Max-Margin Semi-NMF

Vijay Kumar B.G
vijay.kumar@elec.qmul.ac.uk
Irene Kotsia
irene.kotsia@eecs.qmul.ac.uk
Ioannis Patras
i.patras@eecs.qmul.ac.uk

Multimedia and Vision Research Group
Queen Mary, University of London
London, UK

Abstract

In this paper, we propose a maximum-margin framework for classification using Non-negative Matrix Factorization. In contrast to previous approaches where the classification and matrix factorization stages are separated, we incorporate the maximum margin constraints within the NMF formulation, *i.e* we solve for a base matrix that maximizes the margin of the classifier in the low dimensional feature space. This results in a non-convex constrained optimization problem with respect to the bases, the projection coefficients and the separating hyperplane, which we propose to solve in an iterative way, solving at each iteration a set of convex sub-problems with respect to subsets of the unknown variables. The resulting basis matrix is used to extract features that maximize the margin of the resulting classifier. The performance of the proposed algorithm is evaluated on several publicly available datasets where it is shown to consistently outperform Discriminative NMF and SVM classifiers that use features extracted by semi-NMF.

1 Introduction

The Non-Negative Matrix Factorization (NMF) algorithm is one of the most popular Machine Learning techniques, being widely used for many computer vision applications. NMF aims at decomposing the data matrix into a product of a non-negative basis matrix with a non-negative coefficients matrix. NMF has been extensively used in many areas due to the fact that it achieves an approximation of the original data using a smaller number of dimensions. More precisely, it has been thoroughly used in Computer Vision applications such as pose estimation [1], action recognition [11, 14], object recognition [9], subspace learning [2] and clustering [5] and in facial analysis including detection [4], recognition [8], verification [17] and expression recognition [6, 15].

NMF was initially introduced in [7] and [10], where the minimization of a reconstruction error, defined as the discrepancy between the approximation obtained using the decomposition matrices and the input data was used as the criterion to acquire the basis and coefficients matrices. An extension of the NMF algorithm, the so called Semi-NMF, in which the non-negativity constraints were relaxed, was proposed in [5] and applied for clustering. The relaxation of the non-negativity constraints leads to the derivation of a basis matrix containing cluster centers and to non-negative coefficients that can be regarded as cluster indicators.

Regarding the discriminative power of the features extracted using NMF, only a few approaches have been proposed. More precisely, in [17] the authors introduced discriminative constraints in order to extract bases that correspond to discriminative facial regions for the problem of face recognition. The proposed Discriminant NMF (DNMF) [17] resulted in bases corresponding to salient facial features, such as eyes, mouth etc. In [6] projected gradients were used in DNMF (PGDNMF) for facial expression and face recognition.

In the proposed approach we choose the projections in such a way that the discriminative ability of an SVM classifier is maximized, therefore ensuring a higher classification performance. More specifically, we introduce soft max-margin constraints to the objective function of NMF to obtain a basis matrix that maximizes the classification margin using the features that are extracted using those bases. In the proposed scheme we optimize a weighted combination of the reconstruction error term that is used in typical NMF formulations and the cost that is used in typical SVM formulations, under SVM-type linear inequality constraints. The optimization is performed with respect to the unknown bases, the projection coefficients and the parameters of the separating hyperplane and is solved in an iterative manner, where at each iteration we solve only for one of them while keeping the others fixed. The resulting sub-optimization problems are either instances of Quadratic Programming with linear inequality constraints or classical SVM-type problems. The proposed method is applied at publicly available databases (the Mediamill dataset, the KTH action) where we demonstrate that it consistently outperforms SVM classification schemes that use features that are extracted using Semi-NMF [5] or DNMF [17].

Summarizing, the main contributions of the paper are

- We introduce a Max-margin framework for semi Non-Negative Matrix Factorization (MNMF).
- We propose an optimization scheme that solves for a max-margin classifier simultaneously with the decomposition matrices and bases.

The rest of the paper is organized as follows. In Section 2, we briefly describe the NMF and Semi-NMF schemes. In Section 3, we formulate the proposed Max-margin framework for semi-NMF and describe an algorithm to solve it. We demonstrate the performance of the proposed algorithm in Section 4 and draw conclusions in Section 5.

2 Semi Non-negative Matrix Factorization

In this section, we present a brief overview of the semi-NMF algorithm for matrix decomposition. Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ represent a non-negative matrix having n examples in its columns. The NMF algorithm [7] decomposes $\mathbf{X}$ into two non-negative matrices, the basis matrix $\mathbf{G} \in \mathbb{R}^{m \times k}$ and the coefficients matrix $\mathbf{H} \in \mathbb{R}^{k \times n}$ such that $\mathbf{X} \approx \mathbf{GH}$. k is typically chosen to be small ($< \min(m,n)$) in order to accomplish dimensionality reduction. The columns of $\mathbf{G}$ can be regarded as the basis vectors and thus each example can be represented as the linear combination of those basis vectors as $\mathbf{x}_i = \mathbf{G}\mathbf{h}_i$. Here $\mathbf{x}_i$ and $\mathbf{h}_i$ are the i^{th} columns of $\mathbf{X}$ and $\mathbf{H}$, respectively.

Ding *et al.* [5] introduced Semi-NMF that relaxes the non-negativity constraints on $\mathbf{G}$ and hence on the data matrix $\mathbf{X}$. Their motivation was based on the case of clustering with $\mathbf{G}$ representing the cluster centers and $\mathbf{H}$ denoting the cluster indicators. The matrices are

determined by minimizing the reconstruction error $\|\mathbf{X}-\mathbf{GH}\|_F^2$ or the Kullback-Leibler divergence $\mathrm{D}(\mathbf{X}||\mathbf{GH})$ w.r.t. $\mathbf{G}$ and $\mathbf{H}$

$$\underset{\mathbf{H}\geq 0}{\operatorname{argmin}}\|\mathbf{X}-\mathbf{GH}\|_F^2 \qquad \text{or} \qquad \underset{\mathbf{H}\geq 0}{\operatorname{argmin}}\,\mathrm{D}(\mathbf{X}||\mathbf{GH}) \tag{1}$$

where $\|.\|_F$ corresponds to the Frobenious norm. From now onwards we will use the notation $\mathbf{H}\geq 0$ to specify that the elements of the matrix $\mathbf{H}$ are non-negative. The above minimization problems are iteratively solved with respect to $\mathbf{G}$ and $\mathbf{H}$ using a set of update rules [5]:
Step 1: Update $\mathbf{G}$ by keeping $\mathbf{H}$ fixed

$$\mathbf{G}=\mathbf{X}\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1} \tag{2}$$

Step2: Update $\mathbf{H}$ by keeping $\mathbf{G}$ fixed at the value computed in the above step,

$$\mathbf{H}=\mathbf{H}\odot\sqrt{\frac{[\mathbf{G}^T\mathbf{X}]^+ + [\mathbf{G}^T\mathbf{G}]^-\mathbf{H}}{[\mathbf{G}^T\mathbf{G}]^+\mathbf{H} + [\mathbf{G}^T\mathbf{X}]^-}} \tag{3}$$

where $\mathbf{M}^+$ and $\mathbf{M}^-$ correspond to a positive and a negative part of the matrix $\mathbf{M}$, respectively, given by

$$\mathbf{M}^+{}_{ik}=\frac{1}{2}(|\mathbf{M}_{ik}|+\mathbf{M}_{ik}), \qquad \mathbf{M}^-{}_{ik}=\frac{1}{2}(|\mathbf{M}_{ik}|-\mathbf{M}_{ik}).$$

3 Max-Margin Semi-NMF

The NMF algorithm described in [7] minimizes either of the cost functions defined in Eq. 1 imposing at the same time non-negativity constraints on $\mathbf{G}$ and $\mathbf{H}$. Several variants of NMF with discriminant constraints were proposed in [5, 6, 17]. The variations were obtained by introducing application specific discriminant constraints to the cost function. Inspired by this, we aim at finding a set of basis vectors that maximizes the margin of an SVM classifier. We illustrate this with the example as shown in Fig. 1.

3.1 Cost Function

Let $\{\mathbf{x}_i, y_i\}_{i=1}^L$ denote a set of data vectors and their corresponding labels, where $\mathbf{x}_i\in\mathbb{R}^m$ and $y_i\in\{-1,1\}$. The objective is to determine a set of basis vectors 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 $\mathbf{G}$. Let us assume that the projection vector for a data example $\mathbf{x}_i$ is given by $\acute{\mathbf{x}}_i=\mathbf{G}^\dagger\mathbf{x}_i$ where $\mathbf{G}^\dagger$ corresponds to the pseudo-inverse of $\mathbf{G}$ and is defined as $\mathbf{G}^\dagger=(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T$. In practice, $\mathbf{G}^\dagger$ may suffer from numerical stability problems and is hard to work with since its calculation requires a matrix inversion. In order to overcome this, we use $\mathbf{G}^T\mathbf{x}$ as the features for the classifier [6, 17]. Then, the optimization problem for the proposed criterion is given by

$$\begin{aligned}\underset{\mathbf{G},\mathbf{H},\mathbf{w},b,\xi_i}{\operatorname{argmin}}\ & \lambda\|\mathbf{X}-\mathbf{GH}\|_F^2+\frac{1}{2}\mathbf{w}^T\mathbf{w}+C\sum_{i=1}^{L}\xi_i \\ \text{s.t. }\ & y_i(\mathbf{w}^T\mathbf{G}^T\mathbf{x}_i+b)\geqslant 1-\xi_i \\ & \xi_i>0,\ \ 1\leq i\leq L,\ \mathbf{H}\geq 0\end{aligned} \tag{4}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_L]$, λ and C are positive constants and λ is the weight factor for the NMF cost. The first term in the above optimization problem corresponds to the NMF reconstruction error while the remaining terms correspond to the maximum margin classifier. The above formulation aims at maximizing the margin of the support vectors while at the same time minimizing the reconstruction and misclassification error. We iteratively solve for one of the terms $\mathbf{G}$, $\mathbf{H}$ and $\mathbf{w}, b, \xi_i$ by keeping the remaining parameters fixed as described below. Once the training is complete, the classifier can be applied on the projected data points $\mathbf{G}^T\mathbf{x} \in \mathbb{R}^k$ where typically $k \ll m$.

The steps followed in the proposed max-margin Semi-NMF framework are summarized in Algorithm 1. $\mathbf{G}init$ and $\mathbf{H}init$ are random initializations.

Algorithm 1: Algorithm for MNMF

input : $\mathbf{X}$, $\mathbf{G}init$, $\mathbf{H}init$, $MAXITER$, λ, C
output: $\mathbf{G}$, $\mathbf{H}$, $\mathbf{w}$, b
begin
  $\mathbf{G} = \mathbf{G}init$;
  $\mathbf{H} = \mathbf{H}init$;
  repeat
    $S1$: Solve for α in Eq. 9
    $S2$: Compute $\mathbf{G}$ using Eq. 7
    $S3$: Find the classifier parameters, $\mathbf{w}$, b for the updated $\mathbf{G}$
    $S4$: **foreach** *column of* $\mathbf{H}$ **do**
      Calculate γ using Eq. 17
      Compute $\mathbf{h}$ using Eq. 16
    end
  until $iter \leq MAXITER$ ***or*** *convergence*;
end

Step S1, S2 : Solve for G by keeping H, w and b fixed: Since $\mathbf{w}$ is fixed, the optimization problem in Eq. 4 is simplified as

$$\begin{aligned} \underset{\mathbf{G},\xi_i}{\operatorname{argmin}} \quad & \lambda\|\mathbf{X}-\mathbf{G}\mathbf{H}\|_F^2 + C\sum_{i=1}^{L}\xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{G}^T\mathbf{x}_i + b) \geqslant 1-\xi_i \\ & \xi_i > 0, \;\; 1 \leq i \leq L \end{aligned} \tag{5}$$

The above formulation is a weighted combination of the reconstruction error (1st term) and soft constraints/penalizations for the examples that do not maintain the appropriate distance (margin) from the separating hyperplane (2nd term). Hence we want to find a set of bases $\mathbf{G}$ that simultaneously reduce the reconstruction error and the misclassification. Note, that we arrived at a cost function that is quadratic or linear with respect to the unknowns and at linear inequality constraints. We proceed to solve the above problem in its dual formulation.

The Lagrangian of Eq. 5 is given by

$$\begin{aligned} L(\mathbf{G},\xi_i,\alpha_i,\beta_i) \;=\; & \lambda Tr\Big((\mathbf{X}-\mathbf{GH})(\mathbf{X}-\mathbf{GH})^T\Big)+ \\ C\sum_{i=1}^{L}\xi_i - \sum_{i=1}^{L}\alpha_i\left[y_i(\mathbf{w}^T\mathbf{G}^T\mathbf{x}_i+b)-1+\xi_i\right] & - \sum_{i=1}^{L}\beta_i\xi_i \\ \alpha_i,\beta_i>0, \quad & 1\leq i\leq L \end{aligned} \tag{6}$$

where α_i, β_i are the Lagrangian multipliers. Taking the derivative w.r.t. to the primal variables and equating to 0, we have

$$\frac{\partial L}{\partial \mathbf{G}} = -2\mathbf{X}\mathbf{H}^T + 2\mathbf{G}\mathbf{H}\mathbf{H}^T - \sum_{i=1}^{L}\alpha_i y_i \mathbf{x}_i\mathbf{w}^T = 0$$

$$\Rightarrow \quad \mathbf{G} = \left(2\mathbf{X}\mathbf{H}^T + \sum_{i=1}^{L}\alpha_i y_i \mathbf{x}_i\mathbf{w}^T\right)(2\mathbf{H}\mathbf{H}^T)^{-1} \tag{7}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \quad 0\leq\alpha_i\leq\theta, \quad 1\leq i\leq L \tag{8}$$

where $\theta = C/\lambda$. Substituting the value of $\mathbf{G}$ in Eq. 6 we get the dual problem

$$\begin{aligned} \underset{\alpha}{\text{argmax}} \;\; & \alpha^T(\mathbf{T}_1-\mathbf{T}_2)\alpha + (\mathbf{t}_3-\mathbf{t}_4-\mathbf{t}_5-\mathbf{t}_6+\mathbf{t}_7)\alpha \\ & \text{s.t. } 0\leq\alpha_i\leq\theta \end{aligned} \tag{9}$$

where

$$\begin{aligned} & \alpha\in\mathbb{R}^L, \;\; \mathbf{T}_1,\mathbf{T}_2\in\mathbb{R}^{L\times L}, \; \mathbf{t}_3,\mathbf{t}_4,\mathbf{t}_5,\mathbf{t}_6,\mathbf{t}_7\in\mathbb{R}^{1\times L}, \\ & \mathbf{T}_1 = \left[\sum_{k=1}^{L} y_i y_j \mathbf{h}_k^T\mathbf{B}\mathbf{M}_i^T\mathbf{M}_j\mathbf{B}\mathbf{h}_k\right]_{ij} \\ & \mathbf{T}_2 = \left[y_1y_2\mathbf{w}^T\mathbf{B}\mathbf{M}_j^T\mathbf{x}_i\right]_{ij} \\ & \mathbf{t}_3 = \left[4\sum_{k=1}^{L} y_i\mathbf{h}_k^T\mathbf{B}\mathbf{H}\mathbf{X}^T\mathbf{M}_i\mathbf{B}\mathbf{h}_k\right]_{1i} \\ & \mathbf{t}_4 = \left[2\sum_{k=1}^{L} y_i\mathbf{h}_k^T\mathbf{B}\mathbf{w}\mathbf{x}_i^T\mathbf{x}_k\right]_{1i} \\ & \mathbf{t}_5 = \left[2y_i\mathbf{w}^T\mathbf{B}\mathbf{H}\mathbf{X}^T\mathbf{x}_i\right]_{1i}, \;\; \mathbf{t}_6 = b\left[y_i\right]_{1i} \\ & \mathbf{t}_7 = [111\cdots1]_{1\times L}, \;\; \mathbf{B} = (2\mathbf{H}\mathbf{H}^T)^{-1}, \;\; \mathbf{M}_i = \mathbf{x}_i\mathbf{w}^T, \end{aligned} \tag{10}$$

and $\mathbf{h}_k$ is the k^{th} column of the matrix $\mathbf{H}$.

The above problem is quadratic in α. Therefore the conventional quadratic programming tools [16] can be used to solve for α. The α_i obtained is then used to compute $\mathbf{G}$ using Eq. 7. The constant term $\theta = \dfrac{C}{\lambda}$ in Eq. 9 is used as a tuning parameter. Large values of λ (compared to C), result in low values of θ which in turn reduces α_i. In the extreme, α_i tends to zero and

causes the second term in Eq. 7 to vanish making the update rule of $\mathbf{G}$ to be the one used in semi-NMF, as given in Eq. 2. Hence for large values of λ, the update rule for $\mathbf{G}$ tends to approach the update rule of semi-NMF, something that is also evident in Eq. 5.

Step S3: Solve for $\mathbf{w}, b, \xi$ by keeping G and H fixed: In Eq. 7, we computed the updated basis $\mathbf{G}$ using quadratic programming. We now keep the basis $\mathbf{G}$ and weight matrix $\mathbf{H}$ fixed and determine a hyperplane that maximizes the margin of the classifier. The features $\mathbf{G}^T\mathbf{x}$ are obtained by projecting the data points $\mathbf{x}$ onto the updated basis matrix $\mathbf{G}$. Since $\mathbf{G}$ and $\mathbf{H}$ are fixed, the optimization problem in Eq. 4 is simplified to that of a classical SVM:

$$\begin{aligned} &\underset{\mathbf{w},b,\xi_i}{\operatorname{argmin}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{L}\xi_i \\ &\text{s.t. } y_i(\mathbf{w}^T\mathbf{G}^T\mathbf{x}_i + b) \geqslant 1 - \xi_i \\ &\xi_i > 0, \;\; 1 \leq i \leq L. \end{aligned} \tag{11}$$

The above optimization problem intends to maximize the margin of the classifier while reducing the misclassification error. The hyperplane parameters $\mathbf{w}$ and b are obtained using an off-the-shelf SVM classifier [3].

Step S4: Solve for H by keeping G, w, and b fixed: We now solve for the matrix $\mathbf{H}$ by keeping all the remaining variables fixed. Since only the reconstruction error term of the optimization problem (Eq. 4) depends on $\mathbf{H}$, the optimization problem is simplified as

$$\begin{aligned} &\underset{\mathbf{H}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{G}\mathbf{H}\|_F^2, \\ &\text{s.t. } \mathbf{H} \geq 0 \end{aligned} \tag{12}$$

In order to find an $\mathbf{H}$ that is consistent we solve for $\mathbf{H}$ using quadratic programming. The i^{th} column of $\mathbf{H}$, $\mathbf{h}_i$, contributes only to the i^{th} data point $\mathbf{x}_i$ and hence the columns of $\mathbf{H}$ can be solved independently of each other. The above optimization problem can be solved using the update equation Eq. 3. Here we adopt an alternative optimization method. In particular, the objective function in Eq. 12 can be rewritten as

$$\sum_{i=1}^{L}\|\mathbf{X}_i - \mathbf{G}\mathbf{h}_i\|_F^2 = \sum_{i=1}^{L}(\mathbf{X}_i - \mathbf{G}\mathbf{h}_i)^T(\mathbf{X}_i - \mathbf{G}\mathbf{h}_i) \tag{13}$$

where with $\mathbf{h}_i$ we denote the i^{th} column of the matrix $\mathbf{H}$. Eq. 13 means that we can solve for each $\mathbf{h}_i$ independently of the rest of $\mathbf{H}$. That is we solve for

$$\begin{aligned} &\underset{\mathbf{h}_i}{\operatorname{argmin}} \quad (\mathbf{x}_i - \mathbf{G}\mathbf{h}_i)^T(\mathbf{x}_i - \mathbf{G}\mathbf{h}_i), \\ &\text{s.t. } \mathbf{h}_i \geq 0 \end{aligned} \tag{14}$$

We solve the above constrained optimization problem in its dual form. The Lagrangian of the above cost function is

$$L(\mathbf{h}_i) = (\mathbf{x}_i - \mathbf{G}\mathbf{h}_i)^T(\mathbf{x}_i - \mathbf{G}\mathbf{h}_i) - \gamma^T\mathbf{h}_i, \quad \gamma > 0 \tag{15}$$

where $\gamma \in \mathbb{R}^k$ is a vector of positive Lagrangian multipliers. Differentiating the above equation w.r.t. $\mathbf{h}_i$ and equating to zero, we get

$$\mathbf{h}_i = (2\mathbf{G}^T\mathbf{G})^{-1}(2\mathbf{G}^T\mathbf{x}_i + \gamma). \tag{16}$$

The dual formulation for Eq. 15 is given by

$$\underset{\gamma>0}{\operatorname{argmax}} \frac{1}{2}\gamma^T \mathbf{B}\gamma + 2\gamma^T \mathbf{B}\mathbf{G}^T \mathbf{x}_i \tag{17}$$

$$\text{where } \mathbf{B} = (2\mathbf{G}^T\mathbf{G})^{-1}$$

The above problem is quadratic in γ. We use a Quadratic Programming solver to solve Eq. 17. The weight vector $\mathbf{h}_i$ is obtained by substituting the computed value of γ in Eq. 16. This procedure is repeated for all columns of $\mathbf{H}$.

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

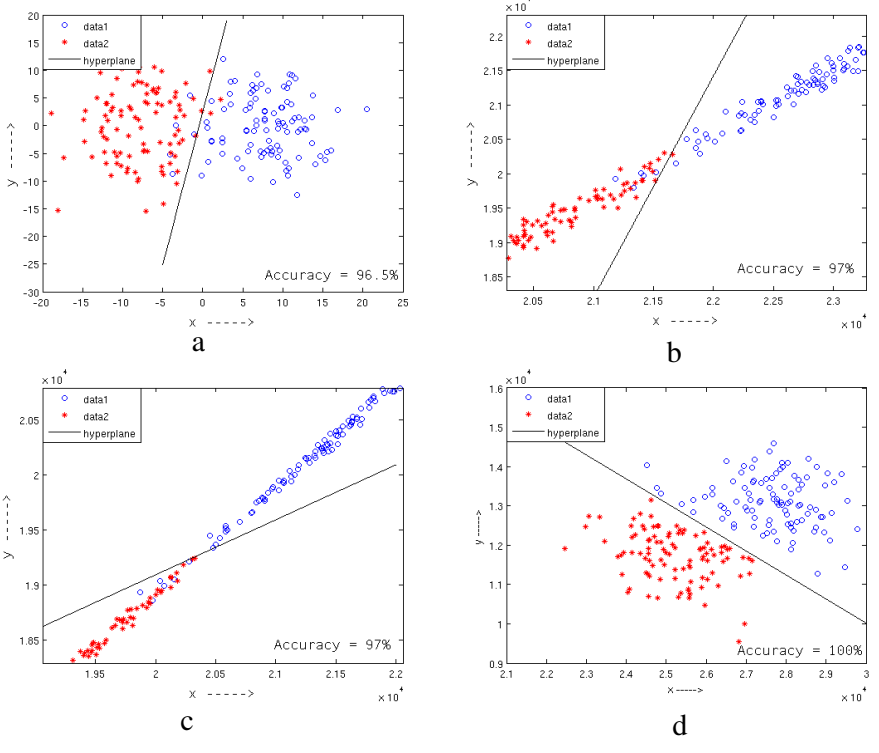

Figure 1: The projections and the SVM separating hyperplane using (*a*) PCA (*b*) Semi-NMF bases. (*c*) MNMF bases (1st iteration) and (*d*) MNMF bases (6^{th} iteration) respectively.

4 Experimental Results

In this section we demonstrate the performance of the proposed framework using both artificial and real, publicly available datasets. More specifically, apart from an artificial toy dataset we use a few object categories from the Mediamill dataset [13] and KTH actions dataset [12]. To allow comparisons with previously reported methods, we use the DNMF

algorithm [17] and also train an SVM classifier on the features that are extracted using Semi-NMF [5]. We show that the classification performance of the proposed scheme that jointly learns the classifier and performs matrix factorization is consistently higher, especially when when only few dimensions are retained.

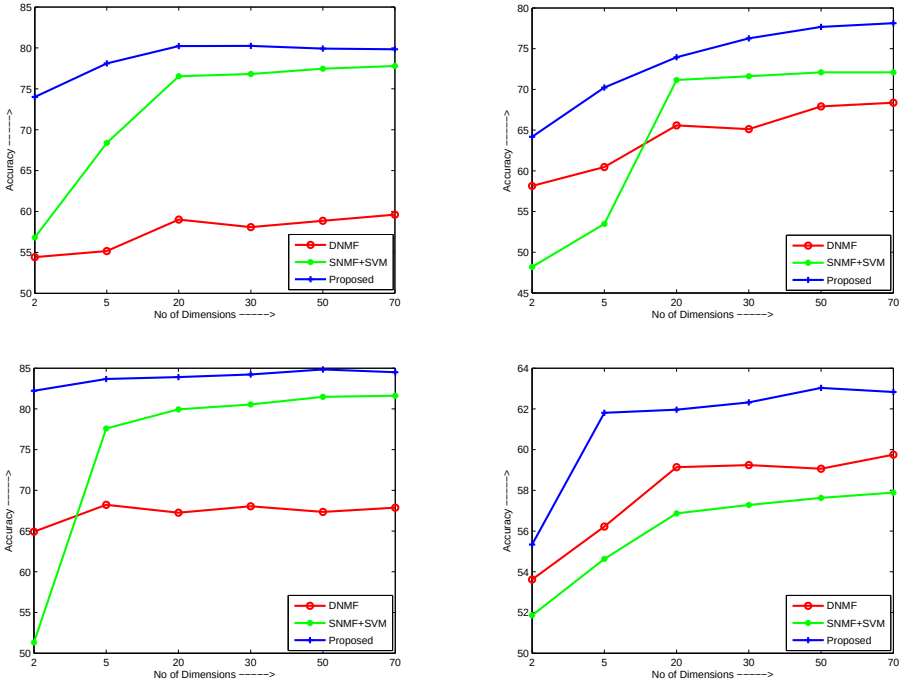

Figure 2: Comparison of the performance of the proposed algorithm with DNMF [17], Semi-NMF [5] + SVM on different categories of Mediamill dataset. The graph shows accuracies computed at different number of bases k.

In order to give an insight to the workings of the proposed algorithm, we first apply it on a toy dataset consisting of two classes each of which contains 100 points that are sampled from two 50-dimensional Gaussian distributions. In order to better visualize our results, we restrict the number of bases taken under consideration to be equal to two ($k = 2$). The basis matrix $\mathbf{G}$ and the weight matrix $\mathbf{H}$ are computed using the Semi-NMF algorithm and the input data points are projected onto the lower dimensional subspace using the acquired $\mathbf{G}$. In Fig. 1(a) we show the projections of the points after applying a common dimensionality reduction technique, Principal Component Analysis (PCA). Fig. 1(b) depicts the projections of the input data points using the bases extracted using Semi-NMF. Fig. 1(c) and Fig. 1(d) show the projections of the proposed MNMF algorithm after the first and the sixth iterations, respectively. When PCA, Semi-NMF (at convergence, i.e. after 2000 iterations) and the proposed MNMF algorithm (at convergence, i.e. after only 6 iterations) were applied the accuracies obtained were equal to 96.5%, 97% and 100%, respectively. It is clear that in the case of the proposed MNMF, the projections are such that the different classes become separable. This verifies the fact that the proposed algorithm updates the bases in such a way that the margin of the classifier in the projected space is maximized.

Table 1: Confusion matrix of the proposed MNMF on KTH dataset

%	box	clap	wave	jog	run	walk
box	90	2	2	2	2	2
clap	4	88	7	1	0	0
wave	5	3	92	0	0	0
jog	0	1	1	87	11	0
run	1	0	0	8	89	2
walk	1	0	0	1	3	95

Table 2: Comparison with other methods for KTH dataset

Method	SVM	NMF [5] + SVM	[11]	Proposed
Accuracy	87.3%	87.5%	86.9%	90.17%

We next tested our algorithm on the object categories from the Mediamill [13] dataset. The dataset consists of 43907 sub-shots with 101 classes. We randomly chose two object categories from the available 101 object categories and perform a binary classification task on the chosen object categories. Fig. 2 shows the results obtained for four different experiments. Each image is represented using a 120-dimensional feature vector. We chose two classes at a time and used the available feature vectors of those classes as the training samples to build the classifier model. For training, we randomly chose 200 images from each class and used the remaining images for testing. This procedure was repeated several times and we averaged the results.

We compared the performance of our algorithm with Semi-NMF [5] + SVM (*i.e* an SVM classifier trained on the projections acquired using Semi-NMF [5]), and DNMF [17] for the Mediamill dataset and reported the classification performance. The classification accuracy for different number of bases, that is for various values of k, is summarized in Fig. 2. For each k we repeat the experiments with different training sets sampled from the main dataset and report the average accuracy over all the runs. For simplicity, for each value of k, we used the value of C=100 that provided the best results for the Semi-NMF + SVM algorithm. It evident from the Fig. 2 that the proposed method outperforms all other methods in terms of the classification accuracy for all values of k. In particular, we notice that the proposed method significantly outperforms other methods when the number of basis vectors is small.

Subsequently, we test the proposed algorithm on the KTH action dataset consisting of 25 subjects in four scenarios performing actions under various scale and illumination scenarios. For the experiments we ignored the temporal information and considered only the spatial one. A period containing 4 naively chosen frames was extracted for every image sequence of every action. In order to handle possible illumination changes we used as features the Histogram of Oriented Gradients (HOGs). We restrict the number of bases k to 135 for this experiment. We employ an all-versus-all strategy for multi class classification, i.e. a binary classifier is built for each pair of classes. During testing, the class that receives maximum number of votes wins. The leave-one-person-out cross validation approach is used to test the performance of the algorithm.

The confusion matrix for the proposed algorithm on the KTH action dataset is shown

in Table 1. Table 2 summarizes the comparison of our algorithm with SVM, Semi-NMF [5] followed by SVMs and [11]. From Table 2, it is evident that the proposed algorithm outperforms the other methods in terms of classification accuracy.

5 Conclusions

In this paper we proposed a max-margin framework for Semi-NMF that introduces max-margin constraints within the Semi-NMF formulation. We simultaneously solve for the max-margin classifier and the decomposition matrices which leads to a basis matrix that maximizes the margin of the classifier in the feature space. We demonstrated the performance of the algorithm on publicly available datasets, where it was shown that the proposed algorithm consistently outperforms the Semi-NMF + SVM and DNMF algorithms in terms of classification accuracy.

6 Acknowledgement

This work was supported by the EPSRC grant 'Recognition and Localization of Human Actions in Image Sequences' (EP/G033935/1).

References

[1] A. Agarwal and B. Triggs. A local basis representation for estimating human pose from cluttered images. In *ACCV*, pages 50–59, 2006.

[2] D. Cai, X. He, X. Wu, and J. Hans. Non-negative matrix factorization on manifold. In *ICDM*, 2008.

[3] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[4] X. Chen, L. Gu, S. Z. Li, and H-J. Zhang. Learning representative local features for face detection. In *CVPR*, volume 1, pages 1126–1131, 2001.

[5] C. H. Q. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55, 2010.

[6] 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*, 2(3):588–595, 2007.

[7] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[8] S. Z. Li, Xin W. Hou, H. J. Zhang, and Q. S. Cheng. Learning spatially localized, parts-based representation. In *CVPR*, volume 1, pages 207–212, 2001.

[9] W. Liu and N. Zhen. Non-negative matrix factorization based methods for object recognition. In *Pattern Recognition Letters*, volume 25, pages 893–897, 2004.

[10] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[11] P. M. Roth, T. Mauthner, I. Khan, and H. Bischof. Efficient human action recognition by cascaded linear classification. In *IEEE Workshop on Video-Oriented Object and Event Classification*, 2009.

[12] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. *ICPR*, 2004.

[13] Cees G.M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W.M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the ACM Multimedia*, pages 421–430, 2006.

[14] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *CVPR*, pages 1–8, 2008.

[15] Y. Wang and Y. Jia. Fisher non-negative matrix factorization for learning local features. In *ACCV*, volume 1, pages 27–30, 2004.

[16] Adrian Wills. *QPC: Quadratic Programming in C*. Software available at http://sigpromu.org/quadprog/.

[17] 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*, 17(3):683–695, 2006.