Part-Based Data Analysis with Masked Non-negative Matrix Factorization

Gabriella Casalino[1], Nicoletta Del Buono[2], and Corrado Mencar[1]

[1] Department of Informatics
University of Bari, Bari 70125, Italy
[2] Department of Mathematics
University of Bari, Bari 70125, Italy
{gabriella.casalino,nicoletta.delbuono,corrado.mencar}@uniba.it

Abstract. We face the problem of interpreting parts of a dataset as small selections of features. Particularly, we propose a novel masked non-negative matrix factorization algorithm which is used either to explain data as a composition of interpretable parts (which are actually hidden in them) and to introduce knowledge in the factorization process. Numerical examples prove the effectiveness of the proposed algorithm as a useful tool for Intelligent Data Analysis.

Keywords: Nonnegative matrix factorization, mask matrix, structure retrieval.

1 Introduction

Non-negative Matrix Factorization (NMF) is a computational technique for low-rank approximation of a numerical dataset [13,14]. Differently to other low-rank approximation techniques, such as Principal Component Analysis (PCA), Singular Value Decomposition (SVD) *et similia* [8], NMF is able to explain data in terms of combination of nonnegative factors (provided that data are nonnegative too). In more formal terms, given a dataset represented as a nonnegative matrix

$$X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m] \in \mathbb{R}_+^{n \times m}$$

where $\mathbf{x}_i \in \mathbb{R}_+^n$ are column vectors representing samples [1], NMF algorithm aims to approximate X into the product of two non-negative matrices – a *base matrix* $W \in \mathbb{R}_+^{n \times k}$ and an *encoding matrix* $H \in \mathbb{R}_+^{k \times m}$ – such that

$$X \approx WH. \tag{1}$$

The value of k is user-specified and identifies the number of factors used to explain data. In fact, each sample is approximated as a nonnegative linear

[1] In the following, a matrix is denoted with an uppercase letter, e.g. X, its elements with the corresponding lowercase letter, e.g. x_{ij}, a column vector in lowercase boldface, e.g. $\mathbf{x}_i$

B. Murgante et al. (Eds.): ICCSA 2014, Part VI, LNCS 8584, pp. 440–454, 2014.

combination of factors, that is:

$$\hat{\mathbf{x}}_j = \sum_{i=1}^{k} \mathbf{w}_i h_{ij}. \tag{2}$$

The non-negativity characterization of NMF makes it a useful tool for Intelligent Data Analysis (IDA). In fact, the non-negativity makes NMF capable of representing data as a additive combination of common factors. Moreover, if such factors have some physical meaning (i.e., they can be interpreted in the domain of the considered problem), NMF makes possible to explain data as a composition of parts, being each part a factor. For examples: student questionnaire results can be explained in terms of basic student skills [9], news can be cathegorized according to the arguments they refer to [17], objects can be detected and localized in still in gray-scale images [6] and so on.

The main issue of NMF is therefore related to the ability of interpreting factors in the problem domain: unfortunately, decomposition (1) is not unique; also, it may be not easy to bring out useful knowledge from the representation of W and H [11].

In order to overcome the limitations of classical NMF and to introduce knowledge in the factorization process, additional constraints to the nonnegativity of the matrices W and H can be added. Some examples of constraints are: the sparseness adopted to increase the parts-based representations of the decomposition [12], several forms of orthogonality used to improve the cluster ability of NMF [4,10,16], local manifold structure [3], label information to perform a semi-supervised learning decomposition process [15], binary structure to produce biclustering structures explicitly [18]. Obviously, any additional constraint leads to the development of different optimization algorithms for NMF factorization.

In this paper, we face the problem of interpreting parts as small selections of features. More precisely, we constrain the column vectors of W so that only a small subset of elements is non-zero. This representation of parts could be very useful for IDA, since it is able to highlight some local linear relationships existing among features that hold for a subset of data. To this purpose we introduce a new optimization problem for NMF, which constrains the columns of the base matrix W to possess a small number of non-zero elements. Then, we adopt a query-based approach, where the structure of the base matrix is defined by a user-provided mask matrix. In this way, the analyst can specify the parts she is interested to discover in data; the proposed technique, in fact, extracts the subset of data that are actually represented by the parts.

The proposed approach has been tested on a number of synthetic datasets in order to show its effectiveness in correctly selecting parts in data according to the provided queries. Moreover, a preliminary experiment on the well known Iris data is also reported in order to demonstrate the validity of the proposed approach on real dataset.

The paper is organized as follows: in the next section, the query-based approach is skeched together with the masked NMF algorithm. This latter is derived by minimizing a novel weighted penalized objective function which has

been proposed to matching the query matrix and to preserve nonnegativity of data. In section 3 some numerical simulations are reported in order to illustrate the effectiveness of the proposed approach. In section 4 some conclusive remarks are outlined, along with future extensions of the proposed approach.

2 Masked NMF

Classical NMF algorithms used to compute the approximating factors W and H as in (1) are typically derived by solving the constrained least square minimization problem:

$$\min_{W\geq 0, H\geq 0} \frac{1}{2} \|X - WH\|_F^2 \tag{3}$$

where $\|\cdot\|_F$ indicates the Frobienus matrix norm on matrices.

The non-negativity constraints imposed by (3) often are not enough to provide for factors (i.e. the columns $\mathbf{w}_k$ of W) that represent useful knowledge. In fact, usually these columns are very dense; moreover, different configurations of W and H lead to the same approximation of X, thus it is difficult to associate a physical meaning to the factors. Indeed

$$X = (WC)\left(C^{-1}H\right)$$

for every $C \in \mathbb{R}^{k\times k}$ such that both C and C^{-1} are non-negative.

In order to overcome the limits of classical NMF and to inject a-priori knowledge in the factorization process we introduce the concept of part. From the vector representation of data derives that each sample is represented by a vector of n features $\{f_1, ..., f_n\}$. A part p is defined as a sparse vector in $\mathbb{R}^n$ where at least two components are non-zero. A feature belongs to a part iff its value is non-zero. In this way we constrain the factorization process to describe data as a linear correlation of different parts, whose features are linear correlated among them. The structure of the part (i.e. the features set to zero, thus excluded by the part), as well as the number of parts, constitutes the a-priori knowledge and is user-defined.

In order to obtain basis factors that are able to extract parts, we constrain the columns $\mathbf{w}_k$ in W to contain only few non-zero elements. We observe that factors possessing this type of structure enable the elicitation of local linear relationships in subsets of data.

In order to incorporate the previously explained additional constraints, we design the following minimization problem:

$$\min_{W\geq 0, H\geq 0} \frac{1}{2} \|X - (P \odot W)H\|_F^2 + \frac{1}{2}\lambda \left\|P \odot \tilde{W}\right\|_F^2 \tag{4}$$

where

$$\tilde{W}_{ij} = \exp\left(-W_{ij}\right)$$

and

$$P \in \{0,1\}^{m\times k}$$

is binary matrix, with a fixed number of non-zero elements per column which is used as query for the NMF problem, while $\lambda \geq 0$ is a regularization parameter.

The objective function in (4) is composed by two terms: the first one represents a weigthed modification of the classical NMF problem where the mask matrix P is used to fix the structure the base matrix W has to possess. The second term is a penalty term used to enhance the elements in W that match the mask structure. For this purpose the exponential function has been chosen: in fact when the value of an entry w_{ij} of W is low it is increased by the penalty term, when it is high the penalty tends to zero. The choice of the exponential function allows us to prevent that zero values correspond to features that we want to include in the parts. The query matrix P is used to identify the parts that the analyst would like to extract from data. This is accomplished by defining P as a set of k column vectors, where each element in a column is 1 if the corresponding feature has to be selected, 0 if it has not be considered.

The objective function (4) automatically imposes the structure of the query P in the factor matrix W, minimizing the non-relevant elements in W and maximizing (when they are actually present) the relevant elements in it. It should be observed, however, that the objective function (4) is not convex in both variables W and H. So, it is thus unrealistic to find the global minima for it. However, an iterative updating algorithm to obtain the local optima of (4) can be derived. Particularly, denoted by $\Psi = [\psi_{ij}]$ and $\Phi = [\phi_{ij}]$ the Lagrangian multipliers for the constraints $W_{ij} \geq 0$ and $H_{ij} \geq 0$, the Lagrangian function associated to the minimization problem in (4) is given by:

$$\begin{aligned}\mathcal{L} = \frac{1}{2} trace\left(\left(X - \left(P \otimes W\right) H\right)^{\mathrm{T}} \left(X - \left(P \otimes W\right) H\right)\right) \\ + \lambda trace\left(\left(P \otimes \tilde{W}\right)^{\mathrm{T}} \left(P \otimes \tilde{W}\right)\right) + trace\left(\Psi W\right) + trace\left(\Phi H\right)\end{aligned} \tag{5}$$

Imposing the Karush-Kuhn-Tucker conditions for the optimality, it follows that the derivative of the Lagragian with respect to W and H are

$$\frac{\partial \mathcal{L}}{\partial W} = \left(P \otimes W\right) H H^{\mathrm{T}} - P \otimes \left(X H^{\mathrm{T}}\right) + \frac{1}{2}\lambda \left(-2\right) P \otimes \tilde{W} + \Psi = 0 \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial H} = \left(P \otimes W\right)^{\mathrm{T}} \left(P \otimes W\right) H - \left(P \otimes W\right) X + \Phi = 0 \tag{7}$$

Solving the previous equations with respect to the elemements in W and H, the following updating formulas for W_{ij} and H_{ij} can be derived:

$$W_{ij} \leftarrow W_{ij} \frac{\left[P \odot \left(X H^{\mathrm{T}}\right)\right]_{ij} + \lambda \left(P \odot \tilde{W}\right)_{ij}}{\left[\left(P \odot W\right)\left(H H^{\mathrm{T}}\right)\right]_{ij} + \epsilon} \tag{8}$$

$$H_{ij} \leftarrow H_{ij} \frac{\left[\left(P \odot W\right)^{\mathrm{T}} X\right]_{ij}}{\left[\left(P \odot W\right)^{\mathrm{T}} \left(P \odot W\right) H\right]_{ij} + \epsilon} \tag{9}$$

where the constant $\epsilon = 10^{-12}$ has been introducted to prevent division by zero. We will refer to (8) and (9) as Masked NMF (MNMF).

Regarding these two updating rules, it is not difficult to prove that the objective function (4) is nonincreasing under the updating rules (8) and (9).

2.1 Query Based MNMF

Algorithm 1 formally describes the proposed approach to analyse data through MNMF. Particularly, the steps the proposed approach is composed by are justified and described in the following.

MNMF is an iterative updating algorithm based on the multiplicative algorithm proposed by Lee and Seung [14]. It alternatively updates the matrices W and H according to the rules (8) and (9) (lines 3 and 4 of algorithm 1) while stopping criteria is not satisfied (line 2). Since the algorithm converges to zero, the adopted stopping criteria is based on the difference between two following values of the objective function: the computation of updates stops when the difference is lower than a prescribed small value ε.

Formally, set $E = Obj(t) - Obj(t-1)$, where $Obj(t)$ indicates the value of the objective function at the $t-th$ iterate, then

$$stop = \begin{cases} true & \text{if } E \leq \varepsilon \\ false & \text{otherwise} \end{cases}$$

In the reported experiments $\varepsilon = 10^{-6}$. A maximum number of iterations (set equal to 1000) is also used as an additional stopping criteria just to avoid a high computational effort when very low convergence rate occurs.

Being the MNMF algorithm based on the gradient descent method, it is sensitive to the starting point. As stardart choice, the matrices W and H have been initialized using two random matrices W_0 and H_0 (line 1), however different initializations can lead to better results [2,5,7], further experiments will be aimed to examine this aspect.

It should be pointed out that data in the matrix X has been normalized (line 1) to lay in the unit sphere (i.e. $\|X_{:i}\|_2 = 1$ for $i = 1, \ldots, m$). This representation has been preferred because NMF works in vectorial space, where data are vectors with an own direction and not points. Normalization eliminates information related to the lenght of the vectors, preserving relationships in data.

After MNMF runs, columns of W are normalized in L_2 (line 6) together with the matrix H (line 7) in order to preserve the factorization results. This is accomplished multiplying the factor W for the diagonal matrix N, and multiplying H for its inverse:

$$\bar{W} = WN \tag{10}$$

$$\bar{H} = N^{-1}H \tag{11}$$

where $N = diag\left(norm(W_{:,1}), norm(W_{:,2}), \ldots, norm(W_{:,k})\right)$.

Algorithm 1. QMNMF

Require: $X \in \mathbb{R}_+^{n \times m}$ {dataset}
Require: $P \in \{0,1\}^{n \times k}$ {mask}
Require: λ {regularization parameter}
Require: $W_0 \in \mathbb{R}_+^{n \times k}$ and $H_0 \in \mathbb{R}_+^{k \times m}$ {initial matrices W and H}
Require: $t > 0$ {threshold}
Require: hardmode $\in \{true, false\}$ {hard-mode selection criterion}
1. Normalize X
2. **while** stopping criterion not satisfied **do**
3. update matrix W according to (8)
4. update matrix H according to (9)
5. **end while**
6. Normalize matrix W according to (10)
7. Adjust matrix H according to (11)
8. Binarize H into $\bar{H}$ according to eq. (12)
9. **if** hardmode **then**
10. Compute the column index set $J = j : \bar{h}_{I,j} = \mathbf{1}$
11. **else**
12. Compute the column index set $J = j : \bar{h}_{I,j} \neq \mathbf{0}$
13. **end if**
14. Select data samples $X' = X[1:n, J]$ {all rows and columns in J}
15. **return** $W \in \mathbb{R}_+^{n \times k}$ {selected parts}
16. **return** $H \in \mathbb{R}_+^{k \times m}$ {coefficents}
17. **return** $X' \in \mathbb{R}_+^{n \times m'}$ {data subset}

When a NMF of a given data matrix X is computed, each sample is approximated in a low-rank subspace (of k dimensionality) by equation (3) . Particularly, the elements of each columns of the encoding matrix H codify the information needed to identify the factors (columns of W) used to reconstruct each sample of X in the low-rank subspace. From a geometrical point of view, the columns of W define the basis vectors of a subspace of dimension k, and each column of H defines the coefficients for each basis vector that is needed to approximate the corresponding data sample in X. Therefore, the elements in a column of H identify the importance of each basis vector in approximating the data sample: if a coefficient is very small, then the corresponding basis vector is useless in approximating the sample; as a consequence, the data sample does not contain the part represented by this basis vector. Information stored in the matrix H can be used therefore for Intelligent Data Analysis. After MNMF optimization a possible occurrence is finding samples of H which have low values corresponding to a part. This means that MNMF was not able to find the corresponding part in that subset, so the analysis could be restricted to the subset of data where parts have been recognized. This can be accomplished by "binarizing" each column of H into $\bar{H}$, i.e.

$$\bar{h}_{ij} = \begin{cases} 1, & h_{ij} \geq t \\ 0, & h_{ij} < t \end{cases} \tag{12}$$

for a user-defined threshold $t > 0$ (line 8). Samples in the matrix H that have not been reconstructed using parts which we are looking for, are then removed from the matrix X. The remaining columns after this removal procedure form a new data matrix that is denoted by X' (line 14). This approach allows the selection of the samples in data that are actually represented by the specified parts. However, the new dataset X' can be extracted according to two alternative criteria: (i) a data sample is selected if it contains at least one part in W (soft-mode) (line 12); (ii) a data sample is selected if it contains all the parts represented in W (hard-mode) (line 10). The first criterion is more conservative as it selects data samples that are represented by possibily other parts that are not included in W. On the other hand, the second criterion is more selective because it selects a data sample only if it can be well reconstructed by all the parts in W. At the end of the selection process, MNMF is re-run for the subset of the selected data samples. The objective of this last step is to re-compute the values in the base and encoding matrices without taking into account data samples that are not composed by the selected parts. This provides a more precise estimation of the parts and their contribution in the data samples.

For technical reasons related to the optimization algorithm, columns of the query matrix P have to be mutually orthogonal, i.e. $p_i^{\mathrm{T}} p_j = 0$ with $i, j = 1 \dots k; i \neq j$. This constrain ensures independence of user queries in the columns of P. Moreover, the method requires extra bases. In order to avoid stretching in the factorization process we add many canonical bases to the mask as the feature that have not been selected in the query are. This ensures the factorization process to find parts that the analyst is actually interested in discovering in data. Without this precaution MNMF would attempt to force the reconstruction of data using only the specified bases. Hence if parts in the query are not enough to describe data, MNMF uses the extra bases. For this reason in the algorithm we analyze the subset of rows of the matrix H that don't refer to extra bases (named I).

3 Numerical Simulations

We illustrate the results of some numerical simulations performed on a synthetic dataset $X \in \mathbb{R}^{6\times 350}$ that is generated in a specific way to evaluate the ability of the proposed approach in finding parts in data (fig. 1).

To generate the data samples, we made use of two random variables, $s_1 \sim \mathcal{N}(5, 1)$ and $s_2 = \alpha s_1$ with $\alpha = 3$ (we cropped to 0 negative values). Then we generated three combinations of two out of six features (mutually orthogonal), namely $(1, 2)$, $(3, 4)$, $(5, 6)$. We considered each combination of features (i_1, i_2) and defined a correponding random basis $\mathbf{c}_h$, $h = 1, 2, 3$, so that $c_{i_1} = s_1$, $c_{i_2} = s_2$ and $c_i = 0$ for $i \notin \{i_1, i_2\}$. We finally generated the dataset in blocks of 50 samples, each block being defined as a combination of the random bases $\mathbf{c}_h$ (i.e. $\mathbf{c}_1$,$\mathbf{c}_2$,$\mathbf{c}_3$, $\mathbf{c}_1 + \mathbf{c}_2$, $\mathbf{c}_1 + \mathbf{c}_3$,$\mathbf{c}_2 + \mathbf{c}_3$,$\mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3$).

Figure 1 shows a graphical representation of the data matrix X, which has been constructed as linear combination of the three bases c_i. It should be ob-

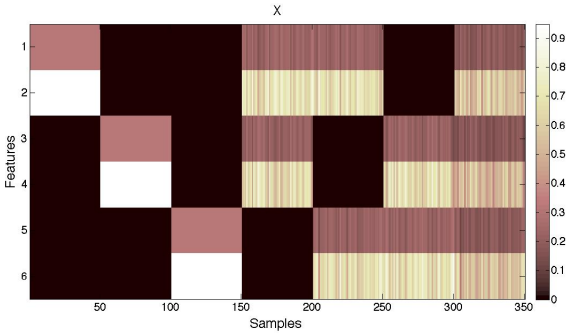

Fig. 1. Graphical illustration of the synthetic dataset X

served that the boxes represent fifty sequential data generated with the same linear combination.

Figure 2 reports the two different masks P_1 and P_2 adopted to query X. The mask P_1 is used to impose on the factors matrix W the same structure occurring in the dataset X, while the mask P_2 represents only partially the structure hidden in data (i.e., the first column in P_2 represents parts that are actually present in X, while the second represents parts that are not present in X). It has been pointed out that in both masks parts we are looking for in data are expressed by the first two columns, the remaining two have been added for techical reasons, and we will refer to them as extra bases. The mask P_1 allows us to verify if the proposed Query-based MNMF is able to recognize as relevant all the examples in dataset that have been constructed using the parts specified by the query mask; while the use of P_2 should show the behaviour of the algorithm when the analyst is looking for parts that are not actually in data.

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad P_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Fig. 2. Mask matrices used in the MNMF algorithm

This means that in both cases the query we submitted to the algorithm does not cover all the examples in the dataset, so we expect that the procedure selects a subset of it containing only the relevant data.

Figure 3 illustrates the mask matrix P_1 together with the factor matrix W computed by MNMF with parameter $\lambda = 0.00001$ and P_1. As it can be observed, the factor W possesses the same of structure of P_1 and the extracted bases $\mathbf{w}_i$ preserve also the multiplicative factor α that has been used to generate the data, being the ratio of the two non-zero features approximately equal to 3.

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad W = \begin{pmatrix} 0.316 & 0 & 0 & 0 \\ 0.949 & 0 & 0 & 0 \\ 0 & 0.316 & 0 & 0 \\ 0 & 0.949 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Fig. 3. Comparison between $W$and P_1 matrices obtained with $\lambda = 0.00001$

Figure 5 illustrates the matrix H obtained with MNMF and query P_2. As it can be observed only a subset of the samples has been reconstructed using only the parts we are looking for (the first two basis of the matrix W). It is made by samples from 1 to 100 and from 151 to 200. The algorithm in the hard mode returns this subset of data. The algorithm in the soft mode returns the samples that have been reconstructed using at least one of the parts in P_1. All the samples in dataset satisfy this requirement except the subset from 100 to 150.

Below a detailed analysis shows the behavior of MNMF in reconstructing samples when they contain the parts in P_1, linear relationship of these parts, and when there are not parts in the mask adequate to describe them.

Samples from 1 to 50 composed by the the first two features in X (whose linear correlation is captured by $\mathbf{w}_1$) have been correctly reconstructed in H using only the first part. Similarly, samples from 51 to 100 have been reconstructed using the part $\mathbf{w}_2$ representing the relationship between features three and four.

MNMF can recognize parts that are composed linearly to describe data. This is the case of the samples from 151 to 200 that have been generated adding data composed by the first two features and data composed by the third and fourth. These samples have been reconstructed in the matrix H using both the bases $\mathbf{w}_1$ and $\mathbf{w}_2$ capturing the linear correlation between respectively first and second features, and third and fourth features.

When the algorithm does not find parts that are able to correctly reconstruct the samples in data it uses the extra bases in the mask. This beaviour suggests to the analyst that the parts he is looking for in data are not enough to describe them. This is the case of samples from 201 to 350 that have been constructed adding data composed by feature caught in the mask and data composed by parts that are not in P_1. Matrix H correctly suggests which part in P_1 we need to reconstruct the samples and extra bases.

An extreme case are the samples from 101 to 150 that have been completely constructed using a part that is not in P_1, the algorithm returns no parts for this samples. Hence, our proposed Query Based MNMF algorithm suggests which parts correctly reconstruct data.

Figure 4 shows the reconstruction error (MSE) obtained for each sample in data. It has been obtained using the equation (13) where $i = i \ldots 350$ indicates the $i-$th sample and n is the total number of samples in the dataset.

$$MSE\,(i) = \frac{1}{2} \frac{\|X_{*i} - WH_{*i}\|_F^2}{n} \tag{13}$$

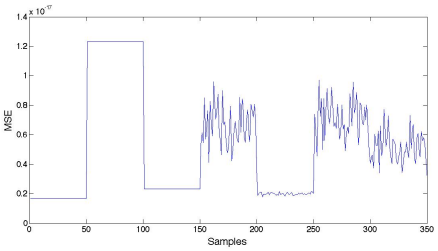

Fig. 4. MSE of each sample obtained with MNMF and P_1

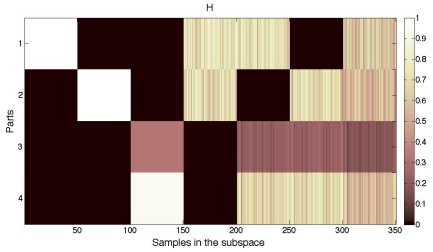

Fig. 5. Matrix H obtained with MNMF and P_1

Mask P_2 shows the behaviour of the Query Based MNMF algorithm when it is looking for parts that not correctly decribe data.

Figure 3 illustrates the basis matrix W computed by MNMF with matrix mask P_2 and $\lambda = 0.00001$. As it can be observed, whilst the first base $\mathbf{w}_1$ that catches the structure of the data has significative values, the second base $\mathbf{w}_2$ tries to describe data with parts that not completely explain them. For this reason one of the two values is very close to the maximum value 1, and the second one is low. This could suggest that the structure imposed does not allow a good reconstruction of the data.

Running the algorithm in the hard mode, it returns only the block of samples from 1 to 50. This result is correct, in fact only this subset is completely explained using the parts in P_2, particularly only the part w_1, instead the part w_2 doesn't match with the structure of the data. Soft mode algorithm returns all the samples except of the block from 51 to 100, it means that there is not in P_2 a part that describes these samples. In fact they have been constructed using the third and forth features, since the value w_{32} is negligible, there is any part in P_2 that allows to reconstruct these samples. Moreover samples in X that have been constructed using the fifth and the sixth features, are partially described by the part w_2, for this reason samples in the block from 100 to 150 have been reconstructed using both the part w_2 and an extra bases. This behaviour confirms that the part w_2 does not completely represent data, it needs extra information.

The graph of the MSE illustrated in figure 8 provides a further confirmation of the previously discussed behavior. Comparing the evolution of the MSE for

both cases, we can note that in the first example (where bases well explain data) the obtained error is near to the machine precision (i.e., 10^{17}) while for the latter example, MSE grows up to 10^{-4}.

$$P_2 = \begin{pmatrix} 1&0&0&0\\ 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1\\ 0&1&0&0 \end{pmatrix} \quad W = \begin{pmatrix} 0.316 & 0 & 0 & 0\\ 0.949 & 0 & 0 & 0\\ 0 & 0.123 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0.993 & 0 & 0 \end{pmatrix}$$

Fig. 6. Comparison between $W$and P_2 matrices obtained with $\lambda = 0.00001$

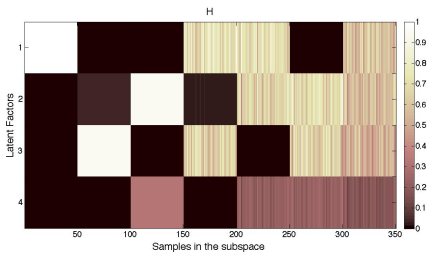

Fig. 7. Matrix H obtained with MNMF and P_2

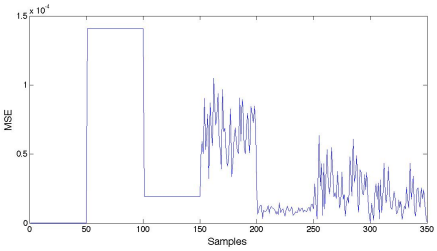

Fig. 8. MSE of each sample obtained with MNMF and P_2

A further synthetic dataset has been constructed to better explain the behaviour of the proposed method when it is forced to find parts that are not actually present in data. Points in $X \epsilon \mathbb{R}^{3x500}$ compose a circumference lying in the plane x,y on the bisector of the first quadrant, thence a linear relationship between features one and two exists in data (i.e. $x = y$). The components on the axes z are equal, or almost equal to 0. MNMF has been executed with two masks $P_3 = \begin{pmatrix} 1&0\\ 0&1\\ 1&0 \end{pmatrix}$ and $P_4 = \begin{pmatrix} 1&0\\ 1&0\\ 0&1 \end{pmatrix}$ and $\lambda = 0.0001$. The first mask P_3 tries to

find a linear relationship between the features on the axes x and z. As it can be observed from the figure 9, MNMF returns a matrix W_3 whose component w_{31} has a negligible value, while the component w_{11} has the maximum value 1. This behaviour confirms that there is no part in data corresponding to the axes z. Figure 10 shows the matrices H obtained when the masks P_3 (picture on the left) and P_4 (picture on the right) are used . As it can be expected data have been reconstructed using both bases. In the second example, the mask P_4 tries to find a relationship between the components on the axes x and y. MNMF returns a base matrix W_4 (figure 9) whose components w_{11} and w_{21} are significant, moreover it catches the linear relationship between these two components.The component w_{32}, even though data do not have a z-components, has a value equals to 1. It should be noted that this behavior has been produced by the normalization process. Despite this, the matrix H (as it can be observed by the right picture in Figure 10) confirms that the base w_2 is not necessary in reconstructing data, in fact all the samples use only the first base.

$$W_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0.0004 & 0 \end{pmatrix} W_4 = \begin{pmatrix} 0.7 & 0 \\ 0.7 & 0 \\ 0 & 1 \end{pmatrix}$$

Fig. 9. Masks matrices used in the MNMF algorithm

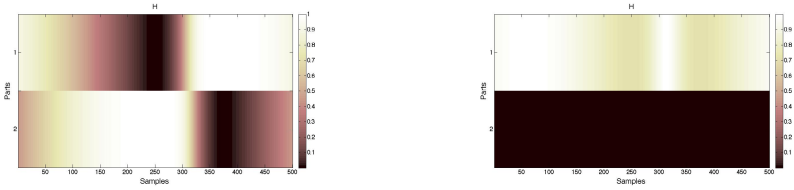

Fig. 10. Comparison between matrices H obtained with MNMF and masks P_3 and P_4

3.1 Iris Dataset

In this section we briefly illustrate the behaviour of the proposed approach when the well known Iris dataset is adopted [1]. The dataset is composed by 150 samples grouped in three different classes: Iris-Setosa, Iris-Veriscolor, Iris-Virginica. This example highlights the use of a specific mask to select features and extract samples which are described by these parts. Particularly, the aim is to discover if there exist any linear correlation between the features in the data samples (i.e., sepal and petal length, sepal width and petal length, sepal and petal width). MNMF has been executed with two different masks $P_5 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$

and $P_6 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$ and with parameter $\lambda = 0.0001$. Mask P_5 aims to discovery linear correlations in data between the lengths and the widths, whilst P_6 between sepal and petal feature. The use of a real dataset better highlights the semantic associated to the parts that have been selected.

Figure 11 illustrates the factor matrices W_5 and W_6 obtained respectively with masks P_5 and P_6. As it can be observed, the factor matrices preserve the structure imposed by the query masks, moreover the parts are represented by significative values; this means that they are actually present in data, i.e., there is correlation between the features selected in data.

$$W_5 = \begin{pmatrix} 0.85 & 0 \\ 0 & 0.96 \\ 0.53 & 0 \\ 0 & 0.29 \end{pmatrix} \quad W_6 = \begin{pmatrix} 0.88 & 0 \\ 0.48 & 0 \\ 0 & 0.95 \\ 0 & 0.31 \end{pmatrix}$$

Fig. 11. Comparison between W_5 and W_6 matrices obtained with masks P_5 and P_6 $\lambda = 0.0001$

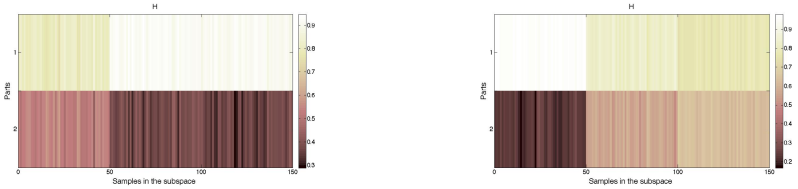

Fig. 12. Comparison between matrices H obtained with MNMF and masks P_5 and P_6

Figure 12 illustrates the encoding matrices H obtained with the masks P_5 and P_6. Observing the left graph one can figures out that samples from 51 to 151 (belonging to the two classes Versicolor and Virginica) can be represented using only the first bases w_1. In fact, the elements in w_1 assume almost the maximum value, that is 1, while the elements in w_2 assume values close to zero. This means that in this subset of data there is a linear relationship between the lengths of the iris, but not between the widths. On the contrary samples from 1 to 50 (belonging to Setosa), have been reconstructed using both bases w_1 and w_2. We executed the MNMF on the modified dataset composed by the first fifty samples, with the query mask P_5. The reconstruction error obtained removing samples that have not been well reconstructed is 4.7475×10^{-4} much smaller than that obtained with the entire dataset 1.76×10^{-2}. This result confirms that data belonging to the class Setosa have linear relationships between the lengths and widths.

Similarly from the matrix H on the right, we can observe that samples from 1 to 50 are reconstructed mainly using the basis w_1. In this case it means that there is a linear relationship between the sepal features but not between the petal features. On the contrary, there is a linear relationship between both sepal and petal features in the subset of data composed by samples from 51 to 150. The reconstruction error obtained after removing samples from 1 to 50 is 8.3235×10^{-4} much lower than that obtained with the whole dataset which is 3.5×10^{-3}.

4 Final Remarks

A novel NMF algorithm, namely Masked NMF has been proposed in order to overcome the limitations of classical NMF and to introduce knowledge in the factorization process, making the proposed MNMF algorithm a useful tool for IDA. The query-based approach has been adopted to allow the analyst to specify what parts she is interested to discover. As shown in the numerical examples, the proposed approach is able to extract the subset of data that are actually represented by the parts, discarding the data in the matrix X that do not find a neat representation by the parts and returning the subset of samples that contains the selected parts.

Future work can be addressed to assess the performance of the query based MNMF approach on different real datasets as well as to further investigate its capability of selecting local features hidden in data.

Acknowledgements This work was supported by *"National Group of Computing Science (GNCS-INDAM)"*.

References

1. Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2013), `http://archive.ics.uci.edu/ml`
2. Boutsidis, C., Gallopoulos, E.: Svd based initialization: a head start for nonnegative matrix factorization. Pattern Recognition 41, 1350–1362 (2008)
3. Cai, D., He, X., Wu, X., Han, J.: Non-Negative Matrix Factorization on Manifold. In: Proc. Eighth IEEE Int'l Conf. Data Mining, pp. 63–72 (2008)
4. Yoo, J., Choi, S.: Orthogonal nonnegative matrix tri-factorization for co-clustering: multiplicative updates on Stiefel manifolds. Information Processing and Management 46, 559–570 (2010)
5. Casalino, G., Del Buono, N., Mencar, C.: Subtractive initialization of nonnegative matrix factorizations for document clustering. In: Petrosino, A. (ed.) WILF 2011. LNCS, vol. 6857, pp. 188–195. Springer, Heidelberg (2011)
6. Casalino, G., Del Buono, N., Minervini, M.: Nonnegative matrix factorizations performing object detection and localization. Applied Computational Intelligence and Soft Computing 15 (2012)
7. Casalino, G., Del Buono, N., Mencar, C.: Subtractive clustering for seeding nonnegative matrix factorizations. Information Sciences 257, 369–387 (2014)

8. Chu, M., Del Buono, N., Lopez, L., Politi, T.: On the low rank approximation of data on the unit sphere. SIAM Journal Matrix Analysis Appl. 27(1), 46–60 (2005)
9. Desmarais, M.C.: Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization. In: Conati, C., Ventura, S., Calders, T., Pechenizkiy, M. (eds.) Proceedings of the 4th International Conference on Educational Data Mining, pp. 41–50 (2011)
10. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix tri factorizations for clustering. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 126–135 (2006)
11. Donoho, D., Stodden, V.: When does non-negative matrix factorization give a correct decomposition into parts. In: Advances in Neural Information Processing Systems, vol. 16 (2003)
12. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. J. Machine Learning Research 5, 1457–1469 (2004)
13. Lee, D.D., Seung, S.H.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
14. Lee, D.D., Seung, S.H.: Algorithms for non-negative matrix factorization. In: Proc. Adv. Neural Information Proc. Syst. Conf., vol. 13, pp. 556–562 (2000)
15. Liu, H., Wu, Z., Cai, D., Huang, T.S.: Constrained Non-negative Matrix Factorization for Image Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(7), 1299–1311 (2012)
16. Ma, H., Zhao, W., Tan, Q., Shi, Z.: Orthogonal nonnegative matrix tri-factorization for semi-supervised document co-clustering. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS, vol. 6119, pp. 189–200. Springer, Heidelberg (2010)
17. Shahnaz, F., Berry, M.W., Pauca, V.P., Plemmons, R.J.: Document clustering using nonnegative matrix factorization. Information Processing and Management 42(2), 373–386 (2006)
18. Zhang, Z.Y., Li, T., Ding, C., Ren, X.W., Zhang, X.S.: Binary matrix factorization for analyzing gene expression data. Data Min. Knowl. Discov. 20(1), 28–52 (2010)