NEURAL NONNEGATIVE MATRIX FACTORIZATION FOR HIERARCHICAL MULTILAYER TOPIC MODELING

M. Gao[1], *J. Haddock*[2], *D. Molitor*[2], *D. Needell*[2], *E. Sadovnik*[2], *T. Will*[3], *R. Zhang*[4]

Univ. of California, Irvine[1], Univ. of California, Los Angeles[2], Michigan State Univ.[3], Peking Univ.[4]

ABSTRACT

We introduce a method for detecting latent hierarchical structure in data based on nonnegative matrix factorization. Datasets with hierarchical structure arise in a wide variety of fields, such as document classification, image processing, and bioinformatics. The proposed method, Neural NMF, recursively applies topic modeling in layers to discover overarching topics encompassing the lower-level features. We derive a backpropagation scheme that allows us to frame our method as a neural network. Numerical results on a synthetic dataset demonstrate that Neural NMF outperforms similar algorithms on a hierarchical classification task.

Index Terms— nonnegative matrix factorization, hierarchical topic modeling, backpropagation

1. INTRODUCTION

Topic modeling is a useful technique for revealing latent themes in a dataset. Topic modeling methods cluster and classify data observations in an unsupervised manner or make use of semisupervision, in which class labels are available for a subset of data points. Algorithms for topic modeling most often find application in the domain of document classification [1], but more recently have found use in image classification [2] and bioinformatics [3]. Feature extraction is an approach related to, but distinct from, topic modeling. Where topic modeling seeks to identify hidden topics and represent data points by these topics, feature extraction aims to find a few features that best represent the set for the task at hand (e.g., classification) [4].

Nonnegative matrix factorization (NMF) is a popular method in machine learning because it is able to both extract features and generate topic models [5, 6]. However, NMF does not inherently discover hierarchical structure in its topics. Borrowing techniques from neural networks, we seek to modify classical nonnegative matrix factorization to successfully handle hierarchy.

DN, JH, ES, and DM are grateful to and were partially supported by NSF CAREER DMS #1348721 and NSF BIGDATA DMS #1740325. This work is based upon work completed at the UCLA CAM REU during Summer 2018 that was funded by NSF DMS #1659676.

1.1. Notation

We distinguish matrices and vectors from scalar quantities using bold font. For a matrix $\mathbf{F}$, $\mathbf{F}_{i,:}$ and $\mathbf{F}_{:,j}$ denote row i and column j, respectively. For sets of indices T and S, $\mathbf{F}_{T,:}$ and $\mathbf{F}_{:,S}$ denote the matrix obtained by deleting the rows of $\mathbf{F}$ not in T or the columns of $\mathbf{F}$ not in S, respectively. By extension, $\mathbf{v}_T$ is the vector $\mathbf{v}$ restricted to the entries with indices in T. We denote the Moore-Penrose pseudoinverse of $\mathbf{F}$ as $\mathbf{F}^\dagger$. Entrywise (Hadamard) multiplication and division between $\mathbf{F}$ and $\mathbf{G}$ are denoted by $\mathbf{F}\odot\mathbf{G}$ and $\frac{\mathbf{F}}{\mathbf{G}}$, respectively. The all ones vector of length k is denoted $\mathbf{1}_k$. All norms denote the Frobenius norm. We perform subscript (indicial) operations before superscript (pseudoinversion, transposition) operations when applicable. The interval $[0,\infty)^k$ is denoted $\mathbb{R}_+^k$. In methods with $\mathcal{L}$ layers, we use $\mathbf{F}^{(\ell)}$ to denote the matrix $\mathbf{F}$ at layer ℓ. We similarly use $k^{(\ell)}$ as the number of topics at layer ℓ. In any supervised setting, we use P as the number of classes.

1.2. NMF

For a given data matrix $\mathbf{X}\in\mathbb{R}_+^{N\times M}$ and hidden topic number k chosen as a hyperparameter, nonnegative matrix factorization (NMF) seeks $\mathbf{A}\in\mathbb{R}_+^{N\times k}$ and $\mathbf{S}\in\mathbb{R}_+^{k\times M}$ such that $\mathbf{X}\approx\mathbf{AS}$. This approximate factorization is meant to find few latent feature vectors which combine to approximately represent the data points in X; these vectors are called *topics*. To find $\mathbf{A}$ and $\mathbf{S}$, we wish to solve the minimization problem

$$\min_{\mathbf{A}\geq 0,\mathbf{S}\geq 0}\|\mathbf{X}-\mathbf{AS}\|^2. \tag{1}$$

The nonnegativity restriction on $\mathbf{X}$, $\mathbf{A}$, and $\mathbf{S}$ differentiates NMF from other topic modeling and feature extraction techniques, including principal component analysis (PCA) and autoencoding. This allows for a natural and intuitive 'parts-based' representation [7]; there are no topics that contribute in a negative manner. Since all values are nonnegative, only additive combinations of topic representatives are allowed to produce data points in $\mathbf{X}$. We can view the columns of the $\mathbf{A}$ matrix as vectors of important features of $\mathbf{X}$, or from a topic modeling perspective, as k hidden themes in our data. In this view, the $\mathbf{S}$ matrix provides the coefficients to represent each data point of $\mathbf{X}$ as a linear combination of the topics. Moreover, we can view the product $\mathbf{AS}$ as a low-rank approximation of $\mathbf{X}$, since $\operatorname{rank}(\mathbf{AS}) \leq \min(\operatorname{rank}(\mathbf{A}), \operatorname{rank}(\mathbf{S})) \leq k$.

The problem (1) is convex in $\mathbf{A}$ and $\mathbf{S}$ separately, but nonconvex in both. We therefore cannot expect to consistently find a global minimum. Several techniques exist for finding local minima. Multiplicative update schemes, in particular, are quick and extensible and alternate between fixing $\mathbf{A}$ and $\mathbf{S}$ and updating the other to iteratively decrease (1).

1.3. Semisupervised NMF (SSNMF)

A natural extension of NMF is to take advantage of any known label information in the factorization [8]. Suppose $\mathbf{Y} \in \mathbb{R}^{P \times M}$ is a matrix containing label information for M objects in P classes and suppose some data may be missing. Let $\mathbf{W} \in \mathbb{R}^{N \times M}$ be the binary indicator matrix for the data, that is $\mathbf{W}_{n,m} = 1$ if $\mathbf{X}_{n,m}$ is known and 0 otherwise, and let $\mathbf{L} \in \mathbb{R}^{P \times M}$ be the label indicator matrix, with $\mathbf{L}_{:,m} = \mathbf{1}_P$ if the label for object m is known and 0 otherwise. We can incorporate label information and missing data by adjusting our problem as

$$\min_{\mathbf{A},\mathbf{S},\mathbf{B}\geq 0} \underbrace{\|\mathbf{W} \odot (\mathbf{X} - \mathbf{AS})\|_F^2}_{\text{Reconstruction Error}} + \lambda \underbrace{\|\mathbf{L} \odot (\mathbf{Y} - \mathbf{BS})\|_F^2}_{\text{Classification Error}}. \quad (2)$$

The resulting $\mathbf{B} \in \mathbb{R}^{P \times k}$ is a classification matrix, with $\mathbf{B}_{i,:}$ defining a hyperplane that attempts to separate objects in class i from objects not in class i. Because of the entrywise multiplication with $\mathbf{W}$ and $\mathbf{L}$, (2) restricts the loss to data and labels which are known. The relative importance of the classification error is controlled by the user-defined hyperparameter λ. We can iteratively decrease (2) by extending the multiplicative updates for NMF; details can be found in [8].

1.4. Hierarchical NMF (hNMF)

A further extension of both NMF and SSNMF illuminates hierarchical structure by recursively factorizing the $\mathbf{S}$ matrices. By performing NMF with $k = k^{(0)}$, we reveal $k^{(0)}$ hidden topics in the data; by repeating the factorization on the $\mathbf{S}$ matrix with $k = k^{(1)}$, we further collect the $k^{(0)}$ topics into $k^{(1)}$ supertopics. This process for $\mathcal{L}$ layers is to approximately factor the data matrix as

$$\begin{aligned} \mathbf{X} &\approx \mathbf{A}^{(0)}\mathbf{S}^{(0)}, \\ \mathbf{X} &\approx \mathbf{A}^{(0)}\mathbf{A}^{(1)}\mathbf{S}^{(1)}, \\ &\vdots \\ \mathbf{X} &\approx \mathbf{A}^{(0)}\mathbf{A}^{(1)}\cdots\mathbf{A}^{(\mathcal{L})}\mathbf{S}^{(\mathcal{L})}. \end{aligned} \quad (3)$$

The $\mathbf{A}^{(i)}$ matrix represents the how the subtopics at layer i collect into the supertopics at layer $i+1$.

Note that as $\mathcal{L}$ increases, $\|\mathbf{X} - \mathbf{A}^{(0)}\mathbf{A}^{(1)}\cdots\mathbf{A}^{(\mathcal{L})}\mathbf{S}^{(\mathcal{L})}\|$ necessarily increases as error propagates with each step. As a result, significant error is introduced when $\mathcal{L}$ is large. Choosing $k^{(0)}, k^{(1)}, ...k^{(\mathcal{L})}$ in practice proves difficult when the number of topics at each layer is unknown, as the number of possibilities grows combinatorially. Additionally, large differences between the number of topics for adjacent layers introduces large error into the factorization. Finally, as the NMF problem is ill-posed and has an infinite number of global minima (one may rescale each of the factors), this ill-posedness is exacerbated in hNMF. There are unicity results for NMF when the matrix $\mathbf{X}$ satisfies specific constraints (see e.g., [9, 10, 11, 12, 13]), but we do not know of such results for hNMF; this is an important direction.

1.5. Deep NMF (DNMF)

In [14], the authors make a first step toward bridging the gap between hNMF and neural networks. Their method achieves a considerable performance improvement over standard NMF in classification. The forward process for DNMF is hNMF with pooling operator, p, applied after each layer of decomposition to introduce nonlinearity and minimize overfitting. Without the pooling operation, the DNMF model is identical to hNMF. The other major contribution of [14] is a proposed backpropagation algorithm meant to refine the result obtained from the forward process. However, the backpropagation technique introduced in [14] differs from backpropagation techniques in neural network settings, as it only propagates one layer at a time and uses multiplicative updates instead of gradient descent to update the values of $\mathbf{A}$ and $\mathbf{S}$.

Similar ideas were explored in [15], [16], and [17]. In [15], the authors develop a hierarchical model in which some of the nonnegativity constraints are relaxed; however, this lacks our proposed backpropagation algorithm for training the model. In [16], the authors propose a NMF backpropagation algorithm using an "unfolding" approach; however, their method does not allow for hierarchy. Finally, a method similar to ours was developed in [17], but differs from ours in that it lacks the nonnegativity constraints that makes our method applicable in topic modeling and feature extraction.

2. PROPOSED METHOD: NEURAL NMF (NNMF)

One potentially problematic aspect of DNMF [14] is that their backpropagation is different than backpropagation typically used in neural networks. Traditional backpropagation determines the gradient of a cost function with respect to *all* the weights in the network, so that all the weights in the network may be updated at once. In [14], the update for $\mathbf{S}^{(\ell)}$ is only allowed to depend on $\mathbf{A}^{(\ell+1)}$ and $\mathbf{S}^{(\ell+1)}$. One of the barriers to formulating a proper backpropagation step for DNMF is that in optimization methods like multiplicative updates [18] or alternating least squares [19], the $\mathbf{A}$ and $\mathbf{S}$ matrices take turns acting as the independent and dependent variables in the updates. This is in contrast to the setting in neural networks, where the weights connecting neurons between layers are independent variables, while the activations of the neurons are dependent. This separation of independent and dependent variables allows one to calculate derivatives with respect to the network weights in a relatively simple way.

Thus, we now make a choice to regard the $\mathbf{A}$ matrices as the independent variables. This is natural since the $\mathbf{S}$ matrix is "passed on " to the next layer, analogous to the neurons' activations being passed to the next network layer. Since we have chosen to regard the $\mathbf{A}$ matrices as the independent variables, we need to determine the $\mathbf{S}$ matrices from the $\mathbf{A}$ matrices. The natural way to do this is to require the $\mathbf{S}$ matrices to solve (4). Suppose $\mathbf{A}^{(0)}, ..., \mathbf{A}^{(\mathcal{L})}$ are given and let $\mathbf{S}^{(-1)} := \mathbf{X}$. Then we let

$$\mathbf{S}^{(\ell)} = \underset{\mathbf{S}\geq 0}{\operatorname{argmin}} \|\mathbf{S}^{(\ell-1)} - \mathbf{A}^{(\ell)}\mathbf{S}\|, \quad \ell = 0, \cdots, \mathcal{L}. \tag{4}$$

These definitions require that, given any configuration of independent variables $\mathbf{A}^{(0)}, \ldots, \mathbf{A}^{(\mathcal{L})}$, the $\mathbf{S}$ matrices we compute give us the best possible NMF decompositions.

We define $q(\mathbf{A}, \mathbf{X})$, for any nonnegative matrices $\mathbf{X}$ and $\mathbf{A}$ with the same number of rows, by

$$q(\mathbf{A}, \mathbf{X}) := \underset{\mathbf{S}\geq 0}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{A}\mathbf{S}\|. \tag{5}$$

This problem is ill-posed if $\mathbf{A}$ does not have full column rank. As the $\mathbf{A}$ matrices are always tall and skinny, we make the reasonable assumption that every $\mathbf{A}$ matrix has full column rank. We can now rephrase the definitions of $\mathbf{S}$ matrices as

$$\mathbf{S}^{(\ell)} = q(\mathbf{A}^{(\ell)}, \mathbf{S}^{(\ell-1)}), \quad \ell = 0, 1, \cdots, \mathcal{L}. \tag{6}$$

These equations immediately show that $\mathbf{S}^{(\ell')}$ depends on $\mathbf{A}^{(\ell)}$ for $\ell \leq \ell'$, but not for $\ell > \ell'$. These equations form the forward-propagation stage of NNMF.

We now want to differentiate a cost function (corresponding to e.g., semisupervision), which depends on both the $\mathbf{A}$ and $\mathbf{S}$ matrices, with respect to the $\mathbf{A}$ matrices in order to perform backpropagation. This requires us to differentiate the q function and apply the chain rule. The derivative of $q(\mathbf{A}, \mathbf{X})$ can be computed columnwise in $\mathbf{X}$, so we need only determine derivative formulas for $q(\mathbf{A}, \mathbf{x})$. Our formulas make use of the fact that differentiation of $q(\mathbf{A}, \mathbf{x})$ requires only pseudoinversion of $\mathbf{A}_{:,\text{supp}(\mathbf{x})}$ where the support of $q(\mathbf{A}, \mathbf{x})$ is constant. Suppose all $\mathbf{S}^{(\ell)}$ are defined as in (6) and let $T_m^{(\ell)} := \operatorname{supp} \mathbf{S}^{(\ell)}_{:,m}$. We define $\mathbf{\Phi}^{(\ell,\ell'),m}$ for $\mathcal{L} \geq \ell' \geq \ell \geq 0$ by

$$\begin{aligned}\mathbf{\Phi}^{(\ell,\ell'),m} \quad := \quad & \left(\mathbf{A}^{(\ell')}_{:,T_m^{(\ell')}}{}^{\dagger}\right)_{:,T_m^{(\ell'-1)}} \left(\mathbf{A}^{(\ell'-1)}_{:,T_m^{(\ell'-1)}}{}^{\dagger}\right)_{:,T_m^{(\ell'-2)}} \\ & \cdots \left(\mathbf{A}^{(\ell+1)}_{:,T_m^{(\ell+1)}}{}^{\dagger}\right)_{:,T_m^{(\ell)}} \left(\mathbf{A}^{(\ell)}_{:,T_m^{(\ell)}}{}^{\dagger}\right).\end{aligned}$$

Suppose C is a cost function depending on all the variables $\mathbf{S}^{(\ell)}$ and $\mathbf{A}^{(\ell)}$. Now we define several matrices which allow us to state the chain rule applied to C with respect to $\mathbf{A}^{(i)}$ more succintly. We let

$$\mathbf{d}^{(\ell,\ell'),m} := \left(\mathbf{\Phi}^{(\ell,\ell'),m}\right)^{\top} \left(\frac{\partial C}{\partial \mathbf{S}^{(\ell')}}\right)_{T_m^{(\ell')},m}, \mathbf{U}^{(\ell,\ell'),m}_{:,T_m^{(\ell)c}} = 0,$$

and

$$\begin{aligned}\mathbf{U}^{(\ell,\ell'),m}_{:,T_m^{(\ell)}} := & -\mathbf{d}^{(\ell,\ell'),m} \left(\mathbf{S}^{(\ell)}_{T_m^{(\ell)},m}\right)^{\top} \\ & + \left(\mathbf{S}^{(\ell-1)} - \mathbf{A}^{(\ell)}\mathbf{S}^{(\ell)}\right)_{:,m} \left(\mathbf{d}^{(\ell,\ell',),m}\right)^{\top} \left(\mathbf{A}^{(\ell)}_{:,T_m^{(\ell)}}{}^{\dagger}\right)^{\top}.\end{aligned}$$

We define $\frac{\partial C}{\partial \mathbf{A}^{(\ell)}}\Big|_{\mathbf{S}}$ to be the derivative of C with respect to $\mathbf{A}^{(\ell)}$, holding the $\mathbf{S}$ matrices constant; the chain rule yields

$$\frac{\partial C}{\partial \mathbf{A}^{(\ell)}} = \frac{\partial C}{\partial \mathbf{A}^{(\ell)}}\bigg|_{\mathbf{S}} + \sum_{\substack{\ell \leq \ell' \leq \mathcal{L} \\ 1 \leq m \leq M}} \mathbf{U}^{(\ell,\ell'),m}.$$

This derivative allows us to perform the gradient descent step in Algorithm 1. We will present derivations and prove that this nearly always provides the correct derivative in [20].

Algorithm 1 Neural NMF

```
Require: data matrix $\mathbf{X} \in \mathbb{R}^{N\times M}$, number of layers $\mathcal{L}$, step
    size $\gamma$, cost function $C$, initial matrices $\mathbf{A}^{(i)}$ for $i = 0, ..., \mathcal{L}$
  procedure ForwardPropagation($\mathbf{A}^{(0)}...\mathbf{A}^{(\ell)}$)
      for $i := 0...\mathcal{L}$ do
          $\mathbf{S}^{(i)} \leftarrow q(\mathbf{A}^{(i)}, \mathbf{S}^{(i-1)})$
  ForwardPropagation($\mathbf{A}^{(0)}...\mathbf{A}^{(\ell)}$)
  while not converged do
      for $i := 0...\mathcal{L}$ do
          $\mathbf{A}^{(i)} \leftarrow \mathbf{A}^{(i)} - \gamma * \frac{\partial C}{\partial \mathbf{A}^{(i)}}$        ▷ Gradient descent
          $\mathbf{A}^{(i)} \leftarrow \mathbf{A}^{(i)}_{+}$        ▷ Project onto positive orthant
      ForwardPropagation($\mathbf{A}^{(0)}...\mathbf{A}^{(\ell)}$)
```

Note that the pseudocode above only provides updates for the factor matrices, $\mathbf{A}^{(i)}$. If one uses additional matrices in the objective function (such as $\mathbf{B}$ in (2)) they need only differentiate with respect to these matrices and update in the same way. We use this approach in our experiments in Section 3.

3. EXPERIMENTAL RESULTS

We test NNMF on a 87×90 noisy toy dataset with a three-layer hierarchical structure. Starting with two large blocks, we overlay increasingly smaller and more intense asymmetric regions (with values 1, 2, and 4) along the diagonal, and finally add a uniform$(0, 0.15)$ noise; see the left plot of Figure 1. We test hNMF, DNMF, and NNMF with one, two, or three layer structure, and various levels of supervision (results are averaged over 25 trials); the cost function to which we apply Algorithm 1 in these experiments is

$$\|\mathbf{X} - \mathbf{A}^{(0)} \cdots \mathbf{A}^{(\mathcal{L})}\mathbf{S}^{(\mathcal{L})}\|_F^2 + \|\mathbf{L} \odot (\mathbf{Y} - \mathbf{B}\mathbf{S}^{(\mathcal{L})})\|_F^2. \tag{7}$$

. The class labels associated to each data point indicate which of the nine highest intensity blocks contain the maximum entry; 40% (semisupervised) or 100% (supervised) are provided. We present the recovery error (computed as $\|\mathbf{X} - \mathbf{A}^{(0)}\mathbf{A}^{(1)} \cdots \mathbf{A}^{(\mathcal{L})}\mathbf{S}^{(\mathcal{L})}\|/\|\mathbf{X}\|$) and classification accuracy (proportion of predicted labels matching true labels) in Table 1. Object m is predicted to have label p if $(\mathbf{BS}^{(\mathcal{L})})_{pm} = \max\,(\mathbf{BS}^{(\mathcal{L})})_{:m}$. Noteworthy improvements of NNMF over hNMF and DNMF are bolded. We expect the advantage NNMF enjoys is due to the backpropagation avoiding suboptimal local minima of hNMF and DNMF.

In Figure 1, we visualize the reconstructions produced by each unsupervised method with two layer structure ($k^{(0)} = 9$ and $k^{(1)} = 4$). We cannot resolve all the highest-intensity features from the original data, as the NMF approximations have lower than necessary rank. Although each method captures some of the two-layer structure, NNMF outperforms hNMF and DNMF, resolving sharper blocks. In Figure 2, NNMF outperforms hNMF and DNMF in a semisupervised three-layer trial ($k^{(0)} = 9, k^{(1)} = 4, k^{(2)} = 2$).

Table 1. Reconstruction error / classification accuracy

	Layers	Hier. NMF	Deep NMF	Neural NMF
Unsuper.	1	0.053	0.031	0.029
	2	0.399	0.414	**0.310** * [a]
	3	0.860	0.838	**0.492**
Semisuper.	1	0.049 / 0.933	0.031 / 0.947	0.042 / **1**
	2	0.374 / 0.926	0.394 / 0.911	**0.305 / 1**
	3	0.676 / 0.930	0.733 / 0.930	**0.496 / 0.990** **
Supervised	1	0.052 / 0.960	0.042 / 0.962	0.042 / **1**
	2	0.311 / 0.984	0.310 / 0.984	0.307 / 1
	3	0.495 / 1	0.494 / 1	0.498 / 1

[a]Entries marked $*$ and $**$ correspond to Figure 1 and Figure 2, resp.

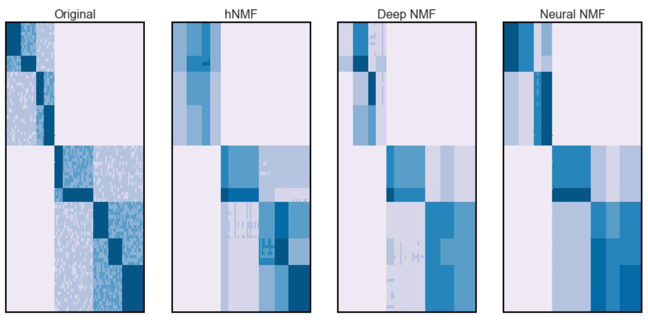

Fig. 1. Unsupervised reconstruction with two-layer structure

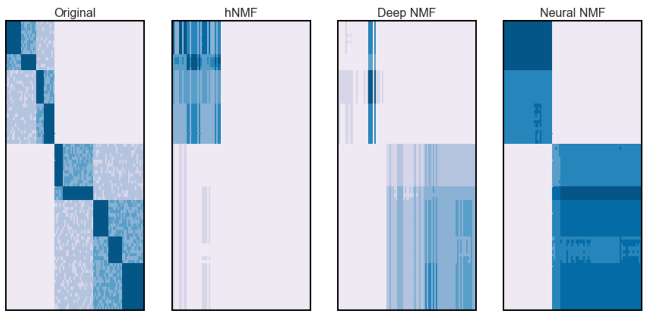

Fig. 2. Semisuper. reconstruction with three-layer structure

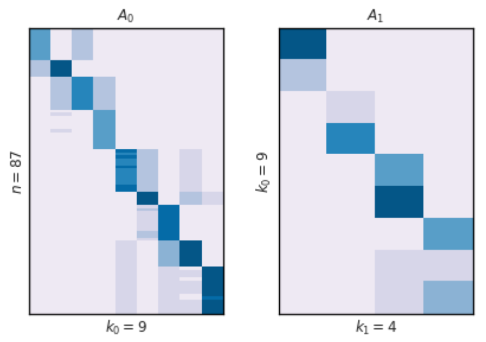

Fig. 3. Two-layer factors illustrating resolved topic hierarchy

While the reconstruction degrades as the number of layers increases (since the rank of the reconstruction decreases), NNMF resolves hierarchical topic structure in the factor matrices, $\mathbf{A}^{(i)}$. We visualize these factor matrices in Figures 3 and 4. The factor matrices in Figure 3 resolve the correct topic structure; one may see that the rows are correctly grouped into $k^{(0)} = 9$ topics in $\mathbf{A}^{(0)}$, and those topics are then correctly grouped into $k^{(1)} = 4$ supertopics in $\mathbf{A}^{(1)}$. In Figure 4, the method does not perfectly resolve the middle layer topic structure, but does resolve the initial layer topic structure.

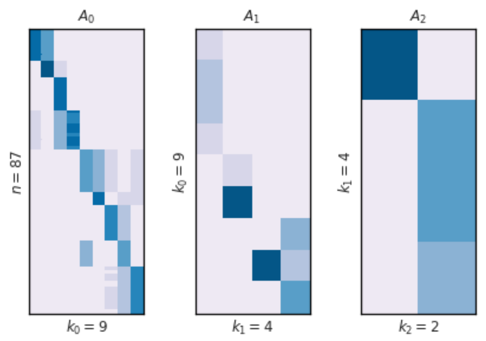

Fig. 4. Three-layer factors illustrating topic hierarchy

4. CONCLUSION

We have presented a novel method for multilayer NMF that incorporates the backpropagation technique from deep learning to minimize error accumulation. Preliminary tests on toy datasets show the proposed method outperforms existing multilayer NMF algorithms. In the near future, we plan to further compare our method and others on various datasets to find precise regimes in which we offer improvement.

5. ACKNOWLEDGMENT

The authors would like to thank Dr. Blake Hunter for proposing the project, and Prof. Andrea Bertozzi, Director of Applied Math at UCLA, for organizing the CAM REU.

6. REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[2] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. CVPR IEEE*, June 2005, vol. 2, pp. 524–531 vol. 2.

[3] Lin Liu, Lin Tang, Wen Dong, Shaowen Yao, and Wei Zhou, "An overview of topic modeling and its current applications in bioinformatics," in *SpringerPlus*, 2016.

[4] Isabelle Guyon and André Elisseeff, *An Introduction to Feature Extraction*, pp. 1–25, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[5] Ioan Buciu, "Non-negative matrix factorization, a new tool for feature extraction: theory and applications," *Int. J. Comput. Commun.*, vol. 3, no. 3, pp. 67–74, 2008.

[6] Da Kuang, Jaegul Choo, and Haesun Park, "Nonnegative matrix factorization for interactive topic modeling and document clustering," in *Partitional Clustering Algorithms*, pp. 215–243. Springer, 2015.

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

[8] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *IEEE Signal Proc. Let.*, vol. 17, no. 1, pp. 4–7, Jan 2010.

[9] David Donoho and Victoria Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?," in *Adv. Neur. In.*, 2004, pp. 1141–1148.

[10] Kejun Huang, Nicholas D Sidiropoulos, and Ananthram Swami, "Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition," *IEEE T. Signal Proces.*, vol. 62, no. 1, pp. 211–224, 2013.

[11] Nicolas Gillis, "Sparse and unique nonnegative matrix factorization through data preprocessing," *J. Mach. Learn. Res.*, vol. 13, no. Nov, pp. 3349–3386, 2012.

[12] Xiao Fu, Kejun Huang, Nicholas D Sidiropoulos, and Wing-Kin Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications," *arXiv preprint arXiv:1803.01257*, 2018.

[13] Hans Laurberg, Mads Græsbøll Christensen, Mark D Plumbley, Lars Kai Hansen, and Søren Holdt Jensen, "Theorems on positive data: On the uniqueness of nmf," *Comput. Intel. Neurosc.*, vol. 2008, 2008.

[14] Jennifer Flenner and Blake Hunter, "A deep non-negative matrix factorization neural network," 2018, Unpublished.

[15] George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Björn W Schuller, "A deep matrix factorization method for learning attribute representations," *IEEE T. Pattern Anal.*, vol. 39, no. 3, pp. 417–429, 2016.

[16] Jonathan Le Roux, John R Hershey, and Felix Weninger, "Deep nmf for speech separation," in *Int. Conf. Acoust. Spee.* IEEE, 2015, pp. 66–70.

[17] Xiaoxia Sun, Nasser M. Nasrabadi, and Trac D. Tran, "Supervised multilayer sparse coding networks for image classification," *CoRR*, vol. abs/1701.08349, 2017.

[18] Daniel D Lee and H Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems (NIPS'01)*, 2001, pp. 556–562.

[19] Pentti Paatero and Unto Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[20] Mengdi Gao, Jamie Haddock, Denali Molitor, Deanna Needell, Eli Sadovnik, Tyler Will, and Runyu Zhang, "Neural nonnegative matrix factorization for hierarchical multilayer topic modeling," In preparation.