbf{WH}) = KL(\mathbf{X}||\mathbf{WH})_{NMF} + \alpha \sum_{i,k} u_{ik} - \beta \sum_{k} v_{kk} + \\ + \gamma \mathbf{S}_w(h) - \delta \mathbf{S}_b(h), \tag{7}$$

subject to $\mathbf{W}, \mathbf{H} \geq 0$. Here γ and δ are constants. Both $\mathbf{S}_w(h)$ and $\mathbf{S}_b(h)$ are associated to the coefficient matrix. Notice that by eliminating the last two terms we obtain LNMF cost function. Obviously, by minimizing within-class scatter matrix we want the dispersion of samples that belong to the same class around their corresponding mean to be as small as possible, while by maximizing the between-class scatter matrix each cluster formed by the samples that belong to the same class is moved as far as possible from the other clusters.

In the light of the same EM strategy the DNMF algorithm updates the coefficients as follows [4]:

$$h_{kl(c)} = \frac{2\mu_c - 1}{4\xi} + \\ + \frac{\sqrt{(1 - 2\mu_c)^2 + 8\xi h_{kl(c)} \sum_i w_{ki} \frac{x_{ij}}{\sum_k w_{ik} h_{kl(c)}}}}{4\xi} \tag{8}$$

where ξ is a constant. The elements h_{kl} are then concatenated for all $\mathcal{Q}$ classes as:

$$h_{kj}^{(t)} = [h_{kl(1)} \,|\, h_{kl(2)} \,|\, \ldots \,|\, h_{kl(\mathcal{Q})}] \tag{9}$$

where "|" denotes concatenation. Since there is no change in the cost function related to the basis images with respect to LNMF their update follows the same rules (5) and (6).

III. Classification procedure

Let us now split the n face images into a training set $n^{(tr)}$ and a disjoint test set $n^{(te)}$ with the corresponding matrices $\mathbf{X}^{(tr)}$ and $\mathbf{X}^{(te)}$, respectively. The training images $\mathbf{X}^{(tr)}$ are used in the expression for evaluationg the decomposition factors. In the classical classification problem, we construct a classifier where the output (predicted value) of the classifier for a test image $\mathbf{x}^{(te)}$ is $\widetilde{l}$. Since $\mathbf{X}^{(tr)} = \mathbf{WH}$, the feature vectors used for classification are formed as $\mathbf{h}^{(tr)} = \mathbf{W}^{-1}\mathbf{x}^{(tr)}$, where $\mathbf{x}^{(tr)}$ is now a zero mean training face. A new test feature vector $\mathbf{h}^{(te)}$ is then formed as $\mathbf{h}^{(te)} = \mathbf{W}^{-1}\mathbf{x}^{(te)}$, where $\mathbf{x}^{(te)}$ is a zero mean test face. The recognition error is defined as the percentage of misclassified face images when $\{\widetilde{l}(\mathbf{h}^{(te)}) \neq l(\mathbf{h}^{(te)})\}$. Once we have formed $\mathcal{Q}$ classes of new feature vectors the following three classifiers are employed to classify a new test image:

1. *Cosine similarity measure* (CSM). This approach is based on the nearest neighbor rule and uses as similarity the angle between a test feature vector and a training one. We choose $\widetilde{l}_{CSM} = \text{argmin}_c\{d_c\}$, where $d_c = \frac{(\mathbf{h}^{(te)})^T \mathbf{h}^{(tr)}}{\|\mathbf{h}^{(te)}\| \|\mathbf{h}^{(tr)}\|}$ and d_c is the cosine of the angle between a test feature vector $\mathbf{h}^{(te)}$ and the training one $\mathbf{h}^{(tr)}$.

2. *Maximum correlation classifier* (MCC). The second classifier is a minimum Euclidean distance classifier. The Euclidean distance from $\mathbf{h}^{(te)}$ to $\mathbf{h}^{(tr)}$ is expressed as $\|\mathbf{h}^{(te)} - \mathbf{h}^{(tr)}\|^2 = -2g_c(\mathbf{h}^{(te)}) + (\mathbf{h}^{(te)})^T\mathbf{h}^{(te)}$, where $g_c(\mathbf{h}^{(te)}) = (\mathbf{h}^{(tr)})^T\mathbf{h}^{(te)} - \frac{1}{2}\|\mathbf{h}^{(tr)}\|^2$ is a linear discriminant function of $\mathbf{h}^{(te)}$. A test image is classified by this classifier by computing $\mathcal{Q}$ linear discriminant functions and choosing $\widetilde{l}_{MCC} = \text{argmax}_c\{g_c(\mathbf{h}^{(te)})\}$.

3. *Support vector machines* (SVMs). For SVMs the class membership for a new test vector $\mathbf{h}^{(te)}$ is given by the sign of the following decision function [8]:

$$f(\mathbf{h}^{(te)}) = \sum_{i=1}^{n^{tr}} \alpha_i \; l_i \; K(\mathbf{h}^{(te)}, \mathbf{h}_i^{(tr)}) + b \tag{10}$$

where $K(\mathbf{h}_1, \mathbf{h}_2)$ is a kernel function that defines the dot product between $\Phi(\mathbf{h}_1)$ and $\Phi(\mathbf{h}_2)$ in an higher-dimensional

Hilbert space $\mathcal{H}$, Φ denoting a nonlinear map $\Phi : \mathcal{R}^m \to \mathcal{H}$. α_i are nonnegative Lagrange multipliers associated with the quadratic optimization problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{y}^T\mathbf{y} + E\sum_{i=1}^{n^{(tr)}} \rho_i \\ \text{subject to} & l_i(\mathbf{y}^T\Phi(\mathbf{h}_i^{(tr)}) + b) \geq 1 - \rho_i, i = 1, \ldots, n^{(tr)} \end{array} \quad (11)$$

In (11), $\mathbf{y}$ and b are the parameters of the optimal hyperplane in $\mathcal{H}$ that attempts to separate the classes. That is, $\mathbf{y}$ is the normal vector to the hyperplane, $|b|/\|\mathbf{y}\|$ is the perpendicular distance from the hyperplane to the origin, with $\|\mathbf{y}\|$ denoting the Euclidian norm of vector $\mathbf{y}$. E is a parameter which penalizes the errors and ρ_i are positive slack variables. Frequently used kernel functions are the polynomial kernel, $K(\mathbf{h}_i, \mathbf{h}_j) = (\mathbf{h}_i^T\mathbf{h}_j + s)^q$ and the Exponential Radial Basis Function (ERBF) kernel, $K(\mathbf{h}_i, \mathbf{h}_j) = \exp\{-\gamma|\mathbf{h}_i - \mathbf{h}_j|\}$. We used $q = 1$ (equivalent to a linear classifier), $q = 2, 3, 4$ and $\gamma = 0.005$ in our experiments. To handle multi-class classification we chose the Decision Directed Acyclic Graph (DDAG) learning architecture proposed by Platt et al. [9].

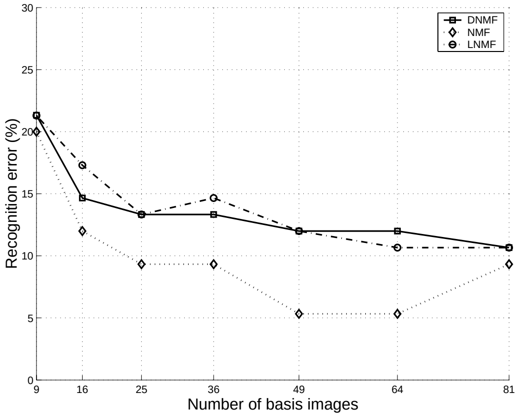

Fig. 1. Recognition error for YALE database and CSM classifier.

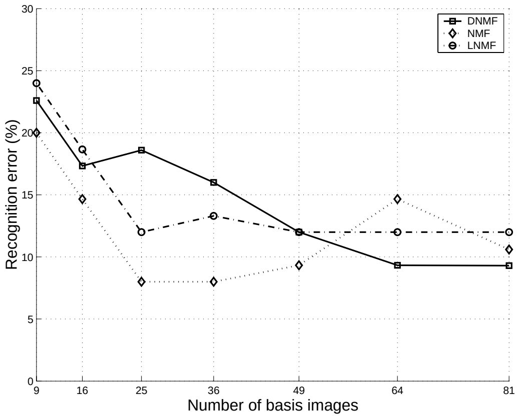

Fig. 2. Recognition error for YALE database and MCC classifier.

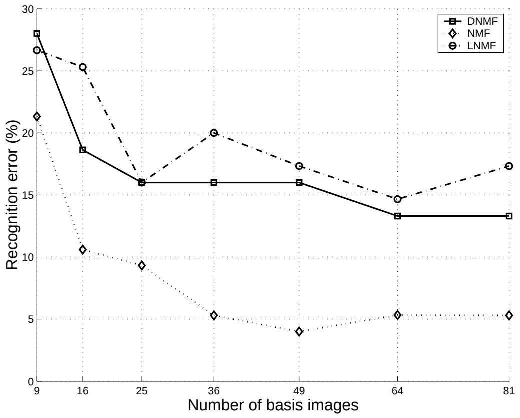

Fig. 3. Recognition error for YALE database and linear SVM classifier.

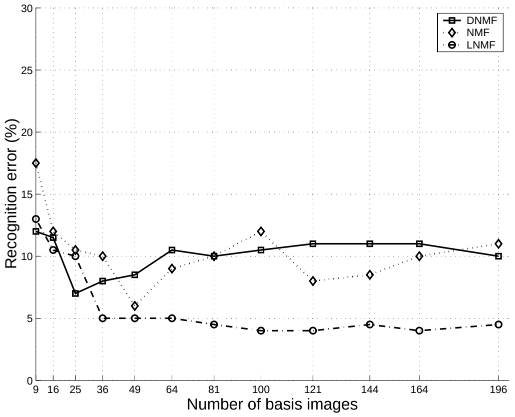

Fig. 4. Recognition error for ORL database and CSM classifier.

IV. Databases description

Two different public available databases have been chosen to work with. The Yale face database contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. For computational reasons the image size was reduced to 42×31 pixels. The second database used was the ORL face database that contains ten different images for forty distinct subjects. All images have been shot against a dark homogeneous background with the subjects in an upright, frontal position with tolerance for some side movement. For the experiments the images were downsampled to 42×31 pixels.

V. Experimental results

For the Yale database, the first six image samples of each subject were used to form the training data set while the remaining five samples were used as test images. In the case of the ORL database, the images have been randomly split in 200 samples for training data set and the rest 200 of them as test data set. Figures 1 and 2 depict the recognition error for the CSM and the MCC classifier, respectively, on the Yale database versus the number of basis images. The smallest errors is attributed to NMF algorithm in conjunction with the CMS metric. As far as SVMs are concerned, the best results were obtained by the linear kernel. Thus, we do not report

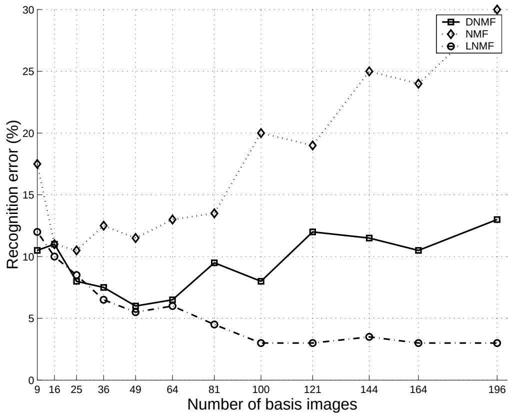

Fig. 5. Recognition error for ORL database and MCC classifier.

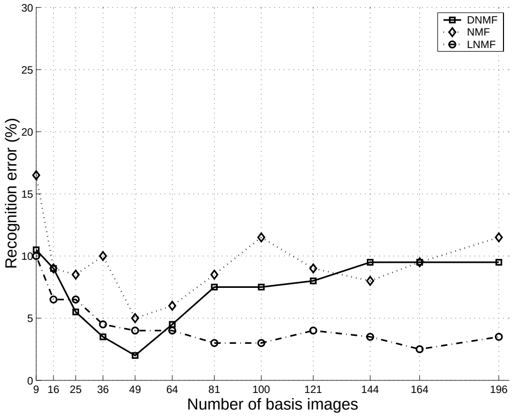

Fig. 6. Recognition error for ORL database and linear SVM classifier.

results obtained using polynomial and ERBF kernels. Figure 3 shows that NMF still outperforms both LNMF and DNMF, DNMF being the second best. Performing face recognition on ORL database lead to somehow opposite results. For this database, LNMF provided the lowest recognition error regardless of the classifier involved as can be seen from Figures 4, 5 and 6. NMF behaved the worst especially when it was associated with MCC classifier which is consistent with the results reported in [3]. DNMF algorithm is situated in the middle regarding its performance.

VI. Discussions and conclusion

The potential of NMF, LNMF and DNMF algorithms in the face recognition task has been investigated in this paper. Features extracted by these techniques from two different databases were classified by using two metric-based classifiers (CSM, MCC) and SVMs. The experiments showed that CSM and SVM yield better recognition performance than MCC. Overall, as we expected, SVMs performed the best, followed by CSM and MCC. As far as the feature extraction approach is concerned, based on the results obtained by the algorithms involved, it seems that NMF is more robust to illumination changes than LNMF and DNMF since the variation of lighting condition for the faces pertaining to Yale database is much more intense than for images from the ORL database. Contrary to the results obtained for ORL, where LNMF gave the highest recognition rate, when face recognition is performed on the YALE database the best results are obtained by the NMF algorithm. Although for the ORL database, generally, the faces are in frontal position, this database also contains poses where the face is slightly rotated. This can contribute to the performance of LNMF since this algorithm is rotation invariant (up to some degree) since it generates local features in contrast to NMF which yields more distributed features. Despite the fact that DNMF approaches was successfully applied to recognize facial expression, in the case of YALE database we found its performance to be inferior to NMF and slightly inferior when compared to LNMF applied for ORL database.

Acknowledgment

This work has been conducted in conjunction with the ”SIMILAR” European Network of Excellence on Multimodal Interfaces of the IST Programme of the European Union (www.similar.cc).

References

[1] W. Zhao, R. Chellappa, A. Rosenfeld and J. Phillips, “Face recognition: A literature survey,” *Technical Report*, CFAR-TR00-948, 2000.

[2] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” *Advances Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

[3] S. Z. Li, X. W. Hou and H. J. Zhang, “Learning spatially localized, parts-based representation,” *Int. Conf. Computer Vision and Pattern Recognition*, pp. 207–212, 2001.

[4] I. Buciu and I. Pitas, “A new sparse image representation algorithm applied to facial expression recognition,” in Proc. *IEEE Workshop on Machine Learning for Signal Processing*, pp. 539–548, 2004.

[5] http://cvc.yale.edu

[6] http://www.uk.research.att.com/

[7] D. Guillamet and Jordi Vitrià, “Non-negative matrix factorization for face recognition,” *Topics in Artificial Intelligence*, Springer Verlag Series: Lecture Notes in Artificial Intelligence, vol. 2504, pp. 336–344, 2002.

[8] V. Vapnik, *The nature of statistical learning theory*, Springer-Verlag, 1995.

[9] J. C. Platt, N. Cristianini, and J. S.-Taylor, “Large margin DAGs for mutliclass classification,” *Advances in Neural Information Procesing Systems*, vol. 12, pp. 547–553, 2000.