Non-Negative Matrix Tri-Factorization for co-clustering: an analysis of the block matrix

N. Del Buono[a], G. Pio[b]

[a]*Dipartimento di Matematica, Università degli Studi di Bari Aldo Moro Via E. Orabona 4, I-70125 Bari, Italy, delbuono@dm.uniba.it.*
[b]*Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro Via E. Orabona 4, I-70125 Bari, Italy, gianvito.pio@uniba.it*

Abstract

Non-negative dyadic data, that is data representing observations which relate two finite sets of objects, appear in several domain applications, such as text-mining-based information retrieval, collaborative filtering and recommender systems, micro-array analysis and computer vision. Discovering latent subgroups among data is a fundamental task to be performed on dyadic data. In this context, clustering and co-clustering techniques are relevant tools for extracting and representing latent information in high dimensional data. Recently, Non-negative Matrix Factorizations attracted a great interest as clustering methods, due to their capability of performing a parts-based decomposition of data. In this paper, we focus our attention on how NMF with additional constraints can be properly applied for co-clustering non-negative dyadic data. In particular, we present a process which aims at enhancing the performance of 3-factors NMF as a co-clustering method, by identifying a clearer correlation structure represented by the block matrix.

Experimental evaluation performed on some common datasets, by applying the proposed approach on two different NMF algorithms, shows that, in most cases, the quality of the obtained clustering increases, especially in terms of average inter-cluster similarity.

Keywords: Co-clustering, non-negative matrix factorization, subspace approximation, text mining

1. Introduction

Several domain applications deal with observational data which relate two finite sets of objects, so that any observation is made on dyads. In the simplest case, an elementary dyadic observation consists in the co-occurrence of a pair of objects taken from each of the two sets in the domain, while other cases provide the strength of preference or association of dyadic pairs. Examples of dyadic data can be found in text-mining-based information retrieval, collaborative filtering and recommender systems, micro-array analysis and computer vision [12, 40].

A fundamental task in the context of unsupervised learning from dyadic data is the structure discovery, that is the identification of subgroups among data [23]. Clustering of one set of objects and co-clustering of both sets can be performed to discover latent relationships synthesized in dyadic data. Co-clustering methods can identify subgroups of documents with similar properties with respect to subgroups of terms in text mining [14, 8]; recognize groups of genes that show similar activity patterns under a specific subset of the experimental conditions in micro-array analysis [31, 7], or with respect to their interactions with microRNAs [32, 33, 34]; discover subgroups of customers with similar preferences or behaviors toward a subset of products in recommender systems [19, 43]. Co-clustering is also useful to automatically perform dimensionality reduction of highly-dimensional data [1].

Many approaches presented in the literature for the clustering task are based on Non-negative Matrix Factorizations (NMF), which aim at providing a minimum error non-negative representation of a data matrix. From the first works of Lee and Seung [25, 26], NMF has been deeply investigated both in theory and in practice and successfully used as a tool in many real applications [2, 4, 9, 24, 38, 39]. For a recent overview, see [20]. In [15] the relationship between NMF (with additional orthogonal constraints on its factors), k-means and spectral-based clustering was demonstrated, while in [35] the mathematical equivalence between orthogonal NMF and a weighted variant of spherical k-means was proved together with some indications about the cases in which orthogonal NMF should be preferred to k-means and spherical k-means. Moreover, experiments performed in [42] showed that some NMF-based approaches outperform spectral-based methods achieving higher accuracy and efficiency in performing document clustering.

Focusing on the co-clustering task, in [28] the authors proposed the Block Value Decomposition (BVD) to explore the latent block structure in dyadic

data matrices by means of a tri-factorization, without any additional constraint. Nonnegative Matrix Tri-factorization (NMTF) was firstly proposed in [16] to co-cluster words and documents at the same time, while a new multiplicative updating algorithm with orthogonality constraints was developed in [44], demonstrating its usefulness in revealing polysemous words. Due to its encouraging empirical results, NMTF methods have been further investigated to address various aspects of co-clustering such as: graph regularization for clustering data on manifolds [21], semi-supervised sentiment analysis for combining some direct and explicit knowledge in text mining [27], treatment of heterogeneous data [6].

Other studies have been conducted for the definition of novel NMTF algorithms to efficiently perform co-clustering on large data sets in real word applications [41] and to compare different approaches to try to identify which algorithms appear more suitable for particular situations [18]. More recently, NMTF has been extended by introducing different forms of structure constraint on the factors, to integrate some prior knowledge and/or domain information in the factorization process in order to achieve more interpretable results [6, 17, 30, 37].

It is noteworthy that most of the recent works on NMTF has been conducted on the identification of proper constraints to impose on the factors in order to optimize some particular aspects of the co-clustering. However, there is no attempt to focus on the structure of the (central) block matrix, which represents the correlation strengths between row and column clusters. Observing such a matrix it is possible to analyze the correlation between groups identified on a dimension with respect to those identified on the other dimension. In this paper, we present an approach which aims at optimizing the correlation structure represented by the block matrix. Such optimization is performed by removing correlations which appear noisy, with the main goal of obtaining *better-separated* co-clusters.

In the following section, a brief review about clustering/co-clustering through NMF (with 2 and 3 factors) is reported, considering the common domain of textual documents. In Section 3, we detail the extraction of a clearer correlation structure and its exploitation as background knowledge for a further execution of any iterative algorithm. Some details about the time complexity of the proposed procedure are also provided. In Section 4, some experimental results are reported in order to evaluate the effect of the proposed procedure in discovering a clean correlation structure between document and term clusters and to use such a structure as a starting point in

some NMTF algorithms. These experiments substantially show that the proposed approach is able to reduce the average inter-cluster similarity, without affecting the intra-cluster similarity. Finally, some conclusions and future works are sketched in Section 5.

2. NMF for clustering and co-clustering: a review

In this section, we briefly review related work on NMF for clustering documents and for co-clustering documents and terms.

2.1. Clustering by 2-Factor NMF

The classical 2-factor NMF approximates a term/document matrix $X \in \mathbb{R}_+^{n \times m}$ 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 $X \approx WH$, where $k < \min(m, n)$.

These approximating factors are typically obtained by solving the constrained least square minimization problem:

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

where $\| \cdot \|_F$ indicates the Frobienus matrix norm. Different objective functions have been proposed in the literature, some of them are also designed to incorporate additional constraints into the factors W and H. However, most of the proposed objective functions (included, the least squared problem formulated in (1)) can be brought back to the general framework of Bregman divergences as described in [10, 11, 13].

The factors W and H in the 2-factor NMF are able to provide basic document or term clustering. In fact, considering the columns of W as document cluster centroids, the j-th document of the input matrix X can be associated to the centroid c_j (i.e., to its corresponding cluster) which gives the maximum contribute in the linear combination:

$$c_j = \operatorname*{argmax}_z H_{zj}. \tag{2}$$

Furthermore, applying the column-sum-to-1 normalization, given by:

$$X' = XD_X^{-1} \quad W' = WD_W^{-1},\ H' = HD_H^{-1}, \tag{3}$$

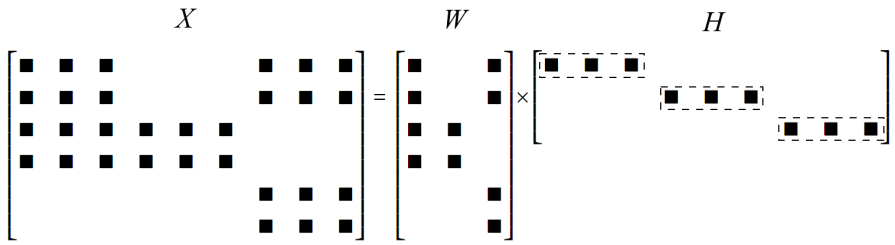

Figure 1: Document clustering ($k = 3$), with orthogonality constraints on the rows of H. $X \in \mathbb{R}_+^{6\times 9}$ is factorized in two matrices $W \in \mathbb{R}_+^{6\times 3}$ and $H \in \mathbb{R}_+^{3\times 9}$. The columns of W represent the document cluster centroids while on each row of H it is possible to group the documents. In the example, 1st to 3rd documents are in the first cluster, 4th to 6th documents are in the second cluster, 7th to 9th documents are in the third cluster.

(where D_X, D_W, D_H are diagonal matrices containing the column sums of X, W and H respectively) the values H'_{zj} can be interpreted as the *probability* that the j-th document belongs the z-th document cluster. Analogously, the row-sum-to-1 normalization allows to identify the *probability* that the i-th term belongs the j-th term cluster.

It is noteworthy that the basic NMF provides an *almost casual* clustering. In order to obtain a solution that guarantees a real clustering interpretation, additional orthogonality constraints on W and/or H should be imposed.

In [16], the minimization problem (1) was modified to generate a real document clustering, by imposing the orthogonality constraint on the rows of H, as follows:

$$\min_{W\geq 0, H\geq 0} \|X - WH\|_F^2, \text{ s.t. } HH^\top = I, \tag{4}$$

Graphical representation of the clusters derived by (4) are illustrated in Figure 1, by considering the maximum value for each column of the matrix H and grouping the documents on the same row. In the same manner, a term clustering can be achieved by requiring that the orthogonality constraint is satisfied by the columns of W (i.e., $W^\top W = I$), as illustrated in Figure 2.

The addition of the orthogonality constraint on the rows of the matrix H let the document centroids (the columns of the matrix W) arrange themselves such that each document is really closer to only one cluster centroid and it is far from the others. In this way, the minimization of the intra-cluster variance and the maximization of the inter-cluster variance can be obtained for column clustering of the input data matrix.

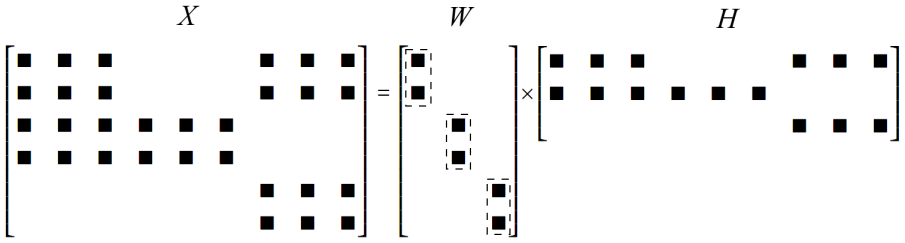

Figure 2: Term clustering ($k = 3$) with orthogonality constraints on the columns of W. $X \in \mathbb{R}_+^{6\times 9}$ is factorized in two matrices $W \in \mathbb{R}_+^{6\times 3}$ and $H \in \mathbb{R}_+^{3\times 9}$. The rows of H represent the three term cluster centroids while on each column of the matrix W it is possible to group the terms. In the example, 1st and 2nd terms are in the first cluster, 3rd and 4th terms are in the second cluster, 5th and 6th terms are in the third cluster.

It is noteworthy that the achievement of the true orthogonality condition[1] depends on the specific algorithm used to solve the orthogonal NMF problem. In this paper, we refer to the *soft-orthogonality* constraint, since most of NMF algorithms aim at satisfying approximately, i.e. not exactly, the true orthogonality. However, recently, in [35] two algorithms are proposed that perfectly satisfy the orthogonality constraint at each iteration.

2.2. Co-clustering by 3-Factor NMF

Co-clustering of both documents and terms can be performed by simultaneously constraining W and H to be orthogonal, i.e., solving the constrained optimization problem:

$$\min_{W\geq 0, H\geq 0} \|X - WH\|_F^2, \text{ s.t. } HH^\top = I, W^\top W = I. \tag{5}$$

However, this double orthogonality constraint is really too restrictive, and the solutions of (5) result a rather poor low-rank approximation of the data matrix X. In fact, we are asking that the document centroids (i.e., the columns of W) have to arrange themselves in order to reach the orthogonality of the rows of the matrix H and, in the same time, we require that the term centroids (i.e., the rows of H) have to arrange themselves in order to reach the orthogonality of the columns of the matrix W. Moreover, the number of terms and document clusters must be the same. To provide additional degree

[1] We refer to *true orthogonality* to intend that orthogonality among vectors is reached (by the algorithm) at the machine precision.

of freedom, we need to add an extra factor in the factorization of the original data matrix X. This new matrix, in fact, allows the low-rank approximation to remain accurate, while the soft-orthogonality of the columns of W and the rows of H is obtained.

Non-negative Matrix Tri-Factorization (NMTF, or 3-factors NMF, also known as *Non-negative Block Value Decomposition (NBVD)* [28]) approximates a term/document matrix as $X \approx USV$, where $U \in \mathbb{R}_+^{n \times l}$ is the *row-coefficient matrix*, $S \in \mathbb{R}_+^{l \times k}$ is the *block matrix*, $V \in \mathbb{R}_+^{k \times m}$ is the *column-coefficient matrix*, the values k and l represent the number of document and term clusters, respectively (generally, $k \ll \min(n, m)$ and $l \ll \min(n, m)$).

Adding the orthogonality constraints on both U and V matrices, as discussed in [16, 44], the optimization problem becomes:

$$\min_{U \geq 0, S \geq 0, V \geq 0} \|X - USV\|_F^2, \text{ s.t. } UU^\top = I, V^\top V = I. \tag{6}$$

In this way, columns and rows are clustered simultaneously and both orthogonality constraints can be satisfied keeping a good low-rank approximation. Conveniently grouping the factors U, S and V, an equivalent 2-factors NMF can be derived. Particularly, for document clustering we put:

$$W = US, \quad H = V. \tag{7}$$

In this way, the columns of the matrix US are the basis vectors of the column space of X (i.e., the document centroids) and, accordingly to (2), the elements in each row of the matrix V belong to the same cluster.

Analogously, for term clustering, we consider:

$$W = U, \quad H = SV, \tag{8}$$

such that the rows of the matrix SV represent the basis vectors of the row space of X(i.e., the term centroids) and the elements of each column of the matrix U belong to the same cluster.

Starting from the above considerations, it is questionable whether there exists any difference between two separate 2-factors NMF clusterings (one for column space and another for row space) and one simultaneous clustering with 3-factors NMF, also in terms of clustering quality. Qualitative descriptions illustrated by Figures 1, 2 and 3, together with the application of Equations (7) and (8), suggest the equivalence of the two approaches. However, as we will show in the next section, a deep analysis of the block matrix S could be helpful to emphasize the real advantages related to the 3-factors NMF approach.

3. Block Matrix Optimization

As shown in the previous section, both 2-factors and 3-factors NMF can be used to group documents and terms. Moreover, imposing additional orthogonality constraints leads to obtain clusters which minimize the intra-cluster variance and maximize the inter-cluster variance. However, in order to satisfy such orthogonality constraints as much as possible on the row-coefficient matrix and/or on the column-coefficient matrix, the discovered block matrix could appear "noisy", since not optimized with respect to any given criterion. In particular, the block matrix could present an unclear structure representing the correlations between document and term clusters, that is, could contain significant (i.e. non-zero) values in (almost) all cells. This is not a desirable result, since it describes a situation in which (almost) all document clusters are related to (almost) all term clusters (low inter-cluster variance). Such a situation can be mainly due to:

1. noise in the data, which let the algorithm identify correlations between document and term clusters which actually do not exist;
2. high overlap among the categories represented by the clusters.

In order to face the first issue, we propose a method which is able to extract a clearer correlation structure and to exploit it as a background knowledge for a further execution of any iterative algorithm. Moreover, we show that the application of proposed method allows us to better identify the presence of the second issue, which is often due to an inappropriate choice of the number of clusters k and l.

As previously stated, each (i, j)-th element of the block matrix S represents the *correlation strength* between the term cluster i and the document

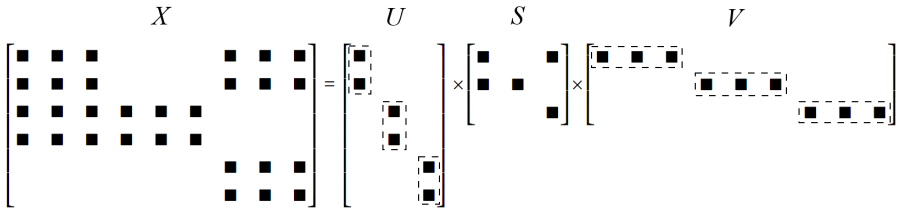

Figure 3: Simultaneous document and term clustering ($k = 3$ and $l = 3$). $X \in \mathbb{R}_+^{6\times 9}$ is factorized in the three matrices $U \in \mathbb{R}_+^{6\times 3}$, $S \in \mathbb{R}_+^{3\times 3}$ and $V \in \mathbb{R}_+^{3\times 9}$. On each row of the matrix V it is possible to group the documents, while on each column of the matrix U it is possible to group the terms. The results are the same of that of Figures 1 and 2.

cluster j. Observing the values of a block matrix, there will usually be some (non-zero) values significantly lower than the others in the matrix. Our aim is to identify which are the most significant values to be used to define a structure for S, which will be then considered as background knowledge for a further execution of an iterative algorithm. However, since the significance of each value is relative to the other values, a threshold-based approach cannot be applied. For example, in (9) the value 2.2×10^2 can be considered low, if compared to the value 3.2×10^7, but appears high if compared to 8.8×10^{-1}.

$$S = \begin{bmatrix} 8.8 \times 10^{-1} & 1.4 \times 10^{-1} & 5.2 \times 10^{-2} & 3.6 \times 10^{-4} & 3.2 \times 10^{7} & 5.5 \times 10^{-8} \\ 7.8 \times 10^{-2} & 3.2 \times 10^{2} & 8.2 \times 10^{-4} & 7.1 \times 10^{-2} & 3.2 \times 10^{-3} & 3.4 \times 10^{-6} \\ 1.2 \times 10^{-4} & 2.2 \times 10^{2} & 3.8 \times 10^{2} & 4.1 \times 10^{-6} & 3.2 \times 10^{-8} & 3.6 \times 10^{2} \\ 4.1 \times 10^{-6} & 8.8 \times 10^{-6} & 2.6 \times 10^{-7} & 6.2 \times 10^{2} & 3.2 \times 10^{-1} & 1.2 \times 10^{-10} \end{bmatrix} \quad (9)$$

The whole clustering process will then consist of three main steps:

1. *preliminary clustering*, which is devoted to compute U, S and V by adopting any existing iterative algorithm for 3-factors NMF;
2. *correlation structure extraction (CSE)*, which extracts a clearer structure for S, starting from that identified in the first step;
3. *clustering refinement*, which consists in a further execution of an iterative algorithm for 3-factors NMF, with the aim of refining the matrix S, on the basis of the identified structure, and of computing the new factors U and V accordingly.

In the following subsections, we focus on the strategy adopted for the second step using the matrix (9) as a toy example, starting from the following observations:

- the maximum value of the matrix has surely to be considered in the identification of the structure of S (minimizes intra-cluster variance);
- values of the same order of magnitude can be considered equally strong;
- each document/term cluster should be related to the smallest possible number (at least one) of term/document clusters (maximizes intra-cluster variance).

According to such observations, we define the procedure to identify the structure from S as follows:

1. Build the matrix S', taking only the strongest correlation value from S, for each column and row cluster, by iteratively ($min(k, l)$ times):
   (a) taking the current maximum value from S;
   (b) removing from S all the values on the same row and column.
2. Build the matrix S'', by replacing zero elements of S' with values of S which are actually of the same order of magnitude of (or greater than) those in S';
3. Build the matrix S''' from S'', by taking from S the maximum value for each column and row of S'' which does not contain any non-zero element.

3.1. Building S': taking the strongest correlation values

In the following, we report the result obtained applying the first step to the matrix (9):

- The maximum value is $S_{1,5} = 3.2 \times 10^7$. Take it and remove the 1st row and the 5th column, to obtain:

$$S^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 7.8 \times 10^{-2} & 3.2 \times 10^2 & 8.2 \times 10^{-4} & 7.1 \times 10^{-2} & 0 & 3.4 \times 10^{-6} \\ 1.2 \times 10^{-4} & 2.2 \times 10^2 & 3.8 \times 10^2 & 4.1 \times 10^{-6} & 0 & 3.6 \times 10^2 \\ 4.1 \times 10^{-6} & 8.8 \times 10^{-6} & 2.6 \times 10^{-7} & 6.2 \times 10^2 & 0 & 1.2 \times 10^{-10} \end{bmatrix}$$

$$S' = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{3.2 \times 10^7} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- The maximum value is $S^{(1)}_{4,4} = 6.2 \times 10^2$. Take it and set remove the 4th row and 4th column to obtain:

$$S^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 7.8 \times 10^{-2} & 3.2 \times 10^2 & 8.2 \times 10^{-4} & 0 & 0 & 3.4 \times 10^{-6} \\ 1.2 \times 10^{-4} & 2.2 \times 10^2 & 3.8 \times 10^2 & 0 & 0 & 3.6 \times 10^2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$S' = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{3.2 \times 10^7} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{6.2 \times 10^2} & 0 & 0 \end{bmatrix}$$

- The maximum value is $S^{(2)}_{3,3} = 3.8 \times 10^2$. Take it and remove the 3th row and 3th column to obtain:

$$S^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 7.8 \times 10^{-2} & 3.2 \times 10^2 & 0 & 0 & 0 & 3.4 \times 10^{-6} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$S' = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{3.2 \times 10^7} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{3.8 \times 10^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{6.2 \times 10^2} & 0 & 0 \end{bmatrix}$$

- Finally, the maximum value is $S^{(3)}_{2,2} = 3.2 \times 10^2$. Take it and remove the 2th row and 2th column to get:

$$S^{(4)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$S' = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{3.2 \times 10^7} & 0 \\ 0 & \mathbf{3.2 \times 10^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{3.8 \times 10^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{6.2 \times 10^2} & 0 & 0 \end{bmatrix}$$

3.2. Building S'': taking the values of the same order of magnitude

As outlined in the general procedure, in order to build the matrix S'', we have to take from S, for each row and column, the values of the same order of magnitude of (or greater than) the values in S'. However, performing this selection only on the basis of the exponent of each value produces poor results. Indeed, the values 9.99×10^4 and 1.00×10^4 would be considered similar, while the values 1.00×10^4 and 9.99×10^3 would not. In order to alleviate such limitation, in the following we define a heuristic selection strategy, which aims at better identifying the values to select.

Let us suppose we want to identify all the values that can be considered of the same order of magnitude of $y \times 10^p$, where $1 \leq y < 10$. The whole interval $[10^p; 10^{p+1})$ should be considered only when the value represents the mean of the interval, i.e. when $y = 5.5$. Obviously, every greater value should be taken, leading to the interval $[10^p; +\infty)$. Generalizing, we impose that when $y \geq 5.5$, the interval to be taken into account is:

$$[(y - 4.5) \times 10^p; +\infty) \tag{10}$$

When $1 \leq y < 5.5$, we consider the percentage of the covered interval width, i.e. $\frac{y-1}{4.5}$, and take the residual percentage, i.e. $\left(1 - \frac{y-1}{4.5}\right)$ from the higher half of the lower interval, i.e. from $[5.5 \times 10^{p-1}; 10^p)$ whose width is $4.5 \times 10^{p-1}$.

The interval to consider then becomes $\left[10^p - \left(1 - \frac{y-1}{4.5}\right) \times 4.5 \times 10^{p-1}; +\infty\right)$, which can be simplified to:

$$\left[10^p - (5.5 - y) \times 10^{p-1}; +\infty\right) \tag{11}$$

Taking, for each value of the matrix S' obtained in the previous step, all the values (in the same row and column) belonging to the intervals defined by (10) and (11), we obtain:

$$S'' = \begin{bmatrix} 0 & 0 & 0 & 0 & 3.2 \times 10^7 & 0 \\ 0 & 3.2 \times 10^2 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{2.2 \times 10^2} & 3.8 \times 10^2 & 0 & 0 & \mathbf{3.6 \times 10^2} \\ 0 & 0 & 0 & 6.2 \times 10^2 & 0 & 0 \end{bmatrix} \tag{12}$$

3.3. Building S''': filling all-zero columns and rows

Since each term/document cluster should be related (even if weakly) to at least one document/term cluster, we impose that no all-zero columns and rows should appear in the final block matrix. At this aim, we build S''' by taking from S the maximum value for each all-zero row and column:

$$S''' = \begin{bmatrix} \mathbf{8.8 \times 10^{-1}} & 0 & 0 & 0 & 3.2 \times 10^7 & 0 \\ 0 & 3.2 \times 10^2 & 0 & 0 & 0 & 0 \\ 0 & 2.2 \times 10^2 & 3.8 \times 10^2 & 0 & 0 & 3.6 \times 10^2 \\ 0 & 0 & 0 & 6.2 \times 10^2 & 0 & 0 \end{bmatrix} \tag{13}$$

It is noteworthy that the CSE process is able to highlight the possibility that a too much high value of k or l has been chosen. Indeed, the presence of an all-zero row or column could be due to the absence of strong correlations between the corresponding row/column cluster and the other column/row clusters, otherwise difficult to observe on the original matrix S. In this case, in real-world situations (i.e. when the number of categories is really unknown), a refinement on the number of clusters could be performed instead of applying the third step the CSE procedure.

The matrix S''' represents the *ideal structure* we impose on the block matrix S as a starting point for a further execution of an iterative algorithm. Every iterative algorithms with element-wise multiplicative update rules will preserve the "a priori" defined structure.

3.4. Time Complexity Analysis

In this subsection, we report some details about the time complexity of the proposed procedure. In general, the whole clustering process, as outlined in Section 3, could lead to a higher running time with respect to a single run of an NMF algorithm. However, observing the CSE procedure, it is noteworthy that it works only on the matrix S, which is inherently very small (namely, it contains $k \cdot l$ elements).

In particular, to build the matrix S', we need to find the maximum values for $min(k, l)$ times, among $k \cdot l$ possible values, removing the values on the

same row and on the same column. Therefore, in the worst case, for each extracted value, we have to perform a complete scan of the matrix S (i.e., $k \cdot l$ operations) and a scan of the row and column it belongs to (i.e., $k + l$ operations). This means that the time complexity for the first step is:

$$\begin{aligned} min(k,l)[k \cdot l + (k+l)] &= \mathcal{O}(min(k,l)[k \cdot l + max(k,l)]) \\ &= \mathcal{O}(min(k,l)[max(k,l) \cdot min(k,l) + max(k,l)]) \\ &= \mathcal{O}(min(k,l)^2 \cdot max(k,l)) \end{aligned} \tag{14}$$

The construction of the matrix S'' requires a scan of row and column values (i.e., $k + l$ operations) to which each of the $min(k, l)$ extracted value belongs. This means that the time complexity of the second step is:

$$min(k,l) \cdot (k+l) = \mathcal{O}(min(k,l) \cdot max(k,l)) = \mathcal{O}(k \cdot l) \tag{15}$$

The construction of the matrix S''' requires, in the worst case, to find the maximum value for each row and column. Therefore, the worst-case time complexity to build S''' (i.e., when a complete scan of the matrix is performed) is:

$$k \cdot l = \mathcal{O}(k \cdot l) \tag{16}$$

By combining Equations (14), (15) and (16), the whole CSE process has a time complexity of:

$$\mathcal{O}(min(k,l)^2 \cdot max(k,l)) = \mathcal{O}(min(k,l) \cdot (k \cdot l)) \tag{17}$$

Since k and l are very small if compared to n and m, we can conclude that, asymptotically, the time required to perform the CSE process is negligible with respect to the time required for the factorization.

Moreover, although the proposed approach requires two executions of the NMF algorithm, the final time complexity is not affected, since, in the worst case, it is increased by a constant factor of 2.

4. Experiments

We evaluated the performance of the CSE process by adopting two existing algorithms for tri-factorization. In particular, we applied the non-negative Block Value Decomposition algorithm [28] (henceforth denoted by N) and the algorithm proposed in [44] (henceforth denoted by O), which

imposes the orthogonality constraint on both the matrices U and V. It is noteworthy that, in this experimental evaluation, our focus is to evaluate the effect of the CSE process in discovering a clean correlation structure between document and term clusters when NMTF algorithms are adopted. Comparison with other topic-based algorithms (such as LDA [3]) or with other competitive clustering/co-clustering approaches (such as [17, 32]) is out of the scope of this paper.

Firstly, we evaluated each single algorithm, with and without the application of the CSE process. Then, we evaluated the possibility to use an algorithm for discovering the initial structure of the matrix S and the other one for the clustering refinement step, with and without the application of the CSE process. In this way, the first two steps (preliminary clustering and correlation structure extraction) can be seen as an initialization strategy of the matrix S for the iterative algorithm adopted to perform the clustering refinement step.

The initialization of the matrices U and V was performed randomly, according to a uniform distribution in $[0, 1]$, since it has been empirically observed in [5] that other initialization mechanisms, such as those based on feature extraction and on prototype-based clustering, do not significantly improve the performances of iterative NMF algorithms in terms of the overall error. Results reported in the following sections represent the average values obtained over ten different random initializations. It is noteworthy that, in order to guarantee a fair comparison, the same initial matrices were considered for all the NMF algorithms.

As regards the stopping criterion, we required that the algorithms stop when $||E^{(i)} - E^{(i-1)}||_F < 10^{-4}$, where $||E^{(i)}||_F$ is the Frobenius Norm of the error matrix, at the i-th iteration.

4.1. Datasets

The main characteristics of the datasets used in the evaluation are summarized in Table 1, while a short description is provided in the following.

- **CSTR.** This dataset contains 639 abstracts, belonging to four categories of technical reports published in the Department of Computer Science at The University of Rochester[2].

[2] www.cs.rochester.edu/trs/

Dataset	# documents	# terms	# classes
CSTR	639	4016	4
WebKB4	2803	7287	4
Newsgroups10	500	13709	10
Reuters	5485	14551	8
k1a	2340	21839	20
k1b	2340	21839	6
wap	1560	8460	20
la12	6279	20009	6
sports	8580	14870	7

Table 1: Datasets statistics summary

- **WebKB4.** The dataset contains $2,803$ documents belonging to the four most populous categories[3] of WebKB, a set of web pages collected from computer science departments of various universities in 1997.

- **Newsgroups10.** This dataset contains a subset of 500 documents[3] belonging to the top-10 categories of the Newsgroups20 dataset. This dataset is used to evaluate the effectiveness of the CSE process in the case of datasets with a small number of documents for each category and a relatively high number of categories.

- **Reuters.** This dataset contains $5,485$ documents belonging to the top-8 categories[3] of Reuters-21578, a collection of Reuters newswires collected in 1987. The dataset was obtained by selecting only documents associated with a single category. A more recent dataset, Reuters RCV1-v2, is also available, but it results unsuitable because it has a multi-label categorization, whereas our experiments are conceived on hard clustering (i.e., each document is assigned to a single category).[4]

- **k1a, k1b, wap.** These datasets[5] have been built for the WebACE project [22] and consist of web pages in various subject directories of

[3]Defined starting from the training set at: http://web.ist.utl.pt/∼acardoso/datasets/

[4]Selecting only documents with a single label in Reuters RCV1-v2 leads to preserve only the 3% of documents, affecting the significance of the evaluation on this dataset.

[5]Available in the CLUTO toolkit: glaros.dtc.umn.edu/gkhome/cluto/cluto/download

Yahoo!. k1a and k1b contain exactly the same set of documents, but they are assigned to a different number of categories (i.e. documents in k1a are subdivided into more detailed categories). It is noteworthy that k1a and wap are other examples of datasets with a small percentage of documents for each category and a relatively high number of categories.

- **la12, sports**. These datasets[5] from TREC are all newspaper stories from either the LA Times (la12 dataset) or San Jose Mercury (sports datasets) classied into different topics.

Each dataset has been pre-processed using the Text to Matrix Generator (TMG) tool [45], which allowed us to remove stop-words (according to the integrated common words dictionary), to apply a standard stemming algorithm (i.e. Porter's algorithm [36]) and to generate the term-document matrix. The matrix generation has been performed according to the standard TF-IDF weighting, i.e. $tf(t,d) \cdot log\frac{|D|}{|\{d \in D : t \in d\}|}$, where $tf(t,d)$ is the frequency of the term t in the document d, and D is the whole collection of documents. Moreover, document vectors have been normalized according to the L2-norm, in order to remove any possible bias introduced by the different length of documents.

4.2. Evaluation measures

The evaluation has been performed on the basis of four measures, in order to take into account both the clustering internal quality and the clustering external quality. k and l have been set to the number of categories of the dataset. In the following, we describe the considered evaluation measures:

- **Intra-cluster average similarity**: measures the average similarity among documents belonging to the same cluster. The average similarity for a given cluster C_z is defined as follows:

$$IntraSim(C_z) = \frac{\sum_{i=1}^{m_z} \sum_{j=i+1}^{m_z} sim(d_i, d_j)}{0.5 \times m_z \times (m_z - 1)},$$

where d_i, d_j are documents of the cluster C_z, m_z is the number of documents of C_z and $sim(d_i, d_j)$ is a similarity function (cosine similarity, in our case). The intra-cluster average similarity is defined as:

$$IntraSim = \frac{\sum_{i=1}^{k} IntraSim(C_i) \times m_i}{m},$$

where m is the total number of documents and k is the number of document clusters.

- **Inter-cluster centroids average similarity:** measures the average similarity among documents belonging to different clusters. We calculate the similarity between cluster centroids, instead of calculating the average between all the documents. The measure is defined as follows:

$$InterSim = \frac{\sum_{i=1}^{k} \sum_{j=i+1}^{k} sim(centroid_i, centroid_j)}{0.5 \times k \times (k-1)},$$

where $centroid_i$ is the i-th cluster centroid, k is the number of clusters and $sim(centroid_i, centroid_j)$ is the cosine similarity function.

- **Clustering Accuracy (CA):** measures the percentage of correctly clustered documents. We used the Kuhn-Munkres algorithm [29] to map each cluster to an original category.

  Given a document d_i, its cluster label cl_i and its true label tl_i, the Clustering Accuracy is defined as follows:

$$CA = \frac{\sum_{i=1}^{m} \delta\left(cl_i, map\left(tl_i\right)\right)}{m},$$

  where m is the total number of documents, $\delta(x, y) = 1$ if $x = y$ and $\delta(x, y) = 0$ otherwise and $map(tl_i)$ is the mapping function obtained by the Kuhn-Munkres algorithm.

- **Normalized Mutual Information (NMI):** this measure is based on the mutual information (MI) between two sets of clusters C (the original document categorization) and C' (the clustering result):

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \log_2 \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)},$$

  where $p(c_i)$ and $p(c'_j)$ represent the probabilities that an arbitrarily taken document belongs to the cluster c_i and c'_j, respectively, and $p(c_i, c'_j)$ represents the joint probability that this arbitrarily taken document belongs to the cluster c_i and c'_j at the same time.

This measure is normalized in the $[0, 1]$ interval as follows:

$$NMI(C, C') = \frac{MI(C, C')}{\max\left(E(C), E(C')\right)},$$

where $E(C)$ and $E(C')$ are the entropy of the set C and C', respectively.

4.3. Results analysis

Dataset	N	$N + N^*$	O	$O + O^*$	$N + O$	$N + O^*$	$O + N$	$O + N^*$
CSTR	0.0462	0.0461	0.0552	0.0534	0.0451	0.0452	0.0545	0.0545
WebKB4	0.0389	0.0403	0.0396	0.0388	0.0385	0.0390	0.0389	0.0378
Newsgroups10	0.0418	0.0409	0.0389	0.0387	0.0362	0.0362	0.0395	0.0396
Reuters	0.0938	0.0966	0.0850	0.0833	0.0841	0.0837	0.0913	0.0944
k1a	0.0133	0.0128	0.0118	0.0082	0.0103	0.0103	0.0107	0.0107
k1b	0.0085	0.0084	0.0106	0.0080	0.0071	0.0071	0.0084	0.0084
wap	0.0361	0.0370	0.0373	0.0320	0.0294	0.0296	0.0384	0.0392
la12	0.1410	0.1430	0.1450	0.1260	0.1420	0.1460	0.1480	0.1480
sports	0.1520	0.1600	0.1600	0.1570	0.1520	0.1490	0.1600	0.1600

Table 2: Intra-clusters average similarity. The application of CSE is indicated with the symbol *. For each pair of columns and for each row, significantly better values are reported in boldface.

Dataset	N	$N + N^*$	O	$O + O^*$	$N + O$	$N + O^*$	$O + N$	$O + N^*$
CSTR	0.2694	**0.0280**	0.1380	0.1179	0.2196	**0.1863**	0.2541	0.2541
WebKB4	0.2363	0.2346	0.1817	**0.0523**	0.2999	**0.1756**	0.1756	**0.0315**
Newsgroups10	0.0943	**0.0310**	0.0922	**0.0271**	0.0129	0.0128	0.0608	0.0608
Reuters	**0.0261**	0.0553	0.0736	**0.0189**	0.0095	0.0090	**0.0831**	0.1422
k1a	0.0400	**0.0138**	0.0289	**0.0087**	0.0029	0.0029	0.0201	**0.0135**
k1b	0.0327	**0.0162**	0.0737	**0.0521**	0.0148	0.0148	0.0161	0.0161
wap	0.1800	0.2168	0.0327	0.0351	0.0047	0.0040	0.1432	0.1436
la12	0.3528	0.3724	0.1960	**0.0539**	0.2744	**0.0994**	0.1823	0.1842
sports	0.1836	**0.0516**	0.1240	**0.0172**	0.0160	**0.0065**	0.1944	0.1808

Table 3: Inter-clusters centroids average similarity (lower is better). The application of CSE is indicated with the symbol *. For each pair of columns and for each row, significantly better values are reported in boldface.

Tables 2 - 5 report the results obtained by the execution of the considered NMF algorithms and their hybridization, with and without the application of the CSE process.

Dataset	N	$N+N^*$	O	$O+O^*$	$N+O$	$N+O^*$	$O+N$	$O+N^*$
CSTR	0.5117	0.5008	0.7214	**0.7653**	0.5728	**0.7684**	0.7042	0.7121
WebKB4	0.6223	**0.6460**	**0.6589**	0.5583	**0.5799**	0.5167	0.6621	0.6628
Newsgroups10	0.6360	0.6260	0.6200	0.6080	0.5020	0.5020	0.5380	0.5445
Reuters	0.3987	**0.4122**	**0.3985**	0.3404	0.4458	**0.4627**	0.4162	0.4146
k1a	0.3585	0.3594	**0.3346**	0.2179	0.2329	0.2329	0.2756	0.2795
k1b	0.5436	**0.5829**	0.5684	**0.5842**	0.5684	0.5684	0.5825	0.5825
wap	**0.5026**	0.4295	**0.4718**	0.2788	0.2391	0.2385	0.5244	**0.5513**
la12	0.4254	**0.4950**	**0.4160**	0.3856	**0.4756**	0.4440	0.4381	0.4400
sports	0.3642	0.3724	0.3789	0.3653	**0.4934**	0.4667	0.3969	0.3804

Table 4: Clustering Accuracy results. The application of CSE is indicated with the symbol *. For each pair of columns and for each row, significantly better values are reported in boldface.

Dataset	N	$N+N^*$	O	$O+O^*$	$N+O$	$N+O^*$	$O+N$	$O+N^*$
CSTR	**0.3906**	0.3670	0.6302	**0.6797**	0.4317	**0.6904**	0.6310	**0.6500**
WebKB4	0.3486	**0.3807**	**0.3851**	0.3514	0.2850	**0.3255**	0.3920	0.3943
Newsgroups10	0.6288	0.6317	0.6255	0.6020	0.4891	0.4917	0.5837	0.5879
Reuters	0.3678	0.3724	0.2966	**0.3378**	0.2893	0.2978	0.2862	**0.3338**
k1a	0.3359	**0.3933**	0.2725	0.2615	0.0861	0.0861	0.2479	0.2460
k1b	0.0490	**0.2472**	0.1741	0.1758	0.0223	0.0223	0.2483	0.2461
wap	0.5015	0.4982	**0.5067**	0.2577	0.1741	0.1943	0.5070	**0.5354**
la12	0.2086	**0.2936**	0.2019	0.1809	0.2629	0.2450	0.2758	0.2808
sports	0.3079	0.2977	0.2812	0.2662	**0.3418**	0.2980	0.3299	0.3209

Table 5: Normalized Mutual Information results. The application of CSE is indicated with the symbol *. For each pair of columns and for each row, significantly better values are reported in boldface.

Results obtained in terms of intra-clusters average similarity (Table 2) show that there are no significant changes when the CSE process is applied. On the contrary, results in terms of inter-clusters similarity (Table 3) show that the application of the CSE process almost always lead to an improvement (lower values), which, in some cases, appear very appreciable (see, for example, results obtained on WebKB4, la12 and sports datasets).

As regards the Clustering Accuracy results (Table 4), the situation appears less clear. Indeed, in some cases, the application of CSE process leads to an improvement (e.g. with the algorithm N), whereas in other cases it leads to a decrease of accuracy (e.g. with the algorithm O). Finally, observing Table 5, it is possible to see that the application of the CSE process is, in

some cases, able to slightly improve results in terms of Normalized Mutual Information.

It should be pointed out that the CSE process performs a structure simplification. When the number of categories is relatively small and the correlation between document and term clusters is clear, the structure simplification applied by the CSE process can significantly improve the results. An example is given by the CSTR dataset and the combination N + O* (with respect to N + O). In this case, the CSE process leads to an improvement in terms of all the considered measures. This result emphasizes the ability of the CSE procedure to discard noisy correlation between document and term clusters, as showed in Figure 4.

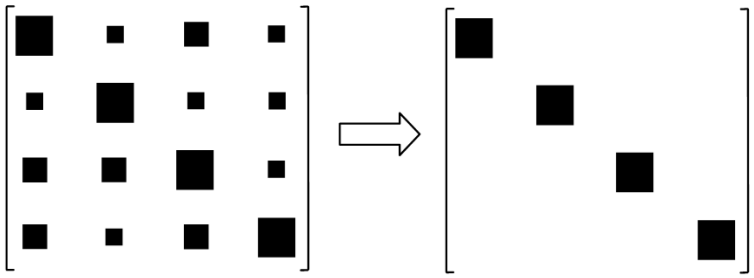

Figure 4: Application of the CSE procedure on the matrix S discovered by the algorithm N on the dataset CSTR. The size of each square represents the order of magnitude of each value. The obtained structure is almost ideal, that is each document cluster is strongly correlated to a single term cluster.

On the other hand, when the categorization of the dataset is detailed (i.e. with a high number of fine-grained classes), the CSE process could apply a strong simplification of the structure, especially if the number of clusters to extract is not appropriated. We show this possible issue on the datasets k1a and k1b, since they consist of the same set of documents, categorized in 20 and 6 classes, respectively.

Figure 5 shows the application of the CSE process to the matrix S extracted by the algorithm O on the dataset k1b (6 categories). In this case, as for the CSTR dataset (Figure 4), the extracted structure appears clear, i.e. each document cluster is strongly correlated to a single term cluster, and the CSE process is able to discard noisy correlations. Such a result is reflected on the performance obtained with respect to all the considered evaluation measures. On the contrary, the structure extracted by the dataset $k1a$ emphasizes a different situation. Indeed, from Figure 6 we observe that:

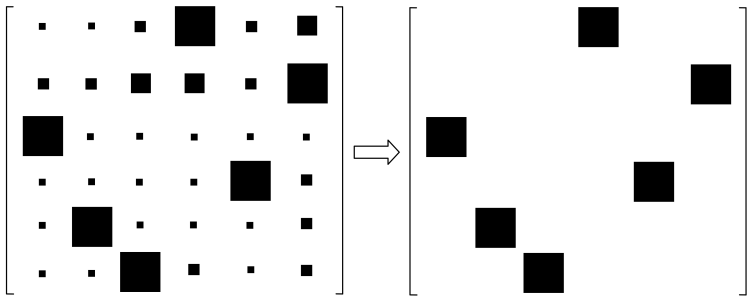

Figure 5: Application of the CSE procedure on the matrix S discovered by the algorithm O on the dataset k1b. The size of each square represents the order of magnitude of each value. The obtained structure is almost ideal, that is each document cluster is strongly correlated to a single term cluster.

- some relevant correlations are lost. This mainly occurs when some particular term (resp. document) clusters are correlated to almost all document (resp. term) clusters, since the CSE process preserves only the strongest correlations. The phenomenon that can be observed in Figure 6 (see red squares in the upper matrix) suggests that the number of specified term clusters is not appropriated and should be reduced.

- some term (resp. document) clusters appear to be highly redundant, i.e., they are correlated to the same document (resp. term) clusters (see blue squares, with different shades, in the lower matrix of Figure 6). This phenomenon suggests that both the number of specified term clusters and the number of specified document clusters should be reduced. Moreover, this also reveals that the categorization is probably too much fine-grained and that some categories actually represent the same (or highly related) concepts.

This situation is also reflected on the results obtained in terms of clustering accuracy (Table 4). Indeed, in this case the CSE process led to a decrease of the performance, possibly due to the lost of some (possibly relevant) correlations.

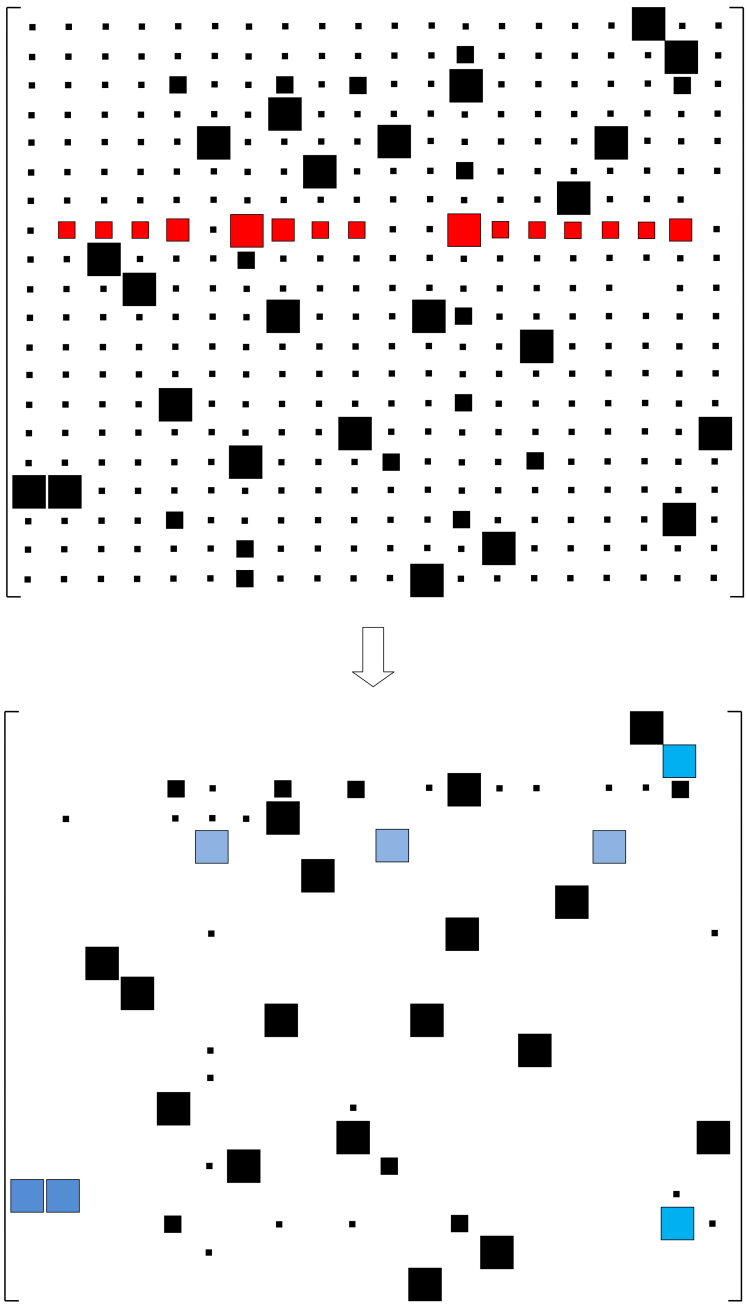

Figure 6: Application of the CSE procedure on the matrix S discovered by the algorithm O on the dataset k1a. The size of each square represents the order of magnitude of each value. The obtained structure emphasizes the presence of redundant term and document clusters.

5. Conclusions and future works

In this work we briefly reviewed the generally adopted approach to perform clustering tasks using Non-negative Matrix Factorizations, in both 2 and 3 factors variants. We introduced the possibility to discover a clean correlation structure between document and term clusters and to use such structure as a starting point for the same (or another) iterative algorithm.

Experiments conducted on many datasets, of different size and classified in a variable number of categories, prove that the application of the proposed strategy is able to increase the quality of the extracted clusters, especially in terms of average inter-cluster similarity.

Future works could be conducted to identify the most appropriate algorithm to show the correlation structure and to better understand on which type of datasets the proposed strategy produces better results. Further improvements could be obtained by the application of fuzzy approaches for the selection of the values of the same order of magnitude in the CSE process.

Moreover, additional studies could be performed to understand whether there is the possibility to automatically exploit the correlation structure extracted by the CSE process to refine the number of clusters.

Acknowledgements

We would like to acknowledge the support of the European Commission through the project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944).

References

[1] Agrawal, R., Gehrke, J., Gunopulus, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM/SIGMOD International Conference on Management of Data* (pp. 94–105).

[2] Berry, M., Browne, M., Langville, A., Pauca, P., & Plemmons, R. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, *52*, 155–173.

[3] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*, 993–1022.

[4] Casalino, G., Del Buono, N., & Mencar, C. (2011). Subtractive initialization of nonnegative matrix factorizations for document clustering. In *Proceedings of the 9th International Conference on Fuzzy Logic and Applications* WILF'11 (pp. 188–195). Berlin, Heidelberg: Springer-Verlag.

[5] Casalino, G., Del Buono, N., & Mencar, C. (2014). Subtractive clustering for seeding non-negative matrix factorizations. *Information Sciences*, *257*, 369–387.

[6] Chen, Y., Wang, L., & Dong, M. (2010). Non-negative matrix factorization for semisupervised heterogeneous data coclustering. *Knowledge and Data Engineering, IEEE Transactions on*, *22*, 1459–1474.

[7] Cho, H., & Dhillon, I. S. (2008). Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, *5*, 385–400.

[8] Chu, M., Del Buono, N., Lopez, L., & Politi, T. (2003). On the low-rank approximation of data on the unit sphere. *SIAM J. Matrix Anal. Appl*, *27*, 46–60.

[9] Chu, M., & Plemmons, R. J. (2005). Nonnegative matrix factorization and applications. *IMAGE, Bulletin of the International Linear Algebra Society*, *34*, 2–7.

[10] Cichocki, A., Lee, H., Kim, Y.-D., & Choi, S. (2008). Non-negative matrix factorization with alpha-divergence. *Pattern Recognition Letters*, *29*, 1433–1440.

[11] Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S. (2009). *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multiway Data Analysis and Blind Source Separation*. Wiley.

[12] Cook, W. L., Kashy, D. A., & Kenny, D. A. (2006). *Dyadic Data Analysis. Methodology in the Social Sciences*. Guilford Publications.

[13] Dhillon, I., & Sra, S. (2005). Generalized nonnegative matrix approximations with Bregman divergences. In *Proceeding of Neural Information Processing Systems*.

[14] Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *In Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 269–274).

[15] Ding, C., He, X., & Simon, H. D. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of SIAM Data Mining Conference*.

[16] Ding, C., Li, T., Peng, W., & Park, H. (2006). 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). ACM.

[17] Ding, C. H. Q., Li, T., & Jordan, M. I. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.*, *32*, 45–55.

[18] Donavall, A., Rege, M., Liu, X., & Jafari-Khouzani, K. (2012). Low-rank matrix factorization and co-clustering algorithms for analyzing large data sets. In *Proceedings of the Second international conference on Data Engineering and Management* ICDEM'10 (pp. 272–279). Berlin, Heidelberg: Springer-Verlag.

[19] George, T. (2005). A scalable collaborative filtering framework based on co-clustering. In *Fifth IEEE International Conference on Data Mining* (pp. 625–628).

[20] Gillis, N. (2014). The why and how of nonnegative matrix factorization. In *Regularization, Optimization, Kernels, and Support Vector Machines* Machine Learning and Pattern Recognition Series. Chapman and Hall/CRC.

[21] Gu, Q., & Zhou, J. (2009). Co-clustering on manifolds. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '09 (pp. 359–368). New York, NY, USA: ACM.

[22] Han, E.-H., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., & Moore, J. (1998). WebACE: a Web agent

for document categorization and exploration. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents* (pp. 408–415). New York, NY, USA: ACM.

[23] Hofmann, T., Puzicha, J., & Jordan, M. I. (2010). Learning from Dyadic Data. *Advances in Neural Information Processing Systems*, *11*.

[24] Kim, H., & Park, H. (2005). Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, *21*, 3970–3975.

[25] Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.

[26] Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Proceedings of the Advances in Neural Information Processing Systems Conference* (pp. 556–562). MIT Press volume 13.

[27] Li, T., Sindhwani, V., Ding, C., & Zhang, Y. (2010). Bridging domains with words: Opinion analysis with matrix tri-factorizations. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA* (pp. 293–302). SIAM.

[28] Long, B., Zhang, Z., & Yu, P. S. (2005). Co-clustering by Block Value Decomposition. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD)*.

[29] Lovasz, L., & Plummer, M. (1986). *Matching Theory*. Akadémiai Kiadó, North Holland, Budapest.

[30] Ma, H., Zhao, W., Tan, Q., & Shi, Z. (2010). Orthogonal nonnegative matrix tri-factorization for semi-supervised document co-clustering. In *Proceedings of the 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part II* PAKDD'10 (pp. 189–200). Berlin, Heidelberg: Springer-Verlag.

[31] Madeira, S. C., & Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *1*.

[32] Pio, G., Ceci, M., D'Elia, D., Loglisci, C., & Malerba, D. (2013). A Novel Biclustering Algorithm for the Discovery of Meaningful Biological Correlations between microRNAs and their Target Genes. *BMC Bioinformatics*, *14(S-7)*, S8.

[33] Pio, G., Ceci, M., D'Elia, D., & Malerba, D. (2014). Integrating microRNA target predictions for the discovery of gene regulatory networks: a semi-supervised ensemble learning approach. *BMC Bioinformatics*, *15(S-1)*, S4.

[34] Pio, G., Ceci, M., Loglisci, C., D'Elia, D., & Malerba, D. (2012). Hierarchical and Overlapping Co-Clustering of mRNA:miRNA Interactions. In L. D. Raedt, C. Bessière, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, & P. J. F. Lucas (Eds.), *ECAI* (pp. 654–659). IOS Press volume 242 of *Frontiers in Artificial Intelligence and Applications*.

[35] Pompili, F., Gillis, N., Absil, P.-A., & Glineur, F. (2014). Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing*, *141*, 15 –25.

[36] Porter, M. F. (1997). Readings in information retrieval. chapter An algorithm for suffix stripping. (pp. 313–316). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

[37] Salunke, A., Liu, X., & Rege, M. (2012). Constrained co-clustering with non-negative matrix factorisation. *IJBIDM*, (pp. 60–79).

[38] Shastri, B. J., & Levine, M. D. (2007). Face recognition using localized features based on non-negative sparse coding. *Machine Vision and Applications*, *18*, 107–122.

[39] Soukup, D., & Bajla, I. (2008). Robust object recognition under partial occlusions using NMF. *Computational Intelligence and Neuroscience*, *vol. 2008*, 14 pages. Article ID 857453.

[40] Tibshirani, R., Hastie, T., Eisen, M., Ross, D., Botstein, D., & Brown, P. (1999). *Clustering methods for the analysis of DNA microarray data*. Technical Report Stanford University.

[41] Wang, H., Nie, F., Huang, H., & Makedon, F. (2011). Fast nonnegative matrix tri-factorization for large-scale data co-clustering. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two* IJCAI'11 (pp. 1553–1558). AAAI Press.

[42] Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 267–273). ACM.

[43] Yang, J., Wang, W., Wang, H., & Yu, P. (2002). Co-clusters: Capturing subspace correlation in a large data set. In *Proceedings of the 18th IEEE International Conference on Data Engineering* (pp. 517–528).

[44] Yoo, J., & Choi, S. (2010). Orthogonal nonnegative matrix tri-factorization for co-clustering: multiplicative updates on Stiefel manifolds. *Information Processing and Management*, *46*, 559–570.

[45] Zeimpekis, D., & Gallopoulos, E. (2006). Tmg: A matlab toolbox for generating term-document matrices from text collections. In J. Kogan, C. Nicholas, & M. Teboulle (Eds.), *Grouping Multidimensional Data* (pp. 187–210). Springer Berlin Heidelberg.