Multiplicative Update Rules for Concurrent Non-negative Matrix Factorization and Maximum Margin Classification

Olga Zoidi, Anastasios Tefas, *Member, IEEE*, Ioannis Pitas, *Fellow, IEEE*

***Abstract*—The state of the art classification methods which employ non-negative matrix factorization (NMF) employ two consecutive independent steps, the first one which performs data transformation (dimensionality reduction) and the second one which classifies the transformed data using classification methods, such as nearest neighbor/centroid or support vector machines (SVM). In the following, we focus on using NMF factorization followed by SVM classification. Typically, the parameters of these two steps, e.g. the NMF bases/coefficients and the support vectors are optimized independently, thus leading to suboptimal classification performance. In this paper, we merge these two steps into one by incorporating maximum margin classification constraints to the standard NMF optimization. The notion behind the proposed framework is to perform non-negative matrix factorization, while ensuring that the margin between the projected data of the two classes is maximal. The concurrent NMF factorization and support vector optimization are performed through a set of multiplicative update rules. In the same context, the maximum margin classification constraints are imposed on the NMF problem with additional discriminant constraints and respective multiplicative update rules are extracted. The impact of the maximum margin classification constraints on the NMF factorization problem is addressed in the experiments section. Experimental results in several databases indicate that the incorporation of the maximum margin classification constraints to the NMF and discriminant NMF objective functions improves the accuracy of the classification.**

***Index Terms*—Non-negative Matrix Factorization, Support Vector Machines, Joint Optimization, Maximum Margin Classification**

I. Introduction

Non-negative matrix factorization (NMF) is a popular method for representing a non-negative matrix $\mathbf{X} \in \Re^{N \times M}$, $X_{ij} \geq 0$ as a product of two other non-negative matrices:

$$\mathbf{X} = \mathbf{ZH}, \tag{1}$$

where $\mathbf{Z} \in \Re^{N \times L}$, $Z_{il} \geq 0$, $\mathbf{H} \in \Re^{L \times M}$, $H_{lj} \geq 0$. If we consider the matrix $\mathbf{X}$ as a data matrix, whose j-th column is the j-th element vector of dimension N, then the matrix $\mathbf{Z}$ can be considered as a basis matrix, whose l-th column is the l-th base vector of dimension N. The projection of the data to the space defined by $\mathbf{Z}$ are the columns of the coefficient matrix $\mathbf{H}$ with dimension L. By choosing $L << N$ we reduce the dimensionality of the original data. Therefore, NMF is often used as a data representation and dimensionality reduction technique for mainly image data in various applications in machine learning, computer vision and signal processing [1]-[5]. In this case, each j-th column $\mathbf{x}_j$ of matrix $\mathbf{X}$ represents an image of $N = N_x \times N_y$ pixels, scanned in a column-wise manner.

NMF was first employed by Lee and Seung in [1] for learning facial image parts and semantic text features. They also introduced two sets of multiplicative update rules for the estimation of the matrices $\mathbf{Z}$ and $\mathbf{H}$ in [6]. These update rules are derived from the minimization of a cost function which represents the error of the factorization. The factorization error is computed by either the Frobenius norm:

$$\|\mathbf{X} - \mathbf{ZH}\|_F^2 \tag{2}$$

or the Kullback-Leibler divergence [1]:

$$\sum_{ij} \left[x_{ij} \ln \left(\frac{x_{ij}}{\sum_l z_{il} h_{lj}} \right) - x_{ij} + \sum_l z_{il} h_{lj} \right] \tag{3}$$

The cost function is non-convex with respect to both variables $\mathbf{Z}$ and $\mathbf{H}$. However, it is convex with respect to either $\mathbf{Z}$ or $\mathbf{H}$. Therefore, the multiplicative update rules of [6] converge to a local minimum of the cost function. An extensive study on the convergence of the multiplicative update rules of NMF is presented in [7].

After NMF is performed on the original data, classification methods, such as Support Vector Machines (SVMs) [8], can be applied on the projected data. SVMs find the hyperplane in the high dimensional projection space, which has the maximum distance to the closest projected data of each class. This hyperplane is called a maximum-margin hyperplane. Consequently, SVMs are maximum-margin classifiers. As in the NMF case, SVMs optimize an objective function under certain constraints. In the case of linear classification, the formulation of the SVM optimization problem depends only on dot products of the data. By applying the kernel trick, the data are projected on a transformed feature space and the dot products are substituted by a non-linear kernel function. The maximum-margin hyperplane is still linear in the feature space but forms a non-linear surface in the original data space, hence achieving non-linear data classification. The number of Support Vectors of SVM classifiers can be reduced by the separable case approximation (SCA) algorithm [9]. SCA first

The authors are with the Department of Informatics, Aristotle University of Thessaloniki,Box 451, Thessaloniki 54124, GREECE, {tefas, pitas}@aiia.csd.auth.gr

The research leading to these results has received funding from the Collaborative European Project MOBISERV FP7-248434 (http://www. mobiserv.eu), An Integrated Intelligent Home Environment for the Provision of Health, Nutrition and Mobility Services to the Elderly.

computes Vapnik's SVM solution on the training data. Then, the training data are modified so that they become separable. Finally, the SVM is recomputed on the modified training data. Moreover, SVM has been combined with independent component analysis (ICA) in [10], in order to perform dimensionality reduction.

There are two ways for selecting the classification model. In-sample and out-of-sample methods. The in-sample approach employs the same data set for model selection and error estimation, while the out-of-sample approach employs two separate sets for training and validating the classification model. The majority of the classification models, including the proposed one, are selected based on out-of-sample methods. In [11], two in-sample methods for model selection and estimation of the SVM error are introduced, based on the data-dependent structural risk minimization (SRM), that outperform out-of-sample methods in the case where only a small number of data are available.

In the current state of the art methods, first data representation methods are optimized, in a way that enhance data separability and, then, classification methods are optimized in order to discriminate the projected data classes, thus leading in suboptimal classification performance. In this paper, we address the problem of data representation and classification optimization in a novel unified framework, i.e., our objective is to find the NMF data projections which maximize the classification accuracy of the SVM classifier. More precisely, we combine the NMF optimization problem and the dual formulation of the SVM optimization problem in a single objective function, under the non-negativity constraints of the NMF basis and coefficient matrices and the Lagrange multipliers of the SVM. An auxiliary function is found and minimized and multiplicative update rules for the NMF bases and coefficients and the Lagrange multipliers of SVM are extracted. Furthermore, we add the Fisher constraint of Discriminant NMF (DNMF) [12] in the proposed framework, leading to even more enhanced classification results.

The rest of the paper is organized as follows. Related works in the field are presented in Section II. A review on the theory of non-negative matrix factorization, discriminant non-negative matrix factorization and support vector machines is introduced in Section III. The proposed non-negative matrix factorization algorithm with maximum margin classification constraints is introduced in Section IV. The incorporation of the maximum margin classification constraints to the problem of discriminant non-negative matrix factorization is introduced in Section V. Section VI presents the experimental results. Finally, conclusions are drawn in Section VII.

II. Related Works

Several NMF modifications exist, which incorporate additional constraints to the initial problem of NMF, in order to enhance its discrimination ability. In [12] DNMF was introduced, which adds the Fisher constraint to the original cost function of NMF. The Fisher constraints maximize the distance between the mean class values and minimize class dispersion after data projection to the reduced dimensionality space. Experimental results on a facial expression recognition database showed that the basis images produced by the DNMF algorithm comprise a parts-based representation of the face, which corresponds better to the intuitive notion of facial regions, e.g. lips, eyebrows, than their NMF counterparts [12]. DNMF will be discussed more thoroughly in section III. The Fisher constraint was also employed by Wang et al. in [13]. They named their method Fisher NMF (FNMF). In [14], the principal components analysis constraint, i.e., the maximization of the coefficient matrix covariance, was added to the formulation of NMF, creating the so called PCA-NMF (PNMF). Similarly to the FNMF and the DNMF algorithms, when applied to facial images, the PNMF basis images can be interpreted as corresponding to facial regions. In [15], the original data matrix $\mathbf{X}$ is first mapped into a higher order reproducing Hilbert space by a non-linear polynomial kernel transformation $\mathbf{\Phi}(\cdot)$ and, then, NMF is applied on the produced feature space. The new cost function of the produced polynomial kernel-NMF (PKNMF) is given by:

$$\|\mathbf{\Phi}(\mathbf{X}) - \mathbf{Z}\mathbf{H}\|_F^2. \tag{4}$$

This way, the non-negative constraints of the basis images and the coefficients are maintained for features with non-linear dependencies.

Instead of formulating multiplicative update rules, the minimization of the NMF cost function (2) may also be achieved by employing projected gradient methods. Projected gradient methods for the problem of NMF were first employed in [16]. In [17], projected gradients are used to solve the problem of DNMF [12] with the difference that, instead of employing the Fisher constraint to the projection coefficient matrix $\mathbf{H}$ of the NMF algorithm, it is applied on the basis matrix $\mathbf{Z}$, by substituting $\mathbf{H}$ with $\mathbf{Z}^T\mathbf{X}$. Projected gradients are also used in [18], in order to extend the PKNMF algorithm for any kernel function. Other approaches in solving the NMF optimization problem include quadratic programming [19] or the gradient projection conjugate gradient (GPCG) algorithms [20]. The NMF computation of large-scale data sets is handled through online learning. In [21] an online algorithm for learning the NMF of large-scale datasets is introduced, called online robust stochastic approximation (RSA) NMF (OR-NMF). At each step, the method receives one new sample, computes its projection onto the learned subspace and, then, updates the NMF basis based on a RSA method.

Experimental results showed that the resulting basis images often form a sparse image representation [1]. However, this sparseness is not guaranteed [22]. Several methods have been proposed, which incorporate sparseness constraints to the cost function of NMF. The simple sparseness constraint used in [1] is the L_1-normalization of the basis vectors, i.e., to impose that the sum of the columns of the basis matrix to be equal to 1. A spatially localized NMF algorithm (LNMF) is introduced in [23]. The constraints of LNMF impose that the resulting basis vectors should be sparse and orthogonal to each other and the coefficient matrix should be non-sparse. Therefore, the basis matrix of LNMF is sparse and represents local image features. In [22], the sparseness level of the base matrix $\mathbf{Z}$ is regulated by measuring the relationship between the L_1-norm and the

L_2-norm of its columns:

$$\text{sparseness}(\mathbf{z}_l) = \frac{\sqrt{N} - \left(\sum_i |z_{il}| / \sqrt{\sum_i z_{il}^2}\right)}{\sqrt{N} - 1}. \quad (5)$$

The sparseness of the coefficient matrix $\mathbf{H}$ is regulated accordingly, by using the relationship between the L_1-norm and the L_2-norm of its rows. Sparseness regularization constraints were also imposed in [24], with additional prior information incorporated into the basis features. A non-smooth NMF (nsNMF) algorithm which imposes sparseness constraints is introduced in [25]. The idea behind nsNMF is that, by forcing non-negativity constraints to both the basis and coefficient matrices, the factorization performance is reduced, leading to bad data representation. nsNMF deals with this drawback by adding a third non-sparse matrix $\mathbf{S} \in \Re^{L \times L}$, called smoothing matrix, to the factorization formulation:

$$\mathbf{X} = \mathbf{ZSH}, \quad (6)$$

which enforces sparseness to both the bases and coefficient matrices. The smoothing matrix is positive and symmetric and regulates the level of sparseness to the bases and coefficient matrices. In [26], sparseness constraints are imposed on the bases and coefficient matrices, by employing sequential quadratic and second order cone programming. New multiplicative update rules for non-negative matrix factorization are introduced in [27], by substituting the Frobenius norm in (2) or the KL divergence in (3) with the Csiszár's divergence. Additional constraints, which regulate the sparseness of the basis and coefficient matrices may also be added to these methods.

In this paper, our objective is to find the NMF data projections which maximize the classification accuracy of the SVM classifier. To our knowledge, the first and only other attempt in combining NMF with SVM was introduced in [28]. The differences between the proposed framework and the one in [28] are:

a) we employ the dual formulation of SVM, which is the traditional method for solving the problem of SVM, instead of the primal problem of SVM used in [28], which requires strong assumptions for the derivation of the update rules of the coefficients matrix,

b) we employ hinge-loss SVMs instead of the least-squares SVMs used in [28].

III. Review of NMF and SVM optimization problems

In this section we review the basic theory on NMF and SVM optimization. More precisely, in Subsection III-A, we describe the problem of NMF and how the multiplicative update rules are derived. Furthermore, in Subsection III-B, we review how the Fisher constraint is applied to NMF for the formulation of Discriminant NMF (DNMF). Finally, we present the theory of Support Vector Machines (SVM) in Subsection III-C. These optimization problems form the current state of the art to be used for comparisons with the proposed novel framework in the experiments section.

A. The NMF algorithm

The objective of NMF is to find a pair of matrices $\mathbf{Z} \in \Re^{N \times L}$, $\mathbf{H} \in \Re^{L \times M}$, minimizing the cost function which measures the Frobenius norm of error between the initial data matrix $\mathbf{X} \in \Re^{N \times M}$, $x_{ij} \geq 0$, $i = 1, \ldots, N$, $j = 1, \ldots, M$ and its approximation by a matrix product $\mathbf{ZH}$:

$$\arg \min_{z_{il}, h_{lj}} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(x_{ij} - \sum_{l=1}^{L} z_{il} h_{lj} \right)^2, \quad (7)$$

subject to the constraints:

$$z_{il} \geq 0, \;\; h_{lj} \geq 0 \;\text{ and }\; \sum_{i=1}^{N} z_{il} = 1, \;\; \forall l = 1, \ldots, L. \quad (8)$$

We notice that (7) is the element-wise formulation of (2). Lee and Seung in [6] solved the optimization problem (7), by using the Expectation-Maximization (EM) algorithm [29][30], leading to the following multiplicative update rules:

$$h_{lj}^{(t+1)} = h_{lj}^{(t)} \frac{\sum_{i=1}^{N} x_{ij} z_{il}^{(t)}}{\sum_{i=1}^{N} \sum_{k=1}^{L} z_{il}^{(t)} z_{ik}^{(t)} h_{kj}^{(t)}} \quad (9)$$

$$z_{il}^{'(t+1)} = z_{il}^{(t)} \frac{\sum_{j=1}^{M} x_{ij} h_{lj}^{(t+1)}}{\sum_{k=1}^{L} \sum_{j=1}^{M} z_{ik}^{(t)} h_{kj}^{(t+1)} h_{lj}^{(t+1)}} \quad (10)$$

$$z_{il}^{(t+1)} = \frac{z_{il}^{'(t+1)}}{\sum_{i=1}^{N} z_{il}^{'(t+1)}}, \quad (11)$$

where the upper scripts (t) and $(t+1)$ denote the t-th and $(t+1)$-th iteration. It is obvious from the above formulation that, in each iteration, the updates (9)-(11) are performed sequentially. Furthermore, another set of multiplicative update rules is presented in [6], for minimizing of the Kullback-Leibler divergence:

$$\arg \min_{z_{il}, h_{lj}} \sum_{i=1}^{N} \sum_{j=1}^{M} \left[x_{ij} \ln \left(\frac{x_{ij}}{\sum_{l=1}^{L} z_{il} h_{lj}} \right) + \sum_{l=1}^{L} z_{il} h_{lj} - x_{ij} \right], \quad (12)$$

subject to the constraints given in (8). The corresponding multiplicative update rules, which are also derived by the EM algorithm, are given by (11) and:

$$h_{lj}^{(t+1)} = h_{lj}^{(t)} \frac{\sum_{i=1}^{N} z_{il}^{(t)} \frac{x_{ij}}{\sum_{l=1}^{L} z_{il}^{(t)} h_{lj}^{(t)}}}{\sum_{i=1}^{N} z_{il}^{(t)}} \quad (13)$$

$$z_{il}^{'(t+1)} = z_{il}^{(t)} \frac{\sum_{j=1}^{M} h_{lj}^{(t+1)} \frac{x_{ij}}{\sum_{l=1}^{L} z_{il}^{(t)} h_{lj}^{(t+1)}}}{\sum_{j=1}^{M} h_{lj}^{(t+1)}}. \quad (14)$$

According to [6], the two sets of multiplicative update rules converge to a locally optimal matrix factorization.

B. The DNMF algorithm

Additional constraints were imposed in [12] on the NMF optimization problem, in order to enhance data discrimination. DNMF is motivated by Fisher's linear discriminant and aims at minimizing the trace of the within-class scatter matrix $\mathbf{S}_W$ and maximizing the trace of the between-class scatter matrix

$\mathbf{S}_B$ of the projected data. Given that the columns $\mathbf{h}_j^c$ of the coefficient matrix $\mathbf{H}$ define the projected data, which belong to class c, the matrices $\mathbf{S}_W$ and $\mathbf{S}_B$ are defined by:

$$\mathbf{S}_W = \sum_{c=1}^{C}\sum_{j=1}^{M_c}(\mathbf{h}_j^c - \bar{\mathbf{h}}_c)(\mathbf{h}_j^c - \bar{\mathbf{h}}_c)^T \quad (15)$$

$$\mathbf{S}_B = \sum_{c=1}^{C} M_c(\bar{\mathbf{h}}_c - \bar{h})(\bar{\mathbf{h}}_c - \bar{\mathbf{h}})^T, \quad (16)$$

where C denotes the total number of classes, M_c is the cardinality of the data class c, $\bar{\mathbf{h}}_c$ is the mean vector of class c and $\bar{\mathbf{h}}$ is the mean vector of all classes. Finally, the objective function of DNMF is defined by:

$$\arg\min_{z_{il},h_{lj}} \sum_{i=1}^{N}\sum_{j=1}^{M}\left[x_{ij}\ln\left(\frac{x_{ij}}{\sum_{l=1}^{L} z_{il}h_{lj}}\right) + \sum_{l=1}^{L} z_{il}h_{lj} - x_{ij}\right] + \gamma \text{tr}[\mathbf{S}_W] - \delta \text{tr}[\mathbf{S}_B], \quad (17)$$

where tr$[\mathbf{A}]$ defines the trace of matrix $\mathbf{A}$. The parameters γ and δ determine the weight of the discriminant constraints tr$[\mathbf{S}_W]$ and tr$[\mathbf{S}_B]$ in the factorization problem. The classical NMF problem is a subcase of (17) for $\gamma = 0$ and $\delta = 0$. More details on the choice of γ and δ can be found in [12]. Since $\mathbf{S}_W$ and $\mathbf{S}_B$ depend only on the columns of matrix $\mathbf{H}$, the multiplicative update rules for the basis matrix $\mathbf{Z}$ remain the same with that of NMF algorithm, namely equations (14), (11). The update rule of $\mathbf{H}$ can be found by employing the EM algorithm:

$$h_{lj}^{(t+1)} = \frac{T_1 + \sqrt{T_1^2 + 4T_2 h_{lj}^{(t)} \sum_i z_{il}^{(t)} \frac{x_{ij}}{\sum_l z_{il}^{(t)} h_{lj}^{(t)}}}}{2T_2}, \quad (18)$$

where T_1, T_2 are given by:

$$T_1 = (2\gamma + 2\delta)\left(\frac{1}{M_c}\sum_{k,k\neq j}^{M_c} h_{lk}\right) - 2\delta\frac{1}{M}\sum_{k,k\neq j}^{M} h_{lk} - 1, \quad (19)$$

and

$$T_2 = 2\gamma - (2\gamma + 2\delta)\frac{1}{M_c} + 2\delta\frac{1}{M}. \quad (20)$$

C. Support Vector Machines

Let us consider a set $\mathcal{D} = \{\{\mathbf{x}_j, y_j\}, j = 1, ..., M, \mathbf{x}_j \in \Re^N, y_j \in \{-1, 1\}\}$ of M training data, where $\mathbf{x_j}$ denote the data points and y_j the corresponding labels. Our objective is to separate the positive data points from the negative ones, by finding the maximum-margin hyperplane, i.e., the hyperplane, whose distance from the nearest points of each class is maximal. Let us consider the vector $\mathbf{w} \in \Re^N$, which is normal to the hyperplane and the constant b, such as $|b|/\|\mathbf{w}\|$ is equal to the perpendicular distance between the hyperplane and the origin. The objective of SVM is the minimization of:

$$\arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \quad (21)$$

subject to the constraint:

$$y_j(\mathbf{w}^T\mathbf{x}_j - b) - 1 \geq 0, \forall j = 1, \ldots, M. \quad (22)$$

The mathematical analysis on the derivation of the objective function of SVM can be found in [8]. A point $\mathbf{x}_j$ is called a *support vector*, if the equality in (22) holds. Intuitively, support vectors are the points whose removal from the training data set would change the maximum-margin hyperplane. The solution to the problem of SVM is found by minimizing the Lagrangian function:

$$\arg\min_{\mathbf{w},b}\max_{a_j}\left\{\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{j=1}^{M} a_j y_j(\mathbf{w}^T\mathbf{x}_j - b) + \sum_{j=1}^{M} a_j\right\}, \quad (23)$$

subject to $a_j \geq 0$, where a_j denote the Lagrange multipliers. By computing the Karush-Kuhn-Tucker (KKT) conditions and substituting them to equation (23) we extract the Wolf dual formulation:

$$\arg\max_{a_j}\left\{\sum_{j=1}^{M} a_j - \frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{M} a_j a_k y_j y_k \mathbf{x}_j^T\mathbf{x}_k\right\}, \quad (24)$$

or equivalently:

$$\arg\min_{a_j}\left\{\frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{M} a_j a_k y_j y_k \mathbf{x}_j^T\mathbf{x}_k - \sum_{j=1}^{M} a_j\right\}. \quad (25)$$

We notice that equation (25) is a non-negative quadratic programming problem, which depends only on the Lagrange multipliers a_j. After we find the Lagrange multipliers, the maximum-margin hyperplane $\mathbf{w}$ and the margin b are estimated by using the following equations:

$$\mathbf{w} = \sum_{j=1}^{M} a_j y_j \mathbf{x}_j \quad (26)$$

$$b = \frac{1}{n(\mathcal{M}_{SV})}\sum_{j\in\mathcal{M}_{SV}}(\mathbf{w}^T\mathbf{x}_j - y_j), \quad (27)$$

where $\mathcal{M}_{SV}$ is the set of the indices of the support vectors and $n(\mathcal{M}_{SV})$ is their number. Finally, the decision about the class of a testing sample $\mathbf{x}$ is computed by the decision function:

$$\text{sign}(\mathbf{w}^T\mathbf{x} + b). \quad (28)$$

IV. Joint NMF and SVM optimization

In conventional methods, first the NMF algorithm is applied on the data and then, SVM is performed on the projected data, for the final data classification decision. In this section, we explore how the classification decision of SVM can influence the NMF factorization, in order to enhance the performance of a cascaded NMF/SVM framework. Intuitively, we want to find a base $\mathbf{Z}$, so that the projected data $\mathbf{h}_j$ belonging to the two classes minimize the reconstruction error of the original data $\mathbf{x}_j$ and they are separated with maximum margin by a certain hyperplane $\mathbf{w} = \sum_{j=1}^{M} a_j y_j \mathbf{h}_j$, which lies in the span of the projected data, according to the representer theorem [31].

In the standard NMF/SVM approach, first we solve the optimization problem given by (12) under the constraints (8) and, then, we solve the optimization problem in (25) where we replace the train data $\mathbf{x}_i$ with the NMF projections $\mathbf{h}_j$. By combining the two optimization problems (12), (25) in one,

we formulate a novel cost function, which must be minimized with respect to both z_{il}, h_{lj} and a_j:

$$F(z_{il},h_{lj},a_j)=\lambda\sum_{i,j}^{N,M}\left[x_{ij}\ln\left(\frac{x_{ij}}{\sum_l z_{il}h_{lj}}\right)+\sum_l z_{il}h_{lj}-x_{ij}\right]+\frac{1}{2}\sum_{jk}^{M}a_k a_j y_k y_j\sum_l^L h_{lj}h_{lk}-\sum_j^M a_j \quad (29)$$

subject to the constraints:

$$z_{il}\geq 0,\ \ h_{lj}\geq 0,\ \ a_j\geq 0,\ \text{ and }\ \sum_{i=1}^{N}z_{il}=1,\ \ \forall l=1,\ldots,L. \quad (30)$$

The factor λ in the NMF part of the cost function initially has a large value and then decreases at each iteration. This way, during the first iterations, the algorithm gives increased weight to the data representation, whereas, as time passes, its weight decreases exponentially according to $\lambda_0/(1+e)^t$, where t is the iteration number. Parameter $e<<1$ regulates the decrease rate, hence increasing the importance of the SVM part of the objective function. By employing the EM algorithm, the following theorem holds for the optimization problem (29):

Theorem 1: The cost function $F(z_{il},h_{lj},a_j)$ (29), subject to the constraints (30), is non-increasing under the following iterative update rules:

$$h_{lj}^{(t+1)}=\frac{-\lambda+\sqrt{\lambda+4\sum_k B_{jk}^{+}h_{lk}^{(t)}\left(\lambda\sum_i x_{ij}\frac{z_{il}}{\sum_m z_{im}h_{mj}^{(t)}}+\sum_k B_{jk}^{-}h_{lk}^{(t)}\right)}}{2\sum_k B_{jk}^{+}h_{lk}^{(t)}}h_{lj}^{(t)} \quad (31)$$

$$z_{il}^{'(t+1)}=\sum_j x_{ij}\frac{h_{lj}}{\sum_m z_{im}^{'(t)}h_{mj}}\frac{1}{\sum_j h_{lj}}z_{il}^{'(t)} \quad (32)$$

$$z_{il}^{(t+1)}=\frac{z_{il}^{'(t+1)}}{\sum_{i=1}^{N}z_{il}^{'(t+1)}} \quad (33)$$

$$a_j^{(t+1)}=\frac{1+\sqrt{1+4\sum_k A_{jk}^{+}a_k^{(t)}\sum_k A_{jk}^{-}a_k^{(t)}}}{2\sum_k A_{jk}^{+}a_k^{(t)}}a_j^{(t)} \quad (34)$$

where $A_{jk}=y_j y_k\sum_l h_{lj}h_{lk}$, $A_{jk}^{+}=\max(A_{jk},0)$, $A_{jk}^{-}=\max(-A_{jk},0)$, $B_{jk}=a_j a_k y_j y_k$, $B_{jk}^{+}=\max(B_{jk},0)$, $B_{jk}^{-}=\max(-B_{jk},0)$.

Since the cost function $F(z_{il},h_{lj},a_j)$ cannot be minimized directly, the proof of Theorem 1 is based on the minimization of three auxiliary functions $G(z_{il},z_{il}^{(t)})$, $G(h_{lj},h_{lj}^{(t)})$ and $G(a_j,a_j^{(t)})$. Before we proceed to the proof of Theorem 1 we need to write the following preliminary propositions. Their proofs are included in Appendix A.

Proposition 1: The cost function $F(z_{il},h_{lj},a_j)$ (29) subject to the constraints (30) is convex with respect to either z_{il}, h_{lj}, or a_j.

Let us define the functions $F(z_{il})=F(z_{il},h_{lj},a_j)|_{h_{lj},a_j=\text{constant}}$, $F(h_{lj})=F(z_{il},h_{lj},a_j)|_{z_{il},a_j=\text{constant}}$ and $F(a_j)=F(z_{il},h_{lj},a_j)|_{z_{il},h_{lj}=\text{constant}}$. Then Proposition 1 implies that the functions $F(z_{il})$, $F(h_{lj})$ and $F(a_j)$ are convex.

Definition 1: The function $G(h_{lj},h_{lj}^{(t)})$ is an auxiliary function for $F(h_{lj})$ if $G(h_{lj},h_{lj}^{(t)})\geq F(h_{lj})$ and $G(h_{lj},h_{lj})=F(h_{lj})$.

Theorem 2: If $G(h_{lj},h_{lj}^{(t)})$ is an auxiliary function for the cost function $F(h_{lj})$, then the minimization of $G(h_{lj},h_{lj}^{(t)})$ with respect to h_{lj} leads to minimization of $F(h_{lj})$. Consequently, $F(h_{lj})$ is monotonically decreasing under the update rule:

$$h_{lj}^{(t+1)}=\arg\min_{h_{lj}}\{G(h_{lj},h_{lj}^{(t)})\}. \quad (35)$$

The proof of Theorem 2 can be found in [32].

Proposition 2: The function:

$$G(a_j,a_j^t)=\lambda\sum_{ij}\left[x_{ij}\ln\left(\frac{x_{ij}}{\sum_l z_{il}h_{lj}}+\sum_l z_{il}h_{lj}-x_{ij}\right)\right]+\frac{1}{2}\sum_{jk}\frac{A_{jk}^{+}a_k^t}{a_j^t}a_j^2-\frac{1}{2}\sum_{jk}A_{jk}^{-}a_j^t a_k^t\left(1+\ln\frac{a_j a_k}{a_j^t a_k^t}\right)-\sum_j a_j, \quad (36)$$

where $A_{jk}=y_j y_k\sum_l h_{lj}h_{lk}$, $A_{jk}^{+}=\max(A_{jk},0)$ and $A_{jk}^{-}=\max(-A_{jk},0)$, is an auxiliary function for the cost function $F(a_j)$.

Proposition 3: The function:

$$G(z_{il},z_{il}^t)=\lambda\left[\sum_{ij}(x_{ij}\ln x_{ij}-x_{ij})-\sum_{ijl}x_{ij}\frac{z_{il}^t h_{lj}}{\sum_m z_{im}^t h_{mj}}\left(\ln z_{il}h_{lj}-\ln\frac{z_{il}^t h_{lj}}{\sum_m z_{im}^t h_{mj}}\right)+\sum_{ijl}z_{il}h_{lj}\right]+\frac{1}{2}\sum_{jk}^{M}a_k a_j y_k y_j\sum_l^M h_{lj}h_{lk}-\sum_j^M a_j \quad (37)$$

is an auxiliary function for the cost function $F(z_{il})$.

The proof of Proposition 3 is derived by the proof of the update rule of z_{il} in NMF which can be found in [6].

Proposition 4: The function:

$$G(h_{lj},h_{lj}^t)=\lambda\left[\sum_{ij}(x_{ij}\ln x_{ij}-x_{ij})-\sum_{ijl}x_{ij}\frac{z_{ij}h_{lj}^t}{\sum_m z_{im}h_{mj}^t}\left(\ln z_{il}h_{lj}-\ln\frac{z_{il}h_{lj}^t}{\sum_m z_{im}h_{mj}^t}\right)+\sum_{ijl}z_{il}h_{lj}\right]+\frac{1}{2}\sum_{ljk}\frac{B_{jk}^{+}h_{lk}^t}{h_{lj}^t}h_{lj}^2-\frac{1}{2}\sum_{ljk}B_{jk}^{-}h_{lj}^t h_{lk}^t\left(1+\ln\frac{h_{lj}h_{lk}}{h_{lj}^t h_{lk}^t}\right)-\sum_j a_j \quad (38)$$

where $B_{jk}=a_j a_k y_j y_k$, $B_{jk}^{+}=\max(B_{jk},0)$ and $B_{jk}^{-}=\max(-B_{jk},0)$ is an auxiliary function for the cost function $F(h_{lj})$.

Propositions 2, 3 and 4 prove that the functions (36), (37) and (38) are auxiliary functions for the cost function (29) with respect to a_j, z_{il} and h_{lj}, respectively. Therefore, the iterative update rules are extracted by setting the partial derivatives of the auxiliary functions to zero. The derivation of the update rules is given in Appendix B.

The multiplicative update rules of Theorem 1 are computed sequentially for every iteration, until the convergence of the cost function. Experimental results in image databases showed that the cost function converges to a local minimum in approximately 1000 iterations. The classification results are sensitive to λ. Typical values for λ_0 and e are $\lambda_0 = 100$ or 1000 and $e = 10^{-2}$. When the algorithm stops, the training data are projected to the space defined by the extracted basis matrix $\mathbf{Z}$, either by using the pseudo-inverse $\mathbf{Z}^\dagger = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$, the transpose $\mathbf{Z}^T$, or with more iterations of (31). Then, the new projections of the data are used in equations (26) and (27) in order to compute the resulting maximum-margin hyperplane and the margin, respectively. Finally, the test data are classified according to equation (28).

V. Joint DNMF and SVM optimization

The joint DNMF and SVM cost function is the following:

$$F(z_{il}, h_{lj}, a_j) = \lambda \sum_{i,j}^{N,M} \left[x_{ij} \ln \left(\frac{x_{ij}}{\sum_l z_{il} h_{lj}} \right) + \sum_l z_{il} h_{lj} - x_{ij} \right]$$
$$+\gamma tr(\mathbf{S}_W) - \delta tr(\mathbf{S}_B) + \frac{1}{2} \sum_{j,k=1}^{M} a_k a_j y_k y_j \sum_l h_{lk} h_{lj} - \sum_j a_j, \quad (39)$$

where the within-class and the between-class scatter matrices $\mathbf{S}_W$, $\mathbf{S}_B$ are given by (15) and (16), respectively. Since the additional terms in (39) depend only on h_{lj}, the update rules of z_{il} and a_j remain unchanged. The new update rule of h_{lj} is given by:

$$h_{lj} = \frac{-T_2 + \sqrt{T_2^2 + 4T_3 \left[\lambda \sum_i x_{ij} \frac{z_{il} h_{lj}^t}{\sum_m z_{im} h_{mj}^t} + \sum_k B_{jk}^- h_{jl}^t h_{lk}^t \right]}}{2T_3}, \quad (40)$$

where:

$$T_2 = \lambda - 2\gamma \tfrac{1}{M_r} \sum_{k=1,k\neq j}^{M_r} h_{lk} - 2\delta \tfrac{1}{M_r} \sum_{k=1,k\neq j}^{M_r} h_{lk} + 2\delta \tfrac{1}{M} \sum_{k=1,k\neq j}^{M} h_{lk} \quad (41)$$

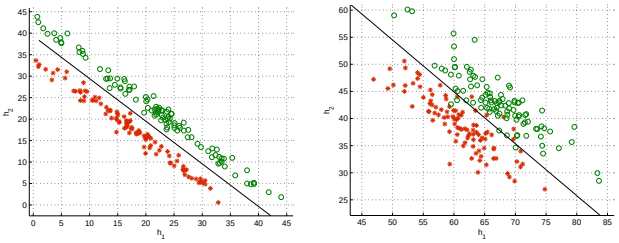
and:

$$T_3 = \sum_k \frac{B_{jk}^+ h_{lk}^t}{h_{lj}^t} + 2\gamma \left(1 - \frac{1}{M_r} \right) - 2\delta \left(\frac{1}{M_r} - \frac{1}{M} \right), \quad (42)$$

as proven in Appendix C. When the algorithm stops, the training data are projected on the space produced by the basis matrix $\mathbf{Z}$, the maximum-margin hyperplane $\mathbf{w}$ and the margin b are estimated from (26) and (27) and the classification decision of the test data is extracted according to (28).

VI. Experimental results

In this section, we examine the influence of the SVM constraint to the objective function of NMF and DNMF. At first, we present an experimental analysis on synthetic data. Then, the performance of the proposed methods is compared with the performance of the standard algorithms in six UCI databases. Furthermore, the proposed methods are employed on the Cohn-Kanade database for facial expression recognition and on the AIIA/MOBISERV database for eating and drinking activity recognition.



(a) (b)

Fig. 1. Projections of (a) joint NMF/SVM and (b) disjoint NMF+SVM iterative update rules

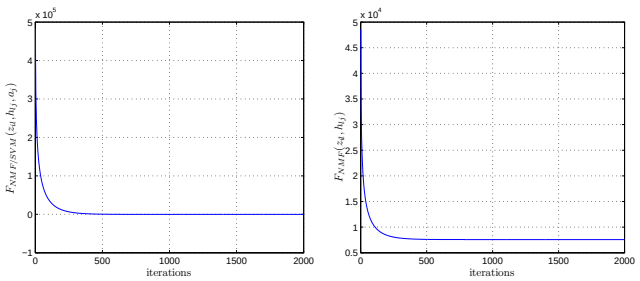
A. Synthetic Data

In this section, we analyze the proposed NMF/SVM algorithm on synthetic data in the 10-dimensional space coming from two classes, having Gaussian distributions with the same covariance matrix $\mathbf{I}$ (the identity matrix), but different mean vectors $\mathbf{m}_1 = 10 \cdot \mathbf{1}_{10}$ and $m_2 = 11 \cdot \mathbf{1}_{10}$, where $\mathbf{1}_{10}$ is the 10-dimensional vector of ones, which ensure positive data values. We notice that the two class samples are not linearly separable and overlap. For the sake of visualization, the chosen data are projected to the two-dimensional space. For each class we generate 100 training samples. Furthermore, we generate 75 samples of class 1 and 25 of class 2 for testing. The scatter plot of the coefficient matrix $\mathbf{H}$ after 5000 iterations of the update rules and the produced maximum-margin hyperplane are shown in Figure 1a. The maximum-margin hyperplane is computed from equations (26) and (27), where $\mathbf{x}_j$ are the columns of $\mathbf{H}$. We notice that the joint NMF and SVM enforces the linear separation of the training data. The classification accuracy in this case is 100%. In Figure 1b, the corresponding projections of the simple NMF methods and the resulting maximum margin hyperplane from the disjoint SVM method are depicted. We notice that, in this state of the art method, the projections $\mathbf{H}$ are no longer linearly separable. The classification accuracy in this case is 95.5%.

In Figure 2a the norm of the cost function of the joint NMF/SVM optimization problem is plotted. The plot shows that the cost function converges exponentially to some local minimum, which, in our case, is -217. Furthermore, the convergence of the first part of the cost function which corresponds to the cost of NMF and $\lambda = 1$ is shown in Figure 2b. It converges exponentially, until it reaches the local minimal value of 9030. The reason for which the cost function of NMF does not converge to zero is the SVM constraint, which dissociates the data of the two classes and prevents their proper reconstruction through the NMF bases and coefficients.

B. UCI Databases

In this section, we test the performance of the proposed NMF/SVM and DNMF/SVM methods in five UCI data sets and compare them against the standard disjoint optimization algorithms NMF+SVM and DNMF+SVM. We use the following two-class UCI data sets: liver disorders, ionosphere, hill-valley, Pima Indians Diabetes and Breast Cancer Wisconsin (Prognostic).



(a) (b)

Fig. 2. (a) Plot of cost function (29) of joint NMF/SVM. (b) Plot of the first part of the cost function of NMF/SVM which corresponds to the NMF error.

TABLE I
CLASSIFICATION ACCURACY (%) OF NMF+SVM, NMF/SVM, DNMF+SVM AND DNMF/SVM ALGORITHMS FOR VARIABLE L FOR THE LIVER DISORDERS DATA SET.

L	NMF+SVM	NMF/SVM	DNMF+SVM	DNMF/SVM
1	61.47	**66.47**	61.76	56.18
2	60.59	**67.65**	62.65	**67.65**
3	59.12	**68.24**	61.18	67.94
4	62.35	68.82	60.59	**69.41**
5	61.18	70.00	61.76	**71.18**
6	57.06	**71.47**	59.12	70.88

At first, we test the performance of the algorithms for varying dimensionality L of the projected data. For the experiments we use the liver disorders data set [33]. The classification performance is estimated through ten-fold-validation, i.e., the data set is partitioned into ten subsets, the nine subsets are used for training and the remaining for testing. The procedure is repeated 10 times, for all combinations of training and test subsets. At each iteration, the four factorization and classification algorithms NMF/SVM, DNMF/SVM, NMF+SVM, DNMF+SVM are performed and their classification accuracies are calculated. The final classification accuracy of each algorithm is computed by averaging the classification accuracies at each iteration. The classification accuracy for different values of the reduced dimensionality L are shown in Table I. We notice that the classification accuracy of the proposed NMF/SVM and DNMF/SVM algorithms increases by increasing dimensionality L. On the contrary, the classification accuracy of the state of the art NMF+SVM and DNMF+SVM methods does not seem to increase, as the dimension L increases. We also notice that, for all checked values of L, the proposed NMF/SVM method achieves better performance than the corresponding state of the art NMF+SVM algorithm. The same goes for the proposed DNMF/SVM vs the state of the art DNMF+SVM methods. Finally, by comparing the performances of the proposed NMF/SVM and DNMF/SVM methods we notice that, NMF/SVM achieves better classification performance, i.e., higher classification accuracy, than DNMF/SVM when the dimensionality L of the projected data is lower or equal to half the original data dimensionality $N = 6$. On the contrary, when the dimensionality L increases, DNMF/SVM achieves better performance.

Next, we test the general performance of the proposed and the state of the art algorithms in four other UCI data sets.

TABLE II
CLASSIFICATION ACCURACY (%) OF NMF+SVM, NMF/SVM, DNMF+SVM AND DNMF/SVM ALGORITHMS FOR SIX UCI DATA SETS

database	NMF+SVM	NMF/SVM	DNMF+SVM	DNMF/SVM
ionosphere	54.86±3.65	**81.14±1.71**	57.14±5.08	70.86±3.22
hill/valley	70.30±0.5	92.85±0.22	92.95±0.21	**93.00±0.26**
h/v noise	61.06±0.5	**90.92±0.1**	**90.92**±0.1	**90.92**±0.1
pima	80.53±8.41	**86.84±3.67**	85.26±3.91	71.58±8.58
wdbc	64.04±1.58	85.61±2.04	66.84±3.57	**86.49±2.5**
wpbc	94.74±0.94	**98.95±0.42**	97.89±0.7	97.89±0.7

Now, our goal is to compare the performance of the proposed algorithms with their corresponding state of the art methods for some L, in order to examine the effect of the imposition of the maximum margin constraint to the classification accuracy. For this reason, the weights γ and δ in the DNMF/SVM and DNMF+SVM algorithms are held constant and equal to 0.1 and 0.05, respectively.

The first data set used is the Ionosphere data set [34], where we reduce the dimensionality of the projected data from $N = 34$ to $L = 2$. As in the liver data set, ten-fold-cross-validation is employed for calculating classification performance. The results are shown in the first row of Table II. We notice that the mean classification accuracy of the state of the art methods NMF+SVM and DNMF+SVM is low, due to the very small dimensionality value $L = 2$. On the contrary, the proposed NMF/SVM and DNMF/SVM methods achieve much higher classification accuracy and smaller standard deviation, especially NMF/SVM, whose error rate (18.86%) is less than half that of the NMF+SVM algorithm (45.14%).

The second data set used is the Hill-Valley data set [33], which contains artificial data of plots, which create either a Hill (a bump in the terrain) or a Valley (a dip in the terrain). The data dimensionality is $N = 100$. The data set has two versions, one containing smooth plots and one containing plots contaminated by noise. We perform the experiment to both the noiseless and the noisy data sets ten times with different initializations of matrices $\mathbf{Z}$ and $\mathbf{H}$. The data dimensionality is reduced to $L = 10$. The results are shown in the second and third row of Table II. We notice that, in both cases of noisy and noiseless data sets, the algorithms NMF/SVM, DNMF+SVM and DNMF/SVM have similar standard deviation and the same mean classification accuracy which is equal to 92.74% for the noiseless case and 90.92% for the noisy case. On the other hand, the classification accuracy of NMF+SVM is much lower (70.30% and 61.06% for the noiseless and noisy cases, respectively).

The third data set used is the Pima Indians Diabetes data set [35] where we reduced data dimensionality from $N = 8$ to $L = 2$. The classification performance was calculated again through ten fold cross validation. The experimental results are given in the fourth row of Table II. We notice that the proposed NMF/SVM method has higher mean classification accuracy (86.84%) than the state of the art NMF+SVM (80.53%) and smaller standard deviation. On the contrary, the proposed DNMF/SVM algorithm has lower mean classification accuracy (71.58%) than the state of the art DNMF+SVM. This is the only case in our experiments, when the standard method

predominates the proposed one.

Finally, the classification performance is tested on the Breast Cancer Wisconsin (Prognostic) data set [36]. It consists of two data sets: the Wisconsin Prognostic Breast Cancer (WPBC) data set and the Wisconsin Diagnostic Breast Cancer (WDBC) data set. In both data sets, ten fold cross validation is performed, while the reduced dimensionality is $L = 2$. The results are shown in the last two rows of Table II. We notice that, in these data, the proposed algorithm has a higher classification accuracy than the corresponding standard algorithm, except once in the WPBC data set, where the DNMF/SVM method achieves the same classification accuracy with the DNMF+SVM method.

C. Cohn-Kanade Database

In this section, we test the performance of the proposed NMF/SVM and DNMF/SVM algorithms in the problem of facial expression recognition. In facial expression recognition, our goal is the recognition of the six universal expressions, as they were defined by Ekman [37]: anger, disgust, fear, happiness, sadness and surprise. The experiments were conducted in the Cohn-Kanade database [38]. The Cohn-Kanade database was released in 2000. It consists of 486 image sequences from 97 University students aged between 18 to 30 years, performing the six universal expressions. Female students were 65% of the subjects, 15% were African-American and 3% originated from Asia or Latin America. Each image sequence starts from the neutral expression and evolves until it reaches an expression apex. The expression apex is coded according to the Facial Action Codding System (FACS) and is assigned an emotion label that indicates the target expression (and not the expression that was actually performed).

In our experiments we used the five-fold cross validation method, i.e., each class data were partitioned into five complementary subsets and, in each iteration (of the total five), one subset was left for testing and the rest were used for training. Since the proposed framework is a two-class one and the experimental data are multi-class, we decompose the multi-classification task into two-class classification sub-tasks, by selecting 2-combinations from the set of 6 facial expressions. For each sub-task we perform five-fold cross validation and, finally, the overall classification accuracy of each method is estimated by averaging the accuracies of each run. The size of the initial images is 30×40 pixels, meaning that the original data dimensionality is $N = 1200$. In our experiment we reduce the data dimensionality to $L = 100$. The first 25 basis images of the proposed NMF/SVM and DNMF/SVM methods and the standard NMF+SVM and DNMF+SVM methods are depicted in Figure 3. From Figure 3 we notice that, for the proposed NMF/SVM and DNMF/SVM methods, the basis images sparseness is greater than that of NMF and lower than that of DNMF. Finally, the error rates of the four methods are shown in Table III. We notice that the highest classification accuracy is achieved for the proposed DNMF/SVM method (78.10%), followed by the standard DNMF+SVM method (75.71%), while the proposed NMF/SVM algorithm with classification accuracy 74.77% only marginally improves the accuracy of the corresponding NMF+SVM algorithm, which is 74.48%

TABLE III
CLASSIFICATION ACCURACY (%) OF NMF+SVM, NMF/SVM, DNMF+SVM AND DNMF/SVM ALGORITHMS FOR THE COHN-KANADE DATABASE

NMF+SVM	NMF/SVM	DNMF+SVM	DNMF/SVM
74.48%	74.77%	75.71%	$\mathbf{78.10}\%$

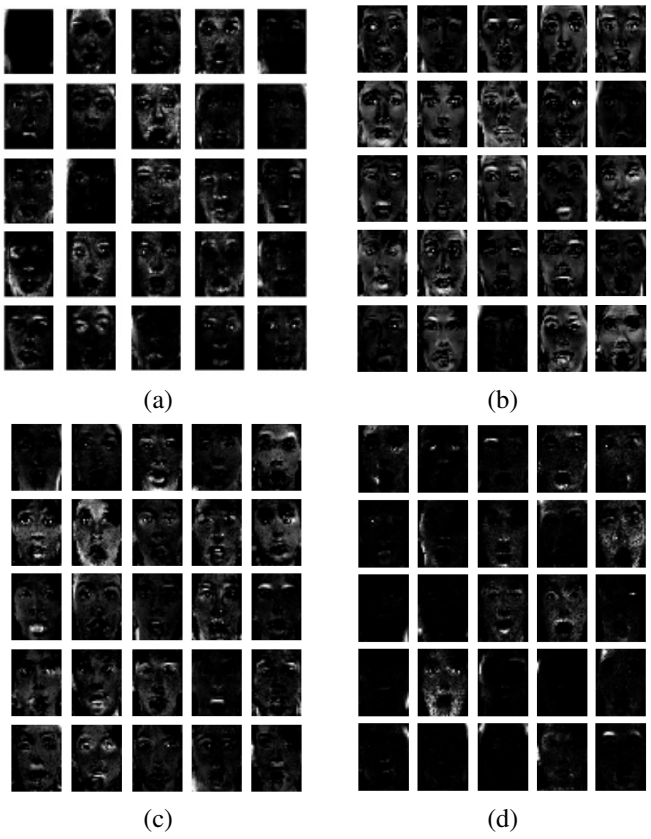


Fig. 3. A set of 25 basis images for (a) NMF/SVM, (b) NMF+SVM, (c) DNMF/SVM, (d) DNMF+SVM.

D. AIIA/MOBISERV database

In this section we test the performance of the proposed NMF/SVM and DNMF/SVM algorithms in the problem of activity recognition, namely eating and drinking activity recognition. This problem is an important one in automatic nutrition support/reporting systems for frail groups, such as patients and elder population. The experiments were conducted in the AIIA/MOBISERV eating and drinking activity recognition database [39]. It consists of videos depicting 12 subjects, 6 male and 6 female, during four meal sessions recorded in four different days. In each session, the subject performs eating and drinking activities in all possible ways: eating with a spoon, or a fork, knife and fork, with one hand, or with both hands; drinking from a cup, or from a glass, or from a glass with a straw. In each video, the skin color information is used, in order to extract the area of the hands and face, creating binary masks, as the ones shown in Figure 4. [40]. Finally, the Motion History Images (MHI) [41] of each activity are extracted and

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4. Masks of video frames depicting (a) eating with spoon, (b) eating with knife and fork, (c) eating with fork, (d) eating with one hand, (e) eating with two hands, (f) drinking from cup, (g) drinking from glass, (h) drinking from glass with straw.

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5. MHIs of videos depicting (a) eating with spoon, (b) eating with knife and fork, (c) eating with fork, (d) eating with one hand, (e) eating with two hands, (f) drinking from cup, (g) drinking from glass, (h) drinking from glass with straw.

down-scaled to 32×32 pixels. In total, 3969 MHIs where created. Examples of these MHIs for each activity are depicted in Figure 5.

Then, we tested the performance of the proposed algorithms using the leave-one-day-out cross validation method: in each iteration, we use the MHIs of three days for training and the MHIs of the remaining day for testing. Finally, the classification accuracy of the four methods are shown in Table IV. We notice that the highest classification accuracies are achieved for the proposed NMF/SVM (79.42%) and DNMF/SVM (79.30%) methods. The error of the standard NMF+SVM (21.68%) is 5.34% higher than the one of the proposed NMF/SVM method (20.58%), while the standard DNMF+SVM has the lowest classification accuracy (64.60%).

VII. Conclusion and future work

In this paper, we propose novel multiplicative update rules for performing joint non-negative matrix factorization and maximum margin classification. We introduce two sets of update rules. The first set incorporates the maximum margin constraint in the objective function of the standard NMF. The second set, incorporates the maximum margin constraint in the objective function of discriminant NMF. Experimental results in various real-data sets showed that the use of the maximum margin constraint enhances the performance of the standard factorization and classification framework.

An important variant of SVM classifiers are the kernel SVMs, which perform classification of non-linear data. Kernel SVMs employ the kernel trick for mapping the data on a transformed feature space where, hopefully, the projected data will be linearly separable. In this case, the inner product of the data projections in (29) will be replaced by the kernel function. However, the unknown form of the kernel prohibits us from following the same procedure in order to derive closed-form update rules for the data projections, as in the linear case. This is a serious limitation of the proposed method. In this case, other approaches such as the projected gradients can be employed, for derivation of the update rules. Moreover, the proposed framework can be extended for the case of joint non-negative matrix factorization and multi-class maximum margin classification. It has been proven that the Wolf dual formulation of the linear multi-class SVM problem can be written in quadratic form similar to (25) [42]. Therefore, the extension of the proposed two-class framework on the multi-class case is straight forward. These topics are the subject of ongoing research.

TABLE IV
Classification accuracy (%) of NMF+SVM, NMF/SVM, DNMF+SVM and DNMF/SVM algorithms for the AIIA/MOBISERV Database

NMF+SVM	NMF/SVM	DNMF+SVM	DNMF/SVM
78.32%	**79.42%**	64.60%	79.30%

Appendix A
Proof of Propositions 1,2,4

A. Proof of Proposition 1:

A function $F(x)$ is convex if and only if $\partial^2 F(x)/\partial x^2 \geq 0$, $\forall x \geq 0$. In our case, the following equations hold:

$$\frac{\partial^2 F(z_{il})}{\partial z_{il}^2} = \lambda \sum_j \frac{x_{ij} h_{lj}^2}{\left(\sum_k z_{ik} h_{kj}\right)^2} \geq 0, \quad \forall z_{il} \geq 0 \quad (43)$$

$$\frac{\partial^2 F(h_{lj})}{\partial h_{lj}^2} = \lambda \sum_i \frac{x_{ij} z_{il}^2}{\left(\sum_k z_{ik} h_{kj}\right)^2} + a_j^2 \geq 0, \quad \forall h_{lj} \geq 0 \quad (44)$$

$$\frac{\partial^2 F(a_j)}{\partial a_j^2} = \sum_l h_{lj}^2 \geq 0, \quad \forall a_j \geq 0. \quad (45)$$

Therefore, Lema 1 holds.

B. Proof of Proposition 2:

The proof of the second condition of Definition 1 $G(a_j, a_j) = F(a_j)$ is straightforward. In order to prove the first condition, we compute the difference $G(a_j, a_j^t) - F(a_j)$:

$$G(a_j, a_j^t) - F(a_j) = \frac{1}{2} \sum_{jk} \frac{A_{jk}^+ a_k^t}{a_j^t} a_j^2 -$$

$$-\frac{1}{2} \sum_{jk} A_{jk}^- a_j^t a_k^t \left(1 + \ln \frac{a_j a_k}{a_j^t a_k^t}\right) - \frac{1}{2} \sum_{jk}^{M} A_{kj} a_k a_j. \quad (46)$$

The proof of the positiveness of equation (46) can be found in [43].

C. Proof of Proposition 4:

It is straightforward to show that $G(h_{lj}, h_{lj}) = F(h_{lj})$ which means that the second condition of definition 1 holds. In order to prove the first condition, we write:

$$G(h_{lj}, h_{lj}^{(t)}) = G_1(h_{lj}, h_{lj}^{(t)}) + G_2(h_{lj}, h_{lj}^{(t)}), \quad (47)$$

where

$$G_1(h_{lj}, h_{lj}^{(t)}) = \lambda \left[\sum_{ij} (x_{ij} \ln x_{ij} - x_{ij}) - \right.$$
$$- \sum_{ijl} x_{ij} \frac{z_{ij} h_{lj}^t}{\sum_m z_{im} h_{mj}^t} \left(\ln z_{il} h_{lj} - \ln \frac{z_{il} h_{lj}^t}{\sum_m z_{im} h_{mj}^t} \right) +$$
$$\left. + \sum_{ijl} z_{il} h_{lj} \right], \quad (48)$$

$$G_2(h_{lj}, h_{lj}^{(t)}) = \frac{1}{2} \sum_{ljk} \frac{B_{jk}^+ h_{lk}^t}{h_{lj}^t} h_{lj}^2 -$$
$$- \frac{1}{2} \sum_{ljk} B_{jk}^- h_{lj}^t h_{lk}^t \left(1 + \ln \frac{h_{lj} h_{lk}}{h_{lj}^t h_{lk}^t} \right) - \sum_j a_j, \quad (49)$$

and $\mathbf{B}_{jk} = a_j a_k y_j y_k$, $\mathbf{B}^+ = \max\{\mathbf{B}, 0\}$, $\mathbf{B}^- = -\min\{\mathbf{B}, 0\}$. Accordingly, we write:

$$F(h_{lj}) = F_1(h_{lj}) + F_2(h_{lj}), \quad (50)$$

where

$$F_1(h_{lj}) = \lambda \sum_{i,j}^{N,M} \left[x_{ij} \ln \left(\frac{x_{ij}}{\sum_l z_{il} h_{lj}} \right) \sum_l z_{il} h_{lj} - x_{ij} \right], \quad (51)$$

$$F_2(h_{lj}) = \frac{1}{2} \sum_{jk}^{M} a_k a_j y_k y_j \sum_l^L h_{lj} h_{lk} - \sum_j^M a_j \quad (52)$$
$$= \frac{1}{2} \sum_l^L \sum_{jk}^M h_{lj} a_k a_j y_k y_j h_{lk} - \sum_j a_j \quad (53)$$
$$= \frac{1}{2} \sum_l^L \sum_{jk}^M h_{lj} \mathbf{B} h_{lk} - \sum_j a_j \quad (54)$$
$$= \frac{1}{2} \sum_l^L \tilde{\mathbf{h}}_l^T \mathbf{B} \tilde{\mathbf{h}}_l - \sum_j a_j, \quad (55)$$

where $\tilde{\mathbf{h}}_l \in \Re^{M \times 1}$ denotes the vector whose elements are the l-th row of matrix $\mathbf{H}$. The proof will be complete if we show that the following inequalities hold:

$$G_1(h_{lj}, h_{lj}^{(t)}) \geq F_1(h_{lj}) \quad (56)$$
$$G_2(h_{lj}, h_{lj}^{(t)}) \geq F_2(h_{lj}). \quad (57)$$

The proof of inequality (56) can be found in [6] and, for a given l, the proof of inequality (57) can be derived by [43].

APPENDIX B
DERIVATION OF THE NMF/SVM UPDATE RULES

Update rule for Lagrange multipliers $\mathbf{a}$:

$$\frac{\partial G(a, a^t)}{\partial a_j} = 0 \quad (58)$$
$$\sum_k \frac{A_{jk}^+ a_k^t}{a_j^t} a_j - \sum_k A_{jk}^- a_j^t a_k^t \frac{1}{a_j} - 1 = 0 \quad (59)$$
$$\sum_k \frac{A_{jk}^+ a_k^t}{a_j^t} a_j^2 - a_j - \sum_k A_{jk}^- a_j^t a_k^t = 0 \quad (60)$$
$$a_j^{t+1} = \frac{1 + \sqrt{1 + 4 \sum_k A_{jk}^+ a_k^t \sum_k A_{jk}^- a_k^t}}{2 \sum_k A_{jk}^+ a_k^t} a_j^t \quad (61)$$

Update rule for coefficient matrix $\mathbf{H}$:

$$\frac{\partial G(h, h^t)}{\partial h_{lj}} = 0 \quad (62)$$
$$\lambda \left[- \sum_i x_{ij} \frac{z_{il} h_{lj}^t}{\sum_m z_{im} h_{mj}^t} \frac{1}{h_{lj}} + \sum_i z_{il} \right] +$$
$$+ \sum_k \frac{B_{jk}^+ h_{lk}^t}{h_{lj}^t} h_{lj} - \sum_k B_{jk}^- h_{lj}^t h_{lk}^t \frac{1}{h_{lj}} = 0 \quad (63)$$
$$\sum_k \frac{B_{jk}^+ h_{lk}^t}{h_{lj}^t} h_{lj}^2 + \lambda \sum_i z_{il} h_{lj} -$$
$$- \left[\lambda \sum_i x_{ij} \frac{z_{il} h_{lj}^t}{\sum_m z_{im} h_{mj}^t} + \sum_k B_{jk}^- h_{lj}^t h_{lk}^t \right] = 0 \quad (64)$$

$$h_{lj}^{t+1} = \frac{\lambda + \sqrt{\lambda + 4 \sum_k B_{jk}^+ h_{lk}^t \left(\lambda \sum_i x_{ij} \frac{z_{il}}{\sum_m z_{im} h_{mj}^t} + \sum_k B_{jk}^- h_{lk}^t \right)}}{2 \sum_k B_{jk}^+ h_{lk}^t} h_{lj}^t \quad (65)$$

Update rule for basis matrix $\mathbf{Z}$:

$$\frac{\partial G(z, z^t)}{\partial z_{il}} = 0 \quad (66)$$
$$- \sum_j x_{ij} \frac{z_{il}^t h_{lj}}{\sum_m z_{im}^t h_{mj}} \frac{1}{z_{il}} + \sum_j h_{lj} = 0 \quad (67)$$
$$\sum_j h_{lj} z_{il} - \sum_j x_{ij} \frac{z_{il}^t h_{lj}}{\sum_m z_{im}^t h_{mj}} = 0 \quad (68)$$
$$z_{il}^{t+1} = \sum_j x_{ij} \frac{h_{lj}}{\sum_m z_{im}^t h_{mj}} \frac{1}{\sum_j h_{lj}} z_{il}^t \quad (69)$$

APPENDIX C
DERIVATION OF THE UPDATE RULES FOR DNMF/SVM

We shall use the EM algorithm. First, we prove that the cost function (39) is convex with respect to h_{lj}, meaning that $\frac{\partial^2 F(h_{lj})}{\partial h_{lj}^2} \geq 0$. We consider that h_{lj}^c belongs to class c. We need to calculate the second order partial derivatives $\frac{\partial^2 \mathbf{tr}[\mathbf{S}_W]}{\partial h_{lj}^2}$

and $\frac{\partial^2 \text{tr}[\mathbf{S}_B]}{\partial h_{lj}^2}$:

$$\begin{aligned}
\frac{\partial \text{tr}[\mathbf{S}_W]}{\partial h_{lj}^c} &= \frac{\partial}{\partial h_{lj}^c}\text{tr}\left[\sum_{r=1}^{C}\sum_{k=1}^{M_c}(\mathbf{h}_k^r-\bar{\mathbf{h}}^r)(\mathbf{h}_k^r-\bar{\mathbf{h}}^r)^T\right] \\
&= \frac{\partial \sum_{l=1}^{L}\sum_{r=1}^{C}\sum_{k=1}^{M_c}(h_{lk}^r-\bar{h}_l^r)^2}{\partial h_{lj}^r} \\
&= \sum_{l=1}^{L}\sum_{r=1}^{C}\sum_{k=1}^{M_c}\frac{\partial (h_{lk}^r-\bar{h}_l^r)^2}{\partial h_{lj}^r} \\
&= \sum_{k=1,\neq j}^{M_c}\frac{\partial (h_{lk}^c-\bar{h}_l^c)^2}{\partial h_{lj}^c}+\frac{\partial (h_{lj}^c-\bar{h}_l^c)^2}{\partial h_{lj}^c} \\
&= \sum_{k=1,\neq j}^{M_c}2(h_{lk}^c-\bar{h}_l^c)\left(-\frac{1}{M_c}\right)+ \\
&+ 2(h_{lj}^c-\bar{h}_l^c)\left(1-\frac{1}{M_c}\right) \\
&= 2(h_{lj}^c-\bar{h}_l^c)-\frac{1}{M_c}\sum_{k=1}^{M_c}2(h_{lk}^c-\bar{h}_l^c) \\
&= 2(h_{lj}^c-\bar{h}_l^c)-\frac{2}{M_c}\sum_{k=1}^{M_c}(h_{lk}^c)-\frac{2}{M_c}(M_c\cdot\bar{h}_l^c) \\
&= 2(h_{lj}^c-\bar{h}_l^c), \qquad (70)
\end{aligned}$$

$$\frac{\partial^2 \text{tr}[\mathbf{S}_W]}{\partial h_{lj}^{c2}} = \frac{\partial}{\partial h_{lj}^c}[2(h_{lj}^c-\bar{h}_l^c)] = 2\left(1-\frac{1}{M_c}\right)\geq 0 \qquad (71)$$

where we used the equality $\bar{h}_l^c = \frac{1}{M_c}\sum_{k=1}^{M_c} h_{lk}^c$, M_c the number of elements of class c and C the total number of classes,

$$\begin{aligned}
\frac{\partial \text{tr}[\mathbf{S}_B]}{\partial h_{lj}^c} &= \frac{\partial}{\partial h_{lj}^c}\text{tr}\left[\sum_{r=1}^{C}M_r(\bar{\mathbf{h}}^r-\bar{\mathbf{h}})(\bar{\mathbf{h}}^r-\bar{\mathbf{h}})^T\right] \\
&= \frac{\partial}{\partial h_{lj}^c}\sum_{l=1}^{L}\sum_{r=1}^{C}M_r(\bar{h}_l^r-\bar{h}_l)^2 \\
&= \sum_{r=1}^{C}M_r\frac{\partial(\bar{h}_l^r-\bar{h}_l)^2}{\partial h_{lj}^c} \\
&= \sum_{r=1,\neq c}^{C}M_r\frac{\partial(\bar{h}_l^r-\bar{h}_l)^2}{\partial h_{lj}^c}+M_c\frac{\partial(\bar{h}_l^c-\bar{h}_l)^2}{\partial h_{lj}^c} \\
&= \sum_{r=1,\neq c}^{C}M_r 2(\bar{h}_l^r-\bar{h}_l)\left(-\frac{1}{M}\right) \\
&+ M_c 2(\bar{h}_l^c-\bar{h}_l)\left(\frac{1}{M_c}-\frac{1}{M}\right) \\
&= 2(\bar{h}_l^c-\bar{h}_l)-\frac{2}{M}\sum_{r=1}^{C}M_r(\bar{h}_l^r-\bar{h}_l) \\
&= 2(\bar{h}_l^c-\bar{h}_l)-\frac{2}{M}\sum_{r=1}^{C}M_r\bar{h}_l^r+\frac{2}{M}M\bar{h}_l \\
&= 2(\bar{h}_l^c-\bar{h}_l) \qquad (72)
\end{aligned}$$

$$\frac{\partial^2 \text{tr}[\mathbf{S}_W]}{\partial h_{lj}^{c2}} = \frac{\partial}{\partial h_{lj}^c}[2(\bar{h}_l^c-\bar{h}_l)] = 2\left(\frac{1}{M_c}-\frac{1}{M}\right)\geq 0, \qquad (73)$$

where we used the inequality $\bar{h}_l = \frac{1}{M}\sum_{r=1}^{C}M_r\bar{h}_l^r$. The second order partial derivative of $F(h_{lj})$ is then given by:

$$\frac{\partial^2 F(h_{lj}}{\partial h_{lj}^{c2}} = \sum_i \frac{x_{ij}}{h_{lj}^{c2}}+2\gamma\left(1-\frac{1}{M_c}\right)-2\delta\left(\frac{1}{M_c}-\frac{1}{M}\right)+a_j^2y_j^2 \qquad (74)$$

which is ≥ 0 for $h_{lj}\geq 0$ and $M_c \geq 1+\frac{\delta}{\gamma}$. If we choose $\gamma\geq\delta$ then the second condition becomes $M_c\geq 2$, which means that in order for the convexity to hold each class must have at least two samples. This restriction is very loose and it is satisfied in all the conducted experiments.

We define the following auxiliary function (with respect to h_{lj}) for the cost function (39):

$$\begin{aligned}
G(h_{lj},h_{lj}^{(t)}) &= \lambda\left[\sum_{ij}(x_{ij}\ln x_{ij}-x_{ij})-\right. \\
&- \left.\sum_{ijl}x_{ij}\frac{z_{ij}h_{lj}^t}{\sum_m z_{im}h_{mj}^t}\left(\ln z_{il}h_{lj}-\ln\frac{z_{il}h_{lj}^t}{\sum_m z_{im}h_{mj}^t}\right)\right. \\
&+ \left.\sum_{ijl}z_{il}h_{lj}\right] \\
&+ \gamma\text{tr}[\mathbf{S}_W]-\delta\text{tr}[\mathbf{S}_B]+\frac{1}{2}\sum_{ljk}\frac{B_{jk}^+h_{lk}^t}{h_{lj}^t}h_{lj}^2- \\
&- \tfrac{1}{2}\textstyle\sum_{ljk}B_{jk}^-h_{lj}^th_{lk}^t\left(1+\ln\frac{h_{lj}h_{lk}}{h_{lj}^th_{lk}^t}\right)-\sum_j a_j. \qquad (75)
\end{aligned}$$

The update rule for h_{lj} is extracted by setting the partial derivative of (75) to zero:

$$\begin{aligned}
\frac{\partial G(h_{lj},h^{c(t)}_{lj})}{\partial h_{lj}^c} &= \lambda\sum_i x_{ij}\frac{z_{il}h_{lj}^{ct}}{\sum_m z_{im}h_{mj}^{ct}}\frac{1}{h_{jl}^c}+\lambda\sum_i z_{il}+ \\
&+ \gamma 2(h_{lj}^c-\bar{h}_l^c)-\delta 2(\bar{h}_l^c-\bar{h}_l)+\sum_k\frac{B_{jk}^+h_{lk}^{ct}}{h_{lj}^{ct}}h_{lj}^c \\
&- \sum_k B_{jk}^-h_{lj}^{ct}h_{lk}^{ct}\frac{1}{h_{lj}^c} \\
&= \left[2\gamma-2\gamma\tfrac{1}{M_c}-2\delta\tfrac{1}{M_c}+2\delta\tfrac{1}{M}+\textstyle\sum_k\frac{B_{jk}^+h_{lk}^{ct}}{h_{lj}^{ct}}\right]h_{lj}^c \\
&+ \left[\lambda\sum_i z_{il}-2\gamma\frac{1}{M_c}\sum_{k=1,\neq j}^{M_c}h_{lk}^c-\right. \\
&- \left.2\delta\frac{1}{M_c}\sum_{k=1,\neq j}^{M_c}h_{lk}^c+2\delta\frac{1}{M}\sum_{k=1,\neq j}^{M}h_{lk}^c\right] \\
&+ \left[\lambda\textstyle\sum_i x_{ij}\frac{z_{il}h_{lj}^{ct}}{\sum_m z_{im}h_{mj}^{ct}}+\sum_k B_{jk}^-h_{lj}^{ct}h_{lk}^{ct}\right]\frac{1}{h_{lj}^c}=0 \qquad (76)
\end{aligned}$$

$$\begin{aligned}
&\left[2\gamma\left(1-\frac{1}{M_c}\right)-2\delta\left(\frac{1}{M_c}-\frac{1}{M}\right)+\sum_k\frac{B_{jk}^+h_{lk}^{ct}}{h_{lj}^{ct}}\right]h_{lj}^{c2}+ \\
+ &\left[\lambda\sum_i z_{il}-2\gamma\frac{1}{M_c}\sum_{k=1,\neq j}^{M_c}h_{lk}^c-\right. \\
- &\left.2\delta\frac{1}{M_c}\sum_{k=1,\neq j}^{M_c}h_{lk}^c+2\delta\frac{1}{M}\sum_{k=1,\neq j}^{M}h_{lk}^c\right]h_{lj}^c+ \\
+ &\left[\lambda\sum_i x_{ij}\frac{z_{il}h_{lj}^{ct}}{\sum_m z_{im}h_{mj}^{ct}}+\sum_k B_{jk}^-h_{lj}^{ct}h_{lk}^{ct}\right]=0 \qquad (77)
\end{aligned}$$

By solving the second order polynomial equation (77) we obtain the update rule given in (40).

REFERENCES

[1] 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, Oct. 1999.

[2] P. Smaragdis and J. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, oct. 2003, pp. 177 – 180.

[3] K. Devarajan, "Nonnegative matrix factorization: An analytical and interpretive tool in computational biology," *PLoS Comput Biol*, vol. 4, no. 7, p. e1000029, july 2008.

[4] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, ser. SIGIR '03. ACM, 2003, pp. 267–273.

[5] D. Guillamet, J. Vitri, and B. Schiele, "Introducing a weighted non-negative matrix factorization for image classification," *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2447 – 2454, 2003.

[6] D. D. Lee and H. S. Seung, "Algorithms for Non-negative Matrix Factorization," in *Advances in Neural Information Processing Systems 13*. MIT Press, Apr. 2001, pp. 556–562.

[7] C.-J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1589–1596, 2007.

[8] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, pp. 121–167, June 1998.

[9] D. Geebelen, J. A. K. Suykens, and J. Vandewalle, "Reducing the number of support vectors of svm classifiers using the smoothed separable case approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 4, pp. 682–688, april 2012.

[10] S. Moon and H. Qi, "Hybrid dimensionality reduction method based on support vector machine and independent component analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 5, pp. 749–761, may 2012.

[11] D. Anguita, A. Ghio, L. Oneto, and S. Ridella, "In-sample and out-of-sample model selection and error estimation for support vector machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 9, pp. 1390–1406, september 2012.

[12] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 683 –695, may 2006.

[13] Y. Wang and Y. Jia, "Fisher non-negative matrix factorization for learning local features," in *Proc. Asian Conf. on Comp. Vision*, 2004, pp. 27–30.

[14] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Non-negative matrix factorization framework for face recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, pp. 495–511, 2005.

[15] I. Buciu, N. Nikolaidis, and I. Pitas, "Nonnegative matrix factorization in polynomial feature space," *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 1090 –1100, june 2008.

[16] C.-J. Lin, "Projected Gradient Methods for Nonnegative Matrix Factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, Oct. 2007.

[17] I. Kotsia, S. Zafeiriou, and I. Pitas, "A novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 588 –595, sept. 2007.

[18] Z. Liang, Y. Li, and T. Zhao, "Projected gradient method for kernel discriminant nonnegative matrix factorization and the applications," *Signal Processing*, vol. 90, no. 7, pp. 2150 – 2163, 2010.

[19] R. Zdunek and A. Cichocki, "Nonnegative matrix factorization with quadratic programming," *Neurocomputing*, vol. 71, no. 10-12, pp. 2309 – 2320, 2008.

[20] ——, "Nonnegative matrix factorization with constrained second-order optimization," *Signal Processing*, vol. 87, no. 8, pp. 1904 – 1916, 2007.

[21] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1087–1099, july 2012.

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

[23] S. Li, X. W. Hou, H. J. Zhang, and Q. S. Cheng, "Learning spatially localized, parts-based representation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 207 – 212.

[24] B. Gao, W. Woo, and S. Dlay, "Variational regularized 2-d nonnegative matrix factorization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 5, pp. 703 –716, may 2012.

[25] A. Pascual-Montano, J. Carazo, K. Kochi, D. Lehmann, and R. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsnmf)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 403 –415, march 2006.

[26] M. Heiler and C. Schnörr, "Learning sparse representations by non-negative matrix factorization and sequential cone programming," *The Journal of Machine Learning Research*, vol. 7, pp. 1385–1407, December 2006.

[27] A. Cichocki, R. Zdunek, and S.-i. Amari, "Csiszár's divergences for non-negative matrix factorization: Family of new algorithms," in *Independent Component Analysis and Blind Signal Separation*, ser. Lecture Notes in Computer Science, J. Rosca, D. Erdogmus, J. Prncipe, and S. Haykin, Eds. Springer Berlin / Heidelberg, 2006, vol. 3889, pp. 32–39.

[28] M. Das Gupta and J. Xiao, "Non-negative matrix factorization as a feature selection tool for maximum margin classifiers," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 2841 –2848.

[29] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.

[30] L. Saul, F. Pereira, and O. Pereira, "Aggregate and mixed-order markov models for statistical language processing," in *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 1997, pp. 81–89.

[31] T. Hofmann, B. Schlkopf, and A. J. Smola, "Kernel methods in machine learning," *Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, 2008.

[32] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization." *Nature*, vol. 401, no. 6755, pp. 788–91, 1999.

[33] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[34] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks." *Johns Hopkins APL Technical Digest*, vol. 10, pp. 262–266, 1989.

[35] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the Symposium on Computer Applications and Medical Care*. IEEE Computer Society Press, 1988, pp. 261–265.

[36] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast cancer diagnosis and prognosis via linear programming," *OPERATIONS RESEARCH*, vol. 43, pp. 570–577, 1995.

[37] P. Ekman, "Universals and cultural differences in facial expressions of emotion," *Nebraska Symposium on Motiation*, vol. 19, 1971.

[38] T. Kanade, J. Cohn, and Y. T., "Comprehensive database for facial expression analysis," in *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 46 –53.

[39] [Online]. Available: http://www.aiia.csd.auth.gr/MOBISERV-AIIA/index.html

[40] E. Marami, A. Tefas, and I. Pitas, "Nutrition assistance based on skin color segmentation and support vector machines," in *Man-Machine Interactions 2*, ser. Advances in Intelligent and Soft Computing. Springer Berlin / Heidelberg, 2011, vol. 103, pp. 179–187.

[41] J. Davis, "Hierarchical motion history images for recognizing human motion," in *IEEE Workshop on Detection and Recognition of Events in Video*, 2001, pp. 39 –46.

[42] S. Nikitidis, N. Nikolaidis, and I. Pitas, "Multiplicative update rules for incremental training of multiclass support vector machines," *Pattern Recognition*, vol. 45, no. 5, pp. 1838 – 1852, 2012.

[43] F. Sha, Y. Lin, L. K. Saul, and D. D. Lee, "Multiplicative updates for nonnegative quadratic programming," *Neural Comput.*, vol. 19, pp. 2004–2031, August 2007.