Deep Symmetric Matrix Factorization

Pierre De Handschutter[1] Nicolas Gillis[1] Wivine Blekic[2]
[1] Department of Mathematics and Operational Research, University of Mons, Belgium
[2] Department of Psychiatry, NYU Grossman School of Medicine, New York, NY, USA

***Abstract*—Deep matrix factorizations (deep MFs) are recent extensions of standard MFs to several layers. This allows one to extract hierarchical interleaved features in high-dimensional datasets. In this paper, we present a variant of deep MF where the input matrix is symmetric and nonnegative, dubbed deep symmetric nonnnegative matrix factorization (DSNMF). We compare several loss functions to tackle DSNMF and propose different possible initialization techniques. We apply successfully DSNMF to the extraction of several levels of communities, both on synthetic data and on a psychiatric network, a promising application in the medical field.**

I. Introduction

Matrix factorizations (MFs) is a set of well-known unsupervised learning techniques where a data matrix $X \in \mathbb{R}^{m\times n}$ is approximated by the product of two smaller matrices, $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{r\times n}$, such that $X \approx WH$. To make the approximation meaningful, various constraints may be assumed on W and H including nonnegativity [1] and sparsity [2]. Among these variants, symmetric nonnegative matrix factorization (symNMF) [3], [4] requires the data matrix X to be nonnegative and symmetric, that is, $X = X^T$ with $m = n$, and $H = W^T$. This occurs when the entries of X measure the similarities between different items, for example a word co-occurrence matrix in topic modeling [5], or an adjacency matrix of an undirected graph. In this context, symNMF extracts communities of nodes, possibly overlapping, such that the nodes of a given community have more connections (that is, edges) with each other than with nodes belonging to other communities.

Recently, MFs started to scale up by considering several layers in the decomposition, following the path of deep learning. Especially, two milestone frameworks were successively introduced in the literature, namely multilayer MFs [6] and deep MFs [7]. More precisely, L layers of successive factorizations of ranks r_l ($l = 1, ..., L$) are performed on X as follows:

$$\begin{aligned} X &\approx W_1 H_1, \\ W_1 &\approx W_2 H_2, \\ &\dots \\ W_{L-1} &\approx W_L H_L, \end{aligned}$$

where $W_l \in \mathbb{R}^{m\times r_l}$ and $H_l \in \mathbb{R}_+^{r_l \times r_{l-1}}$ ($l = 1, \dots, L$) with $r_0 = n$, so that the matrix X is approximated as $X \approx W_L H_L H_{L-1} \cdots H_1$. In deep MF, the ranks are assumed to be decreasing, that is, $r_1 > r_2 > \cdots > r_L$; see [8], [9] for details.

Both multilayer MFs and deep MFs decompose the input matrix through several layers, but they differ on the way the factors are optimized. These "in-depth" factorizations were leveraged in several applications, such as hyperspectral unmixing [10], recommender systems [11] and multi-view clustering [12], and allow one to extract hierarchical features.

To the best of our knowledge, combining symmetry and depth within MFs has not yet been explored in the literature. In this paper, we explore such a MF, namely deep symmetric nonnegative matrix factorization (DSNMF).

Organization of the paper: In Section II, we describe DSNMF, along with the intuition behind it, and provide an illustration on a simple example. In Section III, we present the loss function for DSNMF and an efficient algorithm to solve it. We also discuss the initialization of DSNMF. Experiments on both synthetic and real data are then performed in Section IV, before concluding in Section V.

II. Proposed DSNMF model

The goal of DSNMF is to leverage L levels of factorizations to give at each layer l a nonnegative symmetric approximation of rank r_l of the original matrix $X \in \mathbb{R}^{n\times n}$. More precisely, at the first layer, X is approximated by $W_1 W_1^T$ where $W_1 \in \mathbb{R}_+^{n\times r_1}$, as in symNMF (see the introduction).

Let X be the symmetric adjacency matrix of a graph. In this case, each column of W_1 can be interpreted as a community, with $W_1(i,k)$ being the indicator of node i to belong to community k. In fact, $X \approx \sum_{k=1}^{r_1} W_1(:,k)W_1(:,k)^T$ means that X is approximated as the sum of r_1 communities which are rank-one nonnegative adjacency matrices. At the second layer, the matrix W_1 is factorized as $W_1 \approx W_2 H_2$ with $W_2 \in \mathbb{R}_+^{n\times r_2}$ and $H_2 \in \mathbb{R}_+^{r_2\times r_1}$, and $r_2 < r_1$. This gives a new symmetric approximation of X, namely $X \approx W_2(H_2 H_2^T)W_2^T$. At the second layer, each column of W_2 indicates to which extent the n data points belong to one of the r_2 communities. The square inner matrix $H_2 H_2^T \in \mathbb{R}^{r_2\times r_2}$ indicates how strongly the r_2 communities interact with each other. In fact, the second layer of DSNMF is a particular case of symmetric nonnegative tri-factorization, namely $X \approx WSW^T$ where $S = H_2 H_2^T$ [13]. As the factorization unfolds, the r_l's columns of the matrices W_l's will identify fewer communities (as the ranks of the factorization are decreasing) and the inner square matrix $H_l \cdots H_2 H_2^T \cdots H_l^T$ indicates how the numerous small communities of the first layers are progressively merged in fewer larger communities at the last layers. Hence, DSNMF provides a deeper level of understanding of the input data matrix than its single-layer version, similarly to the other deep MF models [8].

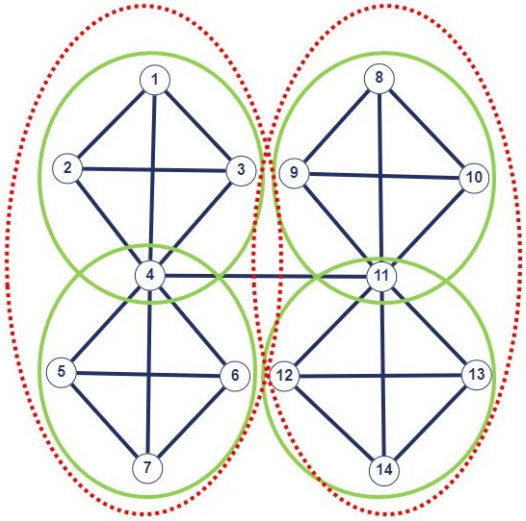

Fig. 1: Simple graph to illustrate the working of DSNMF, with two levels of communities.

Let us illustrate DSNMF with $L = 2$ on the synthetic graph of Fig. 1, made of 14 nodes. DSNMF applied with $r_1 = 4$ and $r_2 = 2$ splits the nodes in four communities at the first layer, namely containing the nodes $\{1, 2, 3, 4\}$, $\{4, 5, 6, 7\}$, $\{8, 9, 10, 11\}$, and $\{11, 12, 13, 14\}$. They correspond to the sets of nodes surrounded with a solid line circle. Note that nodes 4 and 11 belong with the same proportion to two communities. Then, at the second layer, only two communities remain (dashed circles), obtained by merging respectively the first two and the last two communities of the first layer.

III. Algorithm for DSNMF

Given a nonnegative symmetric matrix $X \in \mathbb{R}_+^{n \times n}$, standard symNMF consists in solving the following problem:

$$\min_{W \in \mathbb{R}_+^{n \times r}} \|X - WW^T\|_F^2.$$

To avoid dealing with a fourth-order objective function in W, an alternative formulation is proposed in [4]:

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

For μ sufficiently large, it has been shown that $W = H^T$ holds for the critical points of (1) [14]. Extending (1) to L layers requires to define an appropriate loss function. Inspired by [15] that showed that weighted sums of layer-wise contributions are meaningful loss functions for deep MFs, we propose to minimize

$$\mathcal{L}_{DSNMF} = \frac{1}{2}\big(\|X - W_1 H_1\|_F^2 + \mu_1 \|W_1 - H_1^T\|_F^2 + \lambda_1 \\ (\|W_1 - W_2 H_2\|_F^2 + \mu_2 \|W_2 - (H_2 H_1)^T\|_F^2) + \cdots + \lambda_{L-1} \\ (\|W_{L-1} - W_L H_L\|_F^2 + \mu_L \|W_L - (H_L H_{L-1} \cdots H_1)^T\|_F^2)\big). \tag{2}$$

This layer-centric loss function performs a weighted sum of the layer-wise symNMF errors, that is, $err(l) = \|W_{l-1} - W_l H_l\|_F^2 + \mu_l \|W_l - (H_l \cdots H_1)^T\|_F^2$ for $l = 1, \ldots, L$, with $W_0 = X$. Each layer-wise error is in turn the sum of two contributions. The first one, namely $err_1(l) = \|W_{l-1} - W_l H_l\|_F^2$, is the reconstruction error at layer l, that is, the error between W_{l-1} and its approximation $W_l H_l$ of rank r_l. The second term, namely $err_2(l) = \mu_l \|W_l - (H_l \cdots H_1)^T\|_F^2$ for all l, ensures the symmetry of the factorization, using the same trick as in Eq. (1) to solve non-quadratic optimization subproblems. With such a global loss function, it is possible to derive meaningful update rules for all the factors W_l's and H_l's, which are optimized alternatively, that lead to a monotonic decrease of the objective function.

To minimize $\mathcal{L}_{DSNMF}$ in Eq. (2), we use a block coordinate descent (BCD) method, with the blocks of variables W_l's and H_l's. This general framework is presented in Algorithm 1. The subproblems in one factor matrix at lines 4 and 5 are solved with a fast projected gradient method (FPGM) with Nesterov acceleration [16], similarly to what is described in [15].

Algorithm 1 DSNMF

Input: Symmetric matrix X.
Output: Matrices $W_1, \cdots, W_L$ and $H_1, \cdots, H_L$

1: Choose the number of layers L, the inner ranks $r_1, ..., r_L$, and the initial matrices $W_l^{(0)}$ and $H_l^{(0)}$ for all l.
2: **for** $k = 1, \ldots$ **do**
3: **for** $l = 1, \ldots, L$ **do**
4: $H_l^{(k)} = \arg\operatorname{reduce}_{H_l} \mathcal{L}_{\mathcal{DSNMF}}$
5: $W_l^{(k)} = \arg\operatorname{reduce}_{W_l} \mathcal{L}_{\mathcal{DSNMF}}$
6: **end for**
7: **end for**

A crucial aspect of deep MF models is the choice of the hyperparameters, namely the number of layers, L, and their factorization ranks, r_l's. In the following, we suggest two ways of initializing DSNMF:

- When the depth L and the ranks r_l's are given by the user, the initial factors $W_l^{(0)}$'s and $H_l^{(0)}$'s are initialized with a sequential multilayer approach, as in [17]. This is the strategy that we chose for experiments on synthetic data since it provides control on the network architecture.
- When no prior information on the network is provided, we resort to the well-known Louvain Method (LM) [18]. LM is a widely-used algorithm that extracts communities of nodes in a graph by maximizing the so-called network modularity. LM starts with each node representing its own community and then tries to move nodes from one community to another. After each iteration t, LM provides a split of the graph in r_t disjoint communities, that is, with each node belonging to a single community. In other words, LM extracts a bottom-up hierarchy of communities inside a graph, allowing a different interpretation at each iteration, similarly to what multilayer MF does (except that in multilayer MF, nodes can simultaneously belong to several communities, with some proportions). Hence, in the absence of values provided by the user, we

set the number of layers L of DSNMF to be equal to the number of iterations of LM and the ranks r_l's are taken as the number r_t's of communities successively extracted by LM. The initial matrices $W_l^{(0)}$'s are built for all l such that each column corresponds to a community extracted by LM at iteration l.

IV. Experiments

In this section, we apply DSNMF on synthetic data in Section IV-B, and on a real psychiatric network in Section IV-C. DSNMF can be interpreted as a hierarchical unsupervised fuzzy clustering approach hence no clear ground truth is available, which renders the quantitative assessment of the model challenging, even on synthetic data.

A. Compared methods

In the following experiments, we compare three algorithms, namely:

- DSNMF, see Algorithm 1 in Section II. To set up the parameters λ_l's for $l = 1, ..., L-1$ and μ_l's for $l = 1, ..., L$, we balance the importance of each layer and proceed as follows. The λ_l's are chosen such that the initial contributions of all layers are the same, that is, $\lambda_l = \frac{err^{(0)}(1)}{err^{(0)}(l+1)}$. Similarly, the μ_l's are such that for all l, $err_1^{(0)}(l) = err_2^{(0)}(l)$, that is $\mu_l = \frac{\|W_{l-1}^{(0)} - W_l^{(0)} H_l^{(0)}\|_F^2}{\|W_l^{(0)} - (H_l^{(0)} \cdots H_1^{(0)})^T\|_F^2}$.
- Multilayer symmetric NMF (MSNMF); this is the symmetric version of the sequential multilayer factorization of Cichocki et al. [6]. In other words, the symmetric factorizations are successively performed independently layer by layer.
- Spectral clustering (SpecClust) applied at each layer; this is a well-known clustering algorithm [19] which applies k-means to the n rows of the matrix whose columns are the first (that is, smallest) k eigenvectors of the graph Laplacian matrix. Hence, SpecClust generates disjoint communities, contrary to DSNMF and MSNMF.

All experiments were run in MATLAB, the code is available from https://gitlab.com/ngillis/deep-SymNMF/.

B. Synthetic dataset

We build our dataset in a similar way as the toy example of Fig. 1. More precisely, we set $L = 2$, $r_1 = 4$ and $r_2 = 2$. The noiseless graph consists in two disjoint sub-graphs of the same size, themselves composed of two cliques of the same size that have n^* nodes in common, which is the same for both sub-graphs. Hence, each of these $2n^*$ nodes belong equally to two communities at the first layer. For $n = 14$ and $n^* = 1$, this is exactly the situation represented on Fig. 1 if the edge between nodes 4 and 11 is removed. We add symmetric white Gaussian noise of standard deviation ϵ to the noiseless adjacency matrix $\tilde{X}$ such that the noisy data matrix is given by

$$X = \max\left(0, \tilde{X} + \epsilon ||\tilde{X}||_F \frac{N}{||N||_F}\right),$$

(n, n^*, ν)	DSNMF	MSNMF	SpecClust
$(14, 1, 0.01)$	$\mathbf{0.10} \pm 0.01$	0.11 ± 0.02	9.54
$(14, 1, 0.05)$	$\mathbf{0.53} \pm 0.08$	0.54 ± 0.08	9.54
$(14, 1, 0.1)$	$\mathbf{1.06} \pm 0.16$	1.07 ± 0.17	9.54
$(14, 1, 0.5)$	5.95 ± 0.71	$\mathbf{5.65} \pm 0.73$	9.54
$(100, 10, 0.01)$	$\mathbf{0.05} \pm 0.00$	$\mathbf{0.05} \pm 0.00$	11.17
$(100, 10, 0.05)$	0.27 ± 0.01	$\mathbf{0.23} \pm 0.01$	11.17
$(100, 10, 0.1)$	0.55 ± 0.02	$\mathbf{0.46} \pm 0.02$	11.17
$(100, 10, 0.5)$	3.78 ± 0.15	$\mathbf{2.52} \pm 0.12$	11.33 ± 0.16
$(100, 30, 0.01)$	0.07 ± 0.00	$\mathbf{0.06} \pm 0.00$	18.31
$(100, 30, 0.05)$	0.34 ± 0.01	$\mathbf{0.31} \pm 0.01$	18.31
$(100, 30, 0.1)$	0.72 ± 0.03	$\mathbf{0.62} \pm 0.03$	20.19 ± 5.09
$(100, 30, 0.5)$	6.37 ± 0.20	$\mathbf{3.40} \pm 0.19$	18.31

(a) MRSA at the first layer.

(n, n^*, ν)	DSNMF	MSNMF	SpecClust
$(14, 1, 0.01)$	2.39 ± 7.72	5.09 ± 7.26	$\mathbf{0}$
$(14, 1, 0.05)$	2.42 ± 4.90	5.94 ± 6.30	$\mathbf{0}$
$(14, 1, 0.1)$	5.41 ± 9.24	19.98 ± 11.84	$\mathbf{0}$
$(14, 1, 0.5)$	21.54 ± 12.47	32.05 ± 12.25	$\mathbf{14.49} \pm 17.04$
$(100, 10, 0.01)$	0.05 ± 0.00	8.02 ± 5.14	$\mathbf{0}$
$(100, 10, 0.05)$	0.29 ± 0.05	1.75 ± 1.18	$\mathbf{0}$
$(100, 10, 0.1)$	0.56 ± 0.08	1.64 ± 1.10	$\mathbf{0}$
$(100, 10, 0.5)$	3.85 ± 0.85	23.85 ± 7.47	$\mathbf{0}$
$(100, 30, 0.01)$	0.04 ± 0.00	19.78 ± 2.45	$\mathbf{0}$
$(100, 30, 0.05)$	0.19 ± 0.01	20.72 ± 2.40	$\mathbf{0}$
$(100, 30, 0.1)$	0.39 ± 0.02	20.88 ± 2.00	$\mathbf{0}$
$(100, 30, 0.5)$	2.10 ± 0.16	21.64 ± 2.02	$\mathbf{0}$

(b) MRSA at the second layer.

Table I: Comparison of the MRSA (average and standard deviation) of DSNMF, MSNMF and SpecClust on synthetic data over 25 runs in function of the noise level ϵ and the configuration of the network for (a) $r_1 = 4$ and (b) $r_2 = 2$. The best average MRSA achieved for each configuration is highlighted in bold.

where N is a symmetric square matrix whose elements are drawn from the standard normal distribution.

For different noise levels ϵ and for several combinations of n and n^*, we run the three methods described in Section IV-A with 25 different randomly generated noise matrices N as described above. Table I reports the average and standard deviation of the mean removed spectral angle (MRSA) between the columns of the ground truth and of the computed factors W_l's (which were permuted to minimize the MRSA), over these 25 runs. The MRSA between two vectors x and y is given by $\text{MRSA}(x, y) = \frac{100}{\pi} \arccos\left(\frac{\langle x - \overline{x}, y - \overline{y} \rangle}{\|x - \overline{x}\|_2 \|y - \overline{y}\|_2}\right) \in [0, 100]$ where $\langle \cdot, \cdot \rangle$ indicates the scalar product of two vectors and $\overline{\cdot}$ is the mean of a vector.

The interpretation of the results is not easy since no method clearly outperforms the others. At the first layer, DSNMF and MSNMF perform comparably, and outperform spectral clustering, especially in more challenging settings, when the clusters are more overlapping (that is, when n^* is larger). It was expected that MSNMF performs well on the first layer since it optimizes the first layer independently of the next

ones. It is reassuring to see that DSNMF performs comparably: although it has to take into account the decomposition at the second layer, the error at the first layer is similar to that of MSNMF. On the other hand, spectral clustering only extracts disjoint communities, which is a limitation since the communities at the first layer are overlapping.

At the second layer, MSNMF completely fails, showing the limitations of a purely sequential approach. This behaviour is likely to worsen when the number of layers increases. Spectral clustering nearly performs a perfect clustering at the second layer, which is expected since the two corresponding communities are disjoint (in the noiseless case). It is important to keep in mind that SpecClust always works on the original data without the need to keep a balance between several layers, such as DSNMF. Since it is by definition a single-layer method, SpecClust is not able to interpret the links between successive levels of communities, as opposed to DSNMF.

In summary, DSNMF is the only method able to balance both layers for the various tested configurations.

C. Psychiatric networks

Network analysis has been recently applied successfully in computational psychiatry. The original data matrix $Y \in \mathbb{R}^{m \times n}$ contains the ratings of m subjects on n symptoms on an ordinal scale. Given Y, the symmetric input matrix $X \in \mathbb{R}^{n \times n}$ is made of the partial correlations between the n symptoms, that is, $X(i,j) = -\frac{K(i,j)}{\sqrt{K(i,i)K(j,j)}}$ where $K = \Sigma^{-1}$ is the precision matrix, defined as the inverse of the covariance matrix of the columns of Y [20].

Considering the graph whose adjacency matrix is X and where each node corresponds to a symptom, it is interesting to identify its communities. This would allow us, for example, to evaluate the set of symptoms that will be somehow impacted by an action (such as a medication) on a particular symptom [21]. Most works in this recent field only extract one level of disjoint communities, which does not allow to finely grasp all the possible interactions in the network.

In the following, we consider a dataset of 359 women suffering from post-traumatical stress disorder (PTSD) evaluated through the PTSD Symptom Scale-Self Report (PSS-SR) [22]. The PSS-SR is an ordinal scale assessing the 17 symptoms of PTSD. It is built on the fourth version of the Diagnostic and Statistical Manual of Mental Disorders (DSM) [23] (DSM-IV), a standard classification of mental disorders used by mental health professionals. Note that the fifth version is denoted DSM-5; see below for some details. Items correspond to the frequency of some behaviours considered as characteristic of the pathology. This scale was expected to have three communities representing symptoms of arousal (for example, *being jumpy*), avoidance (for example, *avoid reminders of the trauma*) and re-experiencing (for example, *having bad dreams about the trauma*).

The network of symptoms is built in R with the so-called EBIC graphical Lasso regularization (see [21]) and represented on Fig. 2.

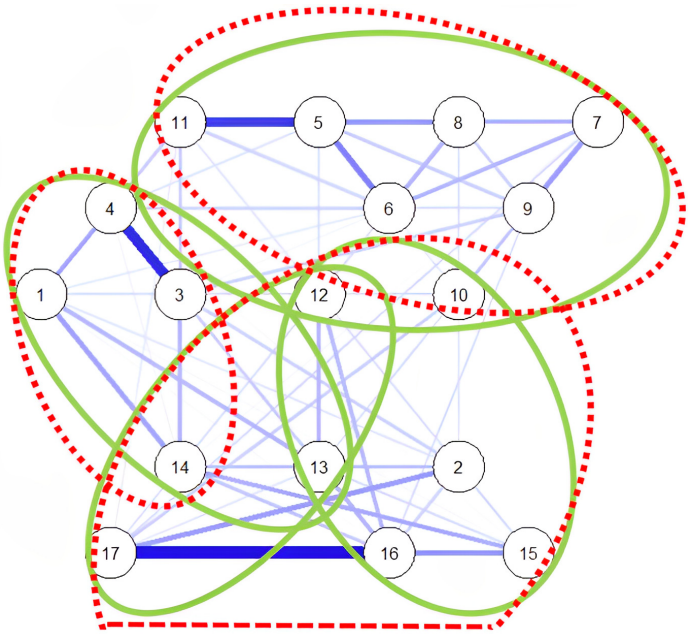

Fig. 2: Communities extracted at each layer by DSNMF on a PTSD dataset. Communities extracted at the first and second layer are circled in green and red, respectively.

We apply DSNMF on this network with the LM initialization. LM extracts 2 layers of respectively 4 and 3 communities hence we perform DSNMF with $L = 2$, $r_1 = 4$, $r_2 = 3$. Fig. 2 displays the extracted communities at the first and second layers which are circled in green and red, respectively. For convenience, we only plot the main communities to which each node belongs to. For a given node, we first assign it to the community for which it has the largest degree of membership (that is, largest value in the corresponding factor W_l). Then, we sequentially assign it to more communities as follows: we assign it to the next community with the largest degree of membership if this degree is at least 60% of the one of the latest community assigned to this node. At the first layer, the 4 extracted communities are

- $\{1, 3, 4, 13, 14\}$, which represents symptoms of avoidance and physical reaction caused by the lack of avoidance of trauma reminders,
- $\{5, 6, 7, 8, 9, 10, 11, 12\}$, which is clinically the most homogeneous community. These symptoms represent a negative mood and correspond to the community added in the new version of the scale (that is, DSM-5), and
- $\{2, 10, 12, 13, 15, 16\}$ and $\{12, 13, 14, 17\}$ gather re-experiencing symptoms and attempts to avoid such re-experiences.

Let us remark that node 10 and 14 belong to two communities and nodes 12 and 13 belong to three communities.

At the second layer, the 3 extracted communities are $\{1, 3, 4, 14\}$, $\{5, 6, 7, 8, 9, 10, 11\}$ and $\{2, 10, 12, 13, 14, 15, 16, 17\}$. Node 10 and 14 again belong to two communities. Roughly speaking, the second layer merges the two last communities of the first layer, keeping the two others mostly unchanged.

The analysis of the different communities shows that the algorithm did not extract the different sub-scales of the PSS-SR, but rather extracted communities representing behaviours that are commonly presented together among patients. In other words, DSNMF did not extract the different symptoms of avoidance, re-experiencing and arousal distinctly, but rather extracted behaviors that can be observed together. This could have strong clinical implication for disorders presenting complex interactions between multiple features, such as PTSD or suicide behaviors.

Finally, it is to be noted that the communities extracted by DSNMF represent well the criticisms raised by the DSM-IV assessment of PTSD. The joint presentation of symptoms representing a *negative mood and cognition* led the research community to modify the assessment of PTSD in the DSM-5 [24]. Indeed, it was shown that (i) certain items represented a different category of symptoms that the one originally conceptualized, and (ii) the formulation of certain items led to inconsistencies [25]. The analysis of the node shared by the most communities strengthens this interpretation. That is, node 12 has been largely questioned, re-written in the DSM-5 and finally added to the new community of symptoms of negative mood and cognition.

V. Conclusion

In this paper, we introduced deep symmetric nonnegative matrix factorization (DSNMF), an extension of symmetric NMF to several layers. We showed the efficiency of the proposed model on both synthetic and real data for the hierarchical extraction of interleaved communities in networks. In particular, we extracted non-disjoint communities in a network of psychiatric symptoms, that lead to meaningful clinical interpretations. We plan to investigate more in details the added value of such hierarchical communities extraction in the psychological field in future works.

Interesting perspectives also include the experimentation of other initialization strategies for real data, especially when the number of layers of factorization is unknown. In this paper, we used the well-known Louvain Method which extracts a hierarchy of disjoint communities but has the drawback to assign each node to a single community at each step. Testing our model on other applications, such as the hierarchical extraction of topics in a document corpus, is an other promising perspective, but without ground truth, the quantitative analysis of the performance of DSNMF would be challenging, as for the other unsupervised deep MF models. Hence, defining proper metrics to assess the quality of such a hierarchical fuzzy clustering also seems crucial to us.

Acknowledgement

This work was supported by the Fonds de la Recherche Scientifique - FNRS (F.R.S.-FNRS) and the Fonds Wetenschappelijk Onderzoek - Vlaanderen (FWO) under EOS Project no O005318F-RG47, by the Francqui Foundation, and by the European Research Council (ERC consolidator grant no 101085607).

References

[1] D.D. Lee and H.S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[2] R. Gribonval and K. Schnass, "Dictionary identification—sparse matrix-factorization via ℓ_1-minimization," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3523–3539, 2010.

[3] C. Ding, X. He, and H.D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the 2005 SIAM Int. Conf. Data Min.* SIAM, 2005, pp. 606–610.

[4] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proceedings of the 2012 SIAM Int. Conf. Data Min.* SIAM, 2012, pp. 106–117.

[5] K. Huang, X. Fu, and N. D. Sidiropoulos, "Anchor-free correlated topic modeling: Identifiability and algorithm," *Adv Neural Inf Process Syst*, vol. 29, 2016.

[6] A. Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorisation," *Electronics Letters*, vol. 42, no. 16, pp. 947–948, 2006.

[7] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller, "A deep matrix factorization method for learning attribute representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 417–429, 2016.

[8] P. De Handschutter, N. Gillis, and X. Siebert, "A survey on deep matrix factorizations," *Computer Science Review*, vol. 42, pp. 100423, 2021.

[9] W.-S. Chen, Q. Zeng, and B. Pan, "A survey of deep nonnegative matrix factorization," *Neurocomputing*, vol. 491, pp. 305–320, 2022.

[10] L. Tong, J. Yu, C. Xiao, and B. Qian, "Hyperspectral unmixing via deep matrix factorization," *Int. J. Wavelets Multiresolution Inf. Process.*, vol. 15, no. 06, pp. 1750058, 2017.

[11] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *IJCAI*, 2017, pp. 3203–3209.

[12] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 2921–2927.

[13] H. Wang, H. Huang, and C. Ding, "Simultaneous clustering of multi-type relational data via symmetric nonnegative matrix tri-factorization," in *Proceedings of the 20th ACM Int. Conf. Inf. Knowl. Manag.*, 2011, pp. 279–284.

[14] X. Li, Z. Zhu, Q. Li, and K. Liu, "A provable splitting approach for symmetric nonnegative matrix factorization," *IEEE Trans. Knowl. Data Eng.*, 2021, 10.1109/TKDE.2021.3125947.

[15] P. De Handschutter and N. Gillis, "A consistent and flexible framework for deep matrix factorizations," *Pattern Recognition*, vol. 134, pp. 109102, 2023.

[16] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate O(1/k^2)," in *Dokl. akad. nauk Sssr*, 1983, vol. 269, pp. 543–547.

[17] A. Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorization using projected gradient approaches," *Int. J. Neural Syst.*, vol. 17, no. 06, pp. 431–446, 2007.

[18] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, pp. P10008, 2008.

[19] Ulrike Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

[20] S. Epskamp and E. I. Fried, "A tutorial on regularized partial correlation networks.," *Psychological methods*, vol. 23, no. 4, pp. 617, 2018.

[21] D. Hevey, "Network analysis: a brief overview and tutorial," *Health Psychol. Behav. Med.*, vol. 6, no. 1, pp. 301–328, 2018.

[22] E. B. Foa, D. S. Riggs, C. V. Dancu, and B. O. Rothbaum, "Reliability and validity of a brief instrument for assessing post-traumatic stress disorder," *Journal of Traumatic Stress*, vol. 6, no. 4, pp. 459–473, 1993.

[23] American Psychiatric Association, *Diagnostic and Statistical Manual of Mental Disorders: DSM-IV-TR (4th ed., text rev.)*, vol. 4, American Psychiatric Association Washington, DC, 2000.

[24] American Psychiatric Association, *Diagnostic and Statistical Manual of Mental Disorders (5th ed.)*, American Psychiatric Association Washington, DC, 2013.

[25] A. Pai, A. M. Suris, and C. S. North, "Posttraumatic stress disorder in the DSM-5: Controversy, change, and conceptual considerations," *Behavioral sciences*, vol. 7, no. 1, pp. 7, 2017.