# Neural Nonnegative CP Decomposition for Hierarchical Tensor Analysis

Joshua Vendrow
*Department of Mathematics*
*University of California, Los Angeles*
Los Angeles, United States
jvendrow@math.ucla.edu

Jamie Haddock
*Department of Mathematics*
*Harvey Mudd College*
Claremont, CA, United States
jhaddock@g.hmc.edu

Deanna Needell
*Department of Mathematics*
*University of California, Los Angeles*
Los Angeles, United States
deanna@math.ucla.edu

2021 55th Asilomar Conference on Signals, Systems, and Computers | 978-1-6654-5828-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IEEECONF53345.2021.9723126

***Abstract*—There is a significant demand for topic modeling on large-scale data with complex multi-modal structure in applications such as multi-layer network analysis, temporal document classification, and video data analysis; frequently this multi-modal data has latent hierarchical structure. We propose a new hierarchical nonnegative CANDECOMP/PARAFAC (CP) decomposition (hierarchical NCPD) model and a training method, Neural NCPD, for performing hierarchical topic modeling on multi-modal tensor data. Neural NCPD utilizes a neural network architecture and backpropagation to mitigate error propagation through hierarchical NCPD. Here, we present this hierarchical NCPD approach and demonstrate its efficacy on experiments using synthetic and temporal document data sets.**

***Index Terms*—hierarchical tensor decomposition, topic modeling, neural network, backpropagation**

## I. Introduction

The recent explosion in the collection and availability of multi-modal tensor formatdata has led to an unprecedented demand for scalable data analysis techniques [1]. The need to reduce redundant dimensions (across modes) and to identify meaningful latent trends within data has rightly become an integral focus of research within signal processing and computer science. An important application of these dimension-reduction techniques is *topic modeling*, the task of identifying latent topics and themes of a dataset in an unsupervised or partially supervised approach. A popular topic modeling approach for matrix data is the dimension-reduction technique nonnegative matrix factorization (NMF) [2], which is generalized to multi-modal tensor data by the nonnegative CP decomposition (NCPD) [3], [4]. These models identify $r$ latent *topics* within the data; here the *rank* $r$ is a user-defined parameter that can be challenging to select without *a priori* knowledge or a heuristic selection procedure.

In topic modeling applications, one often additionally wishes to understand the hierarchical topic structure (i.e., how the topics are naturally related and combine into supertopics). For matrices (tensors), a naive approach is to apply NMF (NCPD) first with rank $r$ and then again with rank $j < r$, and simply identify the $j$ supertopics as linear (multilinear) combinations of the original $r$ subtopics. However, due to the nonconvexity of the NMF (NCPD) objective function, the supertopics identified in this way need not be linearly (multilinearly) related to the subtopics. For this reason, hierarchical models that enforce these relationships between subtopics and supertopics have become a popular direction of research. A challenge of these models is that the nonconvexity of the model at each level of hierarchy can yield cascading error through the layers of models; several works have proposed techniques for mitigating this cascade of error [5], [6], [7], [8], [9].

In this work, we propose a hierarchical NCPD model and Neural NCPD, a method for training this model that exploits backpropagation techniques to mitigate the effects of error introduced at earlier (subtopic) layers of hierarchy propagating downstream to later (supertopic) layers. This approach allows us to (1) explore the topics learned at different ranks simultaneously, and (2) illustrate the *hierarchical* relationship of topics learned at different tensor decomposition ranks.

**Notation.** We follow the notational conventions of [10]; e.g., tensor $\mathbf{X}$, matrix $\boldsymbol{X}$, vector $\boldsymbol{x}$, and (integer or real) scalar $x$. In all models, we use variable $r$ (with superscripts denoting layer of hierarchical models) to denote model rank and use $j$ when indexing through rank-one components. In all tensor decomposition models, we use $k$ to denote the order (number of modes) of the tensor and use $i$ when indexing through modes of the tensor. In all hierarchical models, we use $\mathcal{L}$ to denote the number of layers in the model and use $\ell$ to index layers. We let $\otimes$ denote the vector outer product and adopt the CP decomposition notation

$$[\![\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_k]\!] \equiv \sum_{j=1}^{r} \boldsymbol{x}_j^{(1)} \otimes \boldsymbol{x}_j^{(2)} \otimes \cdots \otimes \boldsymbol{x}_j^{(k)}, \quad (1)$$

where $\boldsymbol{x}_j^{(i)}$ is the $j$th column of the $i$th factor matrix $\boldsymbol{X}_i$ [11].

**Contributions.** Our main contributions are two-fold. First, we propose a novel hierarchical nonnegative tensor decomposition model that we denote *hierarchical NCPD* (HNCPD). Our

The authors are grateful to and were partially supported by NSF CAREER DMS #1348721, NSF DMS #2011140 and NSF BIGDATA DMS #1740325. JH is also partially supported by NSF DMS #2111440.

model treats all tensor modes alike and the output is not affected by the order of the modes in the tensor representation; this is a property not shared by other hierarchical tensor decomposition models such as that of [12]. Second, we propose an effective neural network-inspired training method that we call Neural NCPD. This method builds upon the Neural NMF method proposed in [9], but is not a direct extension; Neural NCPD consists of a branch of Neural NMF for each tensor mode, but the backpropagation scheme must be adapted for factorization information flow between branches.

**Organization.** In the remainder of Section I, we present related work. In Section II, we present our main contributions, HNCPD and the Neural NCPD method. In Section III, we test Neural NCPD on real and synthetic data, and offer some brief conclusions in Section IV.

### A. Related Work

In this section, we introduce NMF, hierarchical NMF, the Neural NMF method, and NCPD, and then summarize some relevant work.

**Nonnegative Matrix Factorization (NMF).** Given a nonnegative matrix $\boldsymbol{X} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2}$, and a desired dimension $r \in \mathbb{N}$, NMF seeks to decompose $\boldsymbol{X}$ into a product of two low-dimensional nonnegative matrices; dictionary matrix $\boldsymbol{A} \in \mathbb{R}_{\geq 0}^{n_1 \times r}$ and representation matrix $\boldsymbol{S} \in \mathbb{R}_{\geq 0}^{r \times n_2}$ so that

$$\boldsymbol{X} \approx \boldsymbol{A}\boldsymbol{S} = \sum_{j=1}^{r} \boldsymbol{a}_j \otimes \boldsymbol{s}_j, \tag{2}$$

where $\boldsymbol{a}_j$ is a column (topic) of $\boldsymbol{A}$ and $\boldsymbol{s}_j$ is a row of $\boldsymbol{S}$. Typically, $r$ is chosen such that $r < \min\{n_1, n_2\}$ to reduce the dimension of the original data matrix or reveal latent themes in the data. Each column of $\boldsymbol{S}$ provides the approximation of the respective column in $\boldsymbol{X}$ in the lower-dimensional space spanned by the columns of $\boldsymbol{A}$. The nonnegativity of the NMF factor matrices yields clear interpretability; thus, NMF has found application in document clustering [13], [14], [15], and image processing and computer vision [2], [16], [17], amongst others. Popular training methods include multiplicative updates [2], [18], [19], projected gradient descent [20], and alternating least-squares [21], [22].

**Hierarchical NMF (HNMF).** HNMF seeks to illuminate hierarchical structure by recursively factorizing the NMF $\boldsymbol{S}$ matrices; see e.g., [1]. We first apply NMF with rank $r^{(0)}$ and then apply NMF with rank $r^{(1)}$ to the $\boldsymbol{S}$ matrix, collecting the $r^{(0)}$ subtopics into $r^{(1)}$ supertopics. HNMF with $\mathcal{L}$ layers approximately factors the data matrix as

$$\begin{aligned} \boldsymbol{X} &\approx \boldsymbol{A}^{(0)}\boldsymbol{S}^{(0)} \\ &\approx \boldsymbol{A}^{(0)}\boldsymbol{A}^{(1)}\boldsymbol{S}^{(1)} \\ &\vdots \\ &\approx \boldsymbol{A}^{(0)}\boldsymbol{A}^{(1)}\cdots\boldsymbol{A}^{(\mathcal{L}-1)}\boldsymbol{S}^{(\mathcal{L}-1)}. \end{aligned} \tag{3}$$

Here the $A^{(i)}$ matrix represents how the subtopics at layer $i$ collect into the supertopics at layer $i+1$. Note that as $\mathcal{L}$ increases, the error $\|\boldsymbol{X} - \boldsymbol{A}^{(0)}\boldsymbol{A}^{(1)}\cdots\boldsymbol{A}^{(\mathcal{L}-1)}\boldsymbol{S}^{(\mathcal{L}-1)}\|_F$ necessarily increases as error propagates with each step. As a result, significant error is introduced when $\mathcal{L}$ is large. Choosing $r^{(0)}, r^{(1)}, \cdots, r^{(\mathcal{L}-1)}$ in practice proves difficult as the number of possibilities grow combinatorially.

**Neural NMF (NNMF).** In the previous work of [9], the authors developed an iterative method for training HNMF that uses backpropagation techniques to mitigate cascading error through the layers. To form this hierarchical factorization, the Neural NMF method uses a neural net architecture. Each layer $\ell$ of the network has weight matrix $\boldsymbol{A}^{(\ell)}$. In the forward propagation step, the network accepts a matrix $\boldsymbol{S}^{(\ell-1)}$, calculates the nonnegative least-squares solution

$$\boldsymbol{S}^{(\ell)} = q(\boldsymbol{A}^{(\ell)}, \boldsymbol{S}^{(\ell-1)}) \equiv \underset{\boldsymbol{S}\geq 0}{\arg\min} \|\boldsymbol{S}^{(\ell-1)} - \boldsymbol{A}^{(\ell)}\boldsymbol{S}\|_F, \tag{4}$$

and sends the matrix $\boldsymbol{S}^{(\ell)}$ to the next layer. In the backpropagation step, the method calculates gradients and updates the weights of the network, which in this case are the $\boldsymbol{A}$ matrices.

**Nonnegative CP Decomposition (NCPD).** The NCPD generalizes NMF to higher-order tensors; specifically, given an order-$k$ tensor $\mathbf{X} \in \mathbb{R}_{\geq 0}^{n_1 \times n_2 \times \cdots \times n_k}$ and a fixed integer $r$, the approximate NCPD of $\mathbf{X}$ seeks $\boldsymbol{X}_1 \in \mathbb{R}_{\geq 0}^{n_1 \times r}, \boldsymbol{X}_2 \in \mathbb{R}_{\geq 0}^{n_2 \times r}, \cdots, \boldsymbol{X}_k \in \mathbb{R}_{\geq 0}^{n_k \times r}$ so that

$$\mathbf{X} \approx [\![\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_k]\!]. \tag{5}$$

The $\boldsymbol{X}_i$ matrices will be referred to as the NCPD factor matrices. A nonnegative approximation with fixed $r$ is obtained by approximately minimizing the reconstruction error between $\mathbf{X}$ and the NCPD reconstruction. This decomposition has found numerous applications in the area of *dynamic topic modeling* where one seeks to discover topic emergence and evolution [23], [24], [25]. Methods for training NMF models can often be generalized to NCPD; for example, multiplicative updates [26] and alternating least-squares [27].

**Other Related Work.** Other works have sought to mitigate error propagation in HNMF models with techniques inspired by neural networks [6], [7], [8], [5]. Additionally, previous works have developed hierarchical tensor decomposition models and methods [28], [29], [30]. The model most similar to ours is that of [12], which we refer to as hierarchical nonnegative tensor factorization (HNTF). This model consists of a sequence of NCPDs, where a factor matrix for one mode is held constant, the remaining factor matrices produce the tensor that is decomposed at the second layer, and this decomposition is combined with the fixed matrix from the previous layer. We note that, unlike our HNCPD model, HNTF is dependent upon the ordering of the modes, and specifically which data mode appears first in the representation of the tensor. We refer to 'HNTF-$i$' as HNTF applied to the representation of the tensor where the modes are reordered with mode $i$ first.

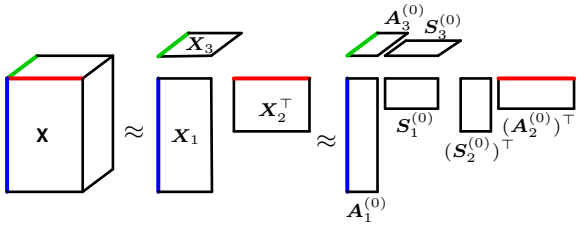


Fig. 1: A visualization of a two-layer HNCPD model. Colored edges of the order-three tensor, $\mathbf{X}$, represent the three modes.

## II. Our Contributions

In this section, we present our two main contributions. We first describe the proposed hierarchical NCPD (HNCPD) model, and then propose a training method, Neural NCPD, for the model.

### A. Hierarchical NCPD (HNCPD)

Given an order-$k$ tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times \ldots \times n_k}$, HNCPD consists of an initial rank-$r$ NCPD layer with factor matrices $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k$, each with $r$ columns, and an HNMF with ranks $r^{(0)}, r^{(1)}, \cdots, r^{(\mathcal{L}-2)}$ for each of these factors matrices; that is, for each $\boldsymbol{X}_i$ at layer $\ell$, we factorize $\boldsymbol{X}_i$ as

$$\boldsymbol{X}_i \approx \widetilde{\boldsymbol{X}}_i \equiv \boldsymbol{A}_i^{(0)} \boldsymbol{A}_i^{(1)} \ldots \boldsymbol{A}_i^{(\ell-2)} \boldsymbol{S}_i^{(\ell-2)} \tag{6}$$

where $\boldsymbol{A}_i^{(\ell)}$ has $r^{(\ell)}$ columns; see Figure 1 for a visualization. Thus, HNCPD consists of tensor approximations

$$\mathbf{X} \approx [\![ \boldsymbol{A}_1^{(0)} \ldots \boldsymbol{A}_1^{(\ell-2)} \boldsymbol{S}_1^{(\ell-2)}, \cdots, \boldsymbol{A}_k^{(0)} \ldots \boldsymbol{A}_k^{(\ell-2)} \boldsymbol{S}_k^{(\ell-2)} ]\!]. \tag{7}$$

To access hierarchical structure between tensor topics at each layer, we need to utilize information in the $\boldsymbol{S}_i^{(\ell)}$ matrices for all modes. To simplify this hierarchical structure, we develop an approximation scheme such that the hierarchical topic structure for all modes is given by a single matrix.

For simplicity, we first consider the two layer case. We note

$$[\![ \widetilde{\boldsymbol{X}}_1, \cdots, \widetilde{\boldsymbol{X}}_k ]\!] = \sum_{1 \le j_1, \ldots j_k \le r^{(0)}} \alpha_{j_1, \ldots j_k} \left( (\boldsymbol{A}_1^{(0)})_{:,j_1} \otimes \ldots \otimes (\boldsymbol{A}_k^{(0)})_{:,j_k} \right) \tag{8}$$

where $\alpha_{j_1, j_2, \ldots j_k} = \sum_{p=1}^{r} (\boldsymbol{S}_1^{(0)})_{j_1,p} (\boldsymbol{S}_2^{(0)})_{j_2,p} \ldots (\boldsymbol{S}_k^{(0)})_{j_k,p}$. We refer to decomposition summands in (8) where $j_1 = j_2 = \cdots = j_k$ as vector outer products of *same-index* factor matrix topics, and all other summands as vector outer products of *different-index* factor matrix topics. To identify clear hierarchy, we remove these different-index column outer products.

The approximation scheme computes matrices $\widetilde{\boldsymbol{A}}_i^{(0)}$ whose columns visualize the desired $r^{(0)}$ NCPD topics along each mode while removing different-index column outer products in the decomposition. We approximate the summation (8) by replacing all summands that include column $p_2$ of $\boldsymbol{A}_k^{(0)}$ with a single rank-one vector outer product, $(\widetilde{\boldsymbol{A}}_1^{(0)})_{:,p_2} \otimes (\widetilde{\boldsymbol{A}}_2^{(0)})_{:,p_2} \otimes \ldots \otimes (\widetilde{\boldsymbol{A}}_{k-1}^{(0)})_{:,p_2} \otimes (\boldsymbol{A}_k^{(0)})_{:,p_2}$. To minimize error introduced by this approximation, we transform factor matrices $\boldsymbol{A}_i^{(0)}$ for $i \neq k$ to $\widetilde{\boldsymbol{A}}_i^{(0)}$ by collecting into $(\widetilde{\boldsymbol{A}}_i^{(0)})_{:,p_2}$ the approximate contribution of all columns of $\boldsymbol{A}_i^{(0)}$ in vector outer products with $(\boldsymbol{A}_k^{(0)})_{:,p_2}$ in (8). That is, for $1 \le p_1, p_2 \le r^{(0)}$ and $1 \le i < k$, let $\boldsymbol{W}_i \in \mathbb{R}^{r^{(0)} \times r^{(0)}}$ be a matrix with

$$(\boldsymbol{W}_i)_{p_1,p_2} = \sum_{j_i = p_1, j_k = p_2, 1 \le j_1, j_2, \ldots j_k \le r^{(0)}} \alpha_{j_1, j_2, \ldots, j_k} \quad \text{and} \quad \widetilde{\boldsymbol{A}}_i^{(0)} = \boldsymbol{A}_i^{(0)} \boldsymbol{W}_i. \tag{9}$$

After this transformation, we can identify the topic hierarchy in the HNCPD from the $\boldsymbol{S}_k^{(0)}$ matrix. We can generalize this process to later layers $\ell$ by noting that we can group the $\boldsymbol{A}_i$ matrices together, so $\boldsymbol{X}_i \approx \left( \boldsymbol{A}_i^{(0)} \boldsymbol{A}_i^{(1)} \ldots \boldsymbol{A}_i^{(\ell)} \right) \boldsymbol{S}_i^{(\ell)}$. Thus, we can treat this approximation as above, replacing $\boldsymbol{A}_i^{(0)}$ with the product $\boldsymbol{A}_i^{(0)} \boldsymbol{A}_i^{(1)} \ldots \boldsymbol{A}_i^{(\ell)}$.

Like in HNMF, errors in earlier layers can propagate through to later layers and produce highly suboptimal approximations. Challenges encountered during computation of HNMF are exacerbated in an HNCPD model. For this reason, we exploit approaches developed for HNMF in [9] in our training method Neural NCPD. Furthermore, the computation of HNMF factor matrices for $\boldsymbol{X}_i$ are independent from $\boldsymbol{X}_j$ if the factorizations are applied sequentially; Neural NCPD allows factor matrices in (6) for all other modes to influence the factorization of a given mode.

### B. Neural NCPD

Our iterative method consists of two subroutines, a forward-propagation and a backpropagation. In Algorithms 1 and 2, we display the pseudocode for our proposed method. Following this learning process for the factor matrices in (6), we apply the approximation scheme described in Section II-A to the learned factor matrices to visualize the hierarchical structure of the computed HNCPD model.

**Algorithm 1** Forward Propagation

**procedure** ForwardProp($\{\boldsymbol{X}_i\}_{i=0}^{k}, \{\boldsymbol{A}_i^{(\ell)}\}_{i=0,\ell=0}^{k,\mathcal{L}-2}$)
  **for** $i = 1, \cdots, k$ **do**
    **for** $\ell = 0, \cdots, \mathcal{L}-2$ **do**
      $\boldsymbol{S}_i^{(\ell)} \leftarrow q(\boldsymbol{A}_i^{(\ell)}, \boldsymbol{S}_i^{(\ell-1)})$ ▷ see equation 4

**Algorithm 2** Neural NCPD

**Input:** Tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_k}$, cost $C$
$\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_k \leftarrow$ `NCPD`($\mathbf{X}$), initialize $\{\boldsymbol{A}_i^{(\ell)}\}_{i=0,\ell=0}^{k,\mathcal{L}-2}$
**for** iterations $= 1, \ldots, T$ **do**
  `ForwardProp`($\{\boldsymbol{X}_i\}_{i=0}^{k}, \{\boldsymbol{A}_i^{(\ell)}\}_{i=0,\ell=0}^{k,\mathcal{L}-2}$) ▷ Alg. 1
  **for** $i = 1, \cdots, k, \ \ell = 0, \cdots, \mathcal{L}-2$ **do**
    $\boldsymbol{A}_i^{(\ell)} \leftarrow \left( \texttt{optimizer}\left( \boldsymbol{A}_i^{(\ell)}, \frac{\partial C}{\partial \boldsymbol{A}_i^{(\ell)}} \right) \right)^{+}$
    ▷ any first-order method

**Forward Propagation.** The forward-propagation treats $\boldsymbol{A}_i^{(\ell)}$ matrices as neural network weights and uses $\boldsymbol{A}_i^{(\ell)}$ and previous layer output to compute

$$\boldsymbol{S}_i^{(\ell)} = q(\boldsymbol{A}_i^{(\ell)}, \boldsymbol{S}_i^{(\ell-1)}), \tag{10}$$

where $q$ is as defined in equation 4 and $\boldsymbol{S}_i^{(-1)} = \boldsymbol{X}_i$, producing the matrices $\boldsymbol{S}_i^{(0)}, \ldots, \boldsymbol{S}_i^{(\mathcal{L}-2)}$ for $1 \leq i \leq k$. The function $q(\boldsymbol{A}^{(\ell)}, \boldsymbol{S}^{(\ell-1)})$, as a nonnegative least-squares problem, can be calculated via any convex optimization solver; we utilize an implementation of the Hanson-Lawson algorithm [31]. Finally, we pass the $\boldsymbol{A}_i^{(\ell)}$ and $\boldsymbol{S}_i^{(\ell)}$ matrices and $\mathbf{X}$ into a loss function, which we differentiate and backpropagate.

**Backpropagation.** Here, we differentiate the cost function $C$ with respect to the weights in each layer, the $\boldsymbol{A}_i^{(\ell)}$ matrices, and backpropagate. This method accepts any first-order optimization method, denoted `optimizer` (e.g., SGD [32], Adam [33]), but projects the updated weight matrix into the positive orthant to maintain nonnegativity.

For the NCPD task, the most natural loss function is the *reconstruction loss*,

$$C = \|\mathbf{X} - [\![\widetilde{\boldsymbol{X}}_1, \widetilde{\boldsymbol{X}}_2, \cdots, \widetilde{\boldsymbol{X}}_k]\!]\|_F. \tag{11}$$

In order to encourage optimal fit at each layer, we also introduce a loss function that we refer to as *energy loss*. We denote the approximation of $\mathbf{X}$ at layer $\ell$ of our network as

$$\mathbf{X}_\ell = [\![\boldsymbol{A}_1^{(0)} \ldots \boldsymbol{A}_1^{(\ell-2)} \boldsymbol{S}_1^{(\ell-2)}, \ldots, \boldsymbol{A}_k^{(0)} \ldots \boldsymbol{A}_k^{(\ell-2)} \boldsymbol{S}_k^{(\ell-2)}]\!]. \tag{12}$$

Then, we calculate energy loss as

$$E = \|\mathbf{X} - [\![\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_k]\!]\|_F + \sum_{\ell=0}^{\mathcal{L}-2} \|\mathbf{X} - \mathbf{X}_\ell\|_F. \tag{13}$$

The derivatives of $q(\boldsymbol{A}, \boldsymbol{X})$ with respect to $\boldsymbol{A}$ and $\boldsymbol{X}$ are derived and exploited to differentiate a generic cost function for the hierarchical NMF model in [9]; here we summarize these derivatives and illustrate how to combine them with simple multilinear algebra for HNCPD.

Results in [9] show that, if $\left(\frac{\partial C}{\partial \boldsymbol{A}_i^{(\ell_1)}}\right)^{\boldsymbol{S}}$ is the derivative of $C$ with respect to $\boldsymbol{A}_i^{(\ell_1)}$ holding the $\boldsymbol{S}$ matrices constant, then

$$\frac{\partial C}{\partial \boldsymbol{A}_i^{(\ell_1)}} = \left(\frac{\partial C}{\partial \boldsymbol{A}_i^{(\ell_1)}}\right)^{\boldsymbol{S}} + \sum_{\substack{\ell_1 \leq \ell_2 \leq \mathcal{L}-2 \\ 1 \leq j \leq r}} \boldsymbol{U}_i^{(\ell_1,\ell_2),j}, \tag{14}$$

where $\boldsymbol{U}_i^{(\ell_1,\ell_2),j}$ relates $C$ to $\boldsymbol{A}_i^{(\ell_1)}$ through $\boldsymbol{S}_i^{(\ell_2)}$ and $\boldsymbol{S}_i^{(\ell_1)}$, is defined column-wise ($j$), and depends upon $\left(\frac{\partial C}{\partial \boldsymbol{S}_i^{(\ell_2)}}\right)^*$, the derivative of $C$ with respect to $\boldsymbol{S}_i^{(\ell_2)}$ holding $\boldsymbol{S}_i^{(\ell_2+1)}$, $\ldots$, $\boldsymbol{S}_i^{(\mathcal{L}-2)}$ constant. The definition of $\boldsymbol{U}_i^{(\ell_1,\ell_2),j}$ is given in [9] and utilizes, via the chain-rule, the partial derivative of $q(\boldsymbol{A}_i^{\ell}, \boldsymbol{S}_i^{\ell-1})$ for all $\ell \in [\ell_1, \ell_2]$.

**Example.** The derivative of the previously defined, or other, differentiable cost functions can be calculated using these results of [9] and some simple multi-linear algebra. As an example, we directly compute the backpropagation step for the reconstruction loss function $C$ given in equation 11. Let $\boldsymbol{X}_{(i)}$ be the mode-$i$ matricized version of $\mathbf{X}$, and define

$$\boldsymbol{H}_i = \widetilde{\boldsymbol{X}}_k \odot \ldots \odot \widetilde{\boldsymbol{X}}_{i+1} \odot \widetilde{\boldsymbol{X}}_{i-1} \odot \ldots \odot \widetilde{\boldsymbol{X}}_1, \tag{15}$$

where $\odot$ denotes the Khatri-Rao product (see e.g., [11]). Then we have that

$$\left(\frac{\partial C}{\partial \boldsymbol{A}_i^{(\ell_j)}}\right)^{\boldsymbol{S}} = 2\left(\boldsymbol{A}_i^{(0)} \cdots \boldsymbol{A}_i^{(\ell_j-1)}\right)^\top \left(\boldsymbol{X}_{(i)} - \widetilde{\boldsymbol{X}}_i \boldsymbol{H}_i^\top\right) \boldsymbol{H}_i \left(\boldsymbol{A}_i^{(\ell_{j+1})} \cdots \boldsymbol{A}_i^{(\mathcal{L}-2)} \boldsymbol{S}_i^{(\mathcal{L}-2)}\right)^\top, \tag{16}$$

$$\text{and } \left(\frac{\partial C}{\partial \boldsymbol{S}_i^{(\ell_j)}}\right)^* = 2\left(\boldsymbol{A}_i^{(0)} \cdots \boldsymbol{A}_i^{(\ell_j)}\right)^\top \left(\boldsymbol{X}_{(i)} - \widetilde{\boldsymbol{X}}_i \boldsymbol{H}_i^\top\right) \boldsymbol{H}_i. \tag{17}$$

With equation 14, these derivatives are sufficient to calculate the partial derivative of $C$ with respect to any $\boldsymbol{A}$ matrix.

## III. Experimental Results

We test Neural NCPD on two datasets: one synthetic and one collected from Twitter. The synthetic dataset is constructed as a simple block tensor with hierarchical structure. The Twitter dataset consists of tweets from political candidates during the 2016 United States presidential election [34].We also compare Neural NCPD to *Standard NCPD*, in which we perform an independent NCPD decomposition at each rank, and to *Standard HNCPD*, in which we perform NCPD first on the full dataset, and apply HNMF to the fixed factor matrices; here we sequentially apply NMF to the factor matrices using multiplicative updates and do not update previous layer factorizations as in Neural NCPD. In all experiments, we use `Tensorly` [35] for Standard NCPD calculations and to initialize the NCPD layer of our hierarchical NCPD, and in Neural NCPD we do not backpropagate to this layer as the initialization has usually found a stationary point. We use Energy Loss (Eq. 13) for all experiments to encourage fit at every layer. Because we do not backpropagage to the initial factor matrices, the first term in (Eq. 13) is fixed. For the Twitter experiment, we use the approximation scheme of Section II-A to recover the relationship between the columns of the $\boldsymbol{A}_i^{(\ell)}$ matrices and visualize the $\widetilde{\boldsymbol{A}}_i^{(\ell)}$ matrices.

### A. Experiment on Synthetic Data

We test the Neural NCPD method first using a synthetic dataset. This dataset is a rank seven tensor of size $40 \times 40 \times 40$ with positive noise added to each entry; we generate noise as $n = |g|$ where $g \sim \mathcal{N}(0, \sigma^2)$. To generate this dataset, we begin with the all-zeros tensor and create two large nonoverlapping blocks with value 0.75, and then overlay each block with either two or additional blocks with value 2. Finally, we add two long parallel tubes to three of the four additional

TABLE I: Topic modeling loss and relative reconstruction loss, $C_{\text{rel}}$, on the synthetic dataset for Neural NCPD, Standard HNCPD, HNTF, Neural NMF, and Standard HNMF with two levels of noise over 10 trials. For HNTF we report runs on three re-orderings of the modes the tensor, and for matrix methods we report results for flattening along each mode of the tensor.

| | | Topic Modeling Loss | | | | | | Relative Reconstruction Loss | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\sigma^2 = 0.1$ | | | $\sigma^2 = 0.4$ | | | $\sigma^2 = 0.1$ | | | $\sigma^2 = 0.4$ | | |
| Method | Mode | $7-2$ | $7-4$ | $4-2$ | $7-2$ | $7-4$ | $4-2$ | $r=7$ | $r^{(0)}=4$ | $r^{(1)}=2$ | $r=7$ | $r^{(0)}=4$ | $r^{(1)}=2$ |
| Neural HNCPD | | **0.043** | 0.042 | **0.042** | **0.087** | **0.087** | **0.081** | 0.119 | 0.252 | 0.563 | **0.454** | 0.508 | 0.714 |
| Standard HNCPD | | 0.106 | 0.101 | 0.189 | 0.145 | 0.193 | 0.204 | 0.119 | 0.494 | 0.828 | **0.454** | 0.612 | 0.892 |
| HNTF [12] | 1 | 0.163 | 0.236 | 0.182 | 0.171 | 0.144 | 0.170 | 0.119 | 0.502 | 0.795 | **0.454** | 0.576 | 0.781 |
| | 2 | 0.087 | **0.040** | 0.101 | 0.090 | 0.116 | 0.142 | 0.119 | 0.309 | 0.665 | **0.454** | 0.587 | 0.765 |
| | 3 | 0.078 | 0.122 | 0.106 | 0.084 | 0.111 | 0.164 | 0.119 | 0.417 | 0.713 | **0.454** | 0.560 | 0.747 |
| Neural NMF [9] | 1 | 0.154 | 0.192 | 0.105 | 0.169 | 0.219 | 0.127 | 0.146 | 0.268 | 0.593 | 0.478 | 0.521 | 0.705 |
| | 2 | 0.075 | 0.244 | 0.146 | 0.153 | 0.190 | 0.160 | 0.141 | 0.289 | 0.585 | 0.475 | 0.513 | 0.710 |
| | 3 | 0.119 | 0.164 | 0.110 | 0.158 | 0.197 | 0.140 | 0.151 | 0.236 | 0.576 | 0.477 | 0.512 | **0.693** |
| Standard HNMF | 1 | 0.098 | 0.182 | 0.052 | 0.164 | 0.219 | 0.139 | **0.118** | 0.235 | **0.558** | 0.472 | 0.524 | 0.707 |
| | 2 | 0.080 | 0.199 | 0.090 | 0.151 | 0.213 | 0.088 | **0.118** | 0.245 | 0.566 | 0.472 | 0.505 | 0.709 |
| | 3 | 0.060 | 0.165 | 0.085 | 0.137 | 0.193 | 0.114 | **0.118** | **0.233** | 0.563 | 0.472 | **0.503** | 0.717 |

blocks with value 3, which meet at one edge, so that each of these tubes matches the shape of the blocks that contain it in one of the three modes. We display a partial visualization of this tensor with two levels of noise at the left of Figure 2; here we plot projections of all tensors (and all approximations) along the third mode; that is, we construct a matrix with entries equal to the largest entries of the mode-three fibers (see e.g., [11] for relevant definitions). For our synthetic tensor, the two top left yellow blocks in this projection hide the true structure of this figure, illustrating how construction of the tensor obscures topic structure from matrix-based methods.

We add comparisons to Hierarchical NMF methods by applying these methods to each of the three possible flattenings of the tensor into a matrix along each of the three modes, as described in [11]. The synthetic data tensor was constructed in such a way that for each flattening along a given mode, a part of the hierarchical structure is obscured. Specifically, each of the three flattened tensors have matrix rank 6, whereas the original tensor has CP rank 7. Thus, we expect that HNMF methods applied to these matrices would produce an incomplete hierarchical structure, even if they can approximate the matrices with low reconstruction loss. We compare to Neural NMF and *Standard HNMF*, in which we do not update previous layer factorizations.

To evaluate the topic models produced by each method, we introduce a metric that we refer to as *topic error*. Given a matrix $\boldsymbol{M}_{\text{true}} \in \mathbb{R}^{r^{(i)} \times r^{(j)}}$ that represents the underlying hierarchical relationship between the $r^{(i)}$ subtopics and the $r^{(j)}$ supertopics, and a matrix $\boldsymbol{M}_{\text{pred}} \in \mathbb{R}^{r^{(i)} \times r^{(j)}}$ produced by the HNCPD or HNMF method, our topic error measures the difference between these matrices subject to permutation of rows and columns. We normalize the rows of $\boldsymbol{M}_{\text{true}}$ and $\boldsymbol{M}_{\text{pred}}$ and interpret each row $i$ as the association and learned association, respectively, between subtopic $i$ and each of the supertopics at rank $r^{(j)}$. Because the model approximations are constant under identical row and column permutations of successive layer factor matrices, we define the error as

$$1 - \frac{1}{r^{(i)}} \min_{\boldsymbol{P}_1 \in S_{r^{(i)}}, \boldsymbol{P}_2 \in S_{r^{(j)}}} \|M_{\text{true}} \odot \boldsymbol{P}_1 M_{\text{pred}} \boldsymbol{P}_2\|_1 \qquad (18)$$

which penalizes deviation of rows of $\boldsymbol{P}_1 M_{\text{pred}} \boldsymbol{P}_2$ from those of $M_{\text{true}}$. For Neural NCPD and Standard HNCPD, we use the matrices $\boldsymbol{S}^{(1)}$, $\boldsymbol{S}^{(0)}$, and $\boldsymbol{A}^{(1)}$. As we do not introduce notation for HNTF due to space constraints, we simply note that we compare to the same topic modeling metric applied to factor matrices learned by this model. For Neural NMF and Standard HNMF, we use $\boldsymbol{A}_2$, $\boldsymbol{A}_1$, and $\boldsymbol{A}_1\boldsymbol{A}_2$.

We run Neural NCPD, Standard HNCPD, HNTF, Neural NMF, and Standard HNMF on this synthetic dataset with two different levels of noise added and with three layers with ranks 7, 4, and 2, and display the results in Figure 2 and Table I; we present the relative reconstruction loss $C_{\text{rel}} = \|\mathbf{X} - [\![\widetilde{\boldsymbol{X}}_1, \widetilde{\boldsymbol{X}}_2, \cdots, \widetilde{\boldsymbol{X}}_k]\!]\|_F / \|\mathbf{X}\|_F$, as well as topic modelling loss between every two layers. For each level of noise, we display the rank 7 approximation shared by all the NCPD methods, and the rank 4 and rank 2 approximations produced by Neural NCPD, Standard HNCPD, and HNTF. We also display the corresponding topic modelling matrices used in calculating the topic modelling loss for each method. For each of the topic modelling matrices, we normalize along the small dimension in order to interprate the matrices as probabilities that each subtopic is associated to each of the supertopics.

From Table I, we see that the the topic modelling loss for Neural NCPD is less than that of every other NCPD and NMF model for all but one level of noise and pair of ranks, and the reconstruction loss for Neural NCPD is no more than that of Standard HNCPD and HNTF at each rank and level of noise. As expected, though the NMF models produced, in most cases, lower reconstruction loss than the NCPD methods, they did not produced lower topic modeling loss than Neural NCPD, suggesting that the improved reconstruction provided by NMF comes at the cost of topic interpretability.

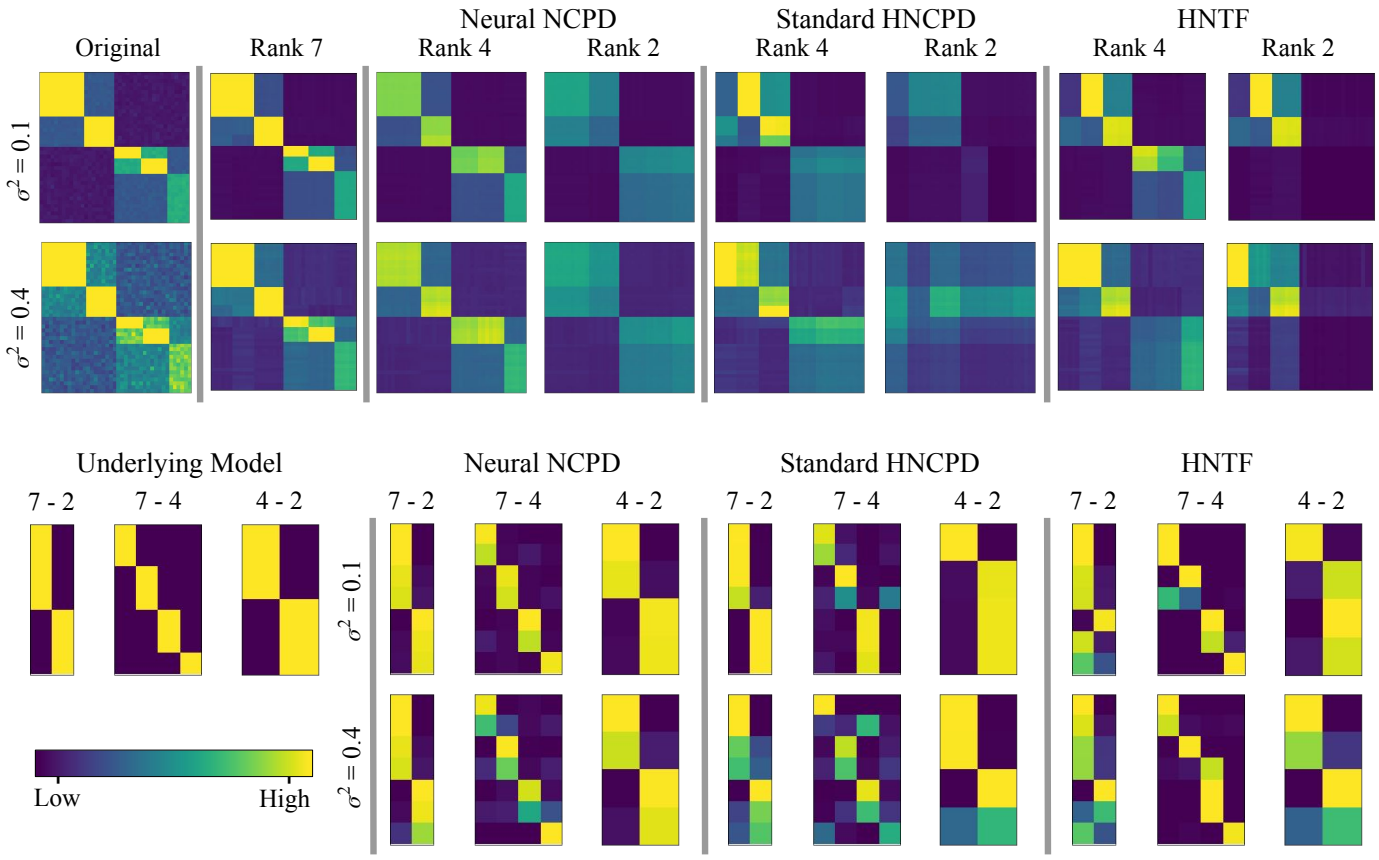


Fig. 2: (Top left) Data tensor $\mathbf{X}$ with two levels of noise. (Top right) ranks 7, 5, and 3 Neural NCPD, Standard HNCPD, and HNTF approximations of $\mathbf{X}$. (Bottom left) Underlying topic modelling matrix. (Bottom right) topic modelling matrices discovered by Neural NCPD, Standard HNCPD, and HNTF, with rows and columns permuted to minimize topic error.

### *B. Temporal Document Analysis*

We next apply Neural NCPD to a dataset of tweets from four Republican [R] and four Democratic [D] 2016 presidential primary candidates, (1) Hillary Clinton [D], (2) Tim Kaine [D], (3) Martin O'Malley [D], (4) Bernie Sanders [D], (5) Ted Cruz [R], (6) John Kasich [R], (7) Marco Rubio [R], and (8) Donald Trump [R]; this is constructed from a subset of the dataset of [34]. We use a bag-of-words (12,721 words in corpus) representation of all tweets made by a candidate within bins of 30 days (from February to December 2016), and cap each of these groups at 100 tweets to avoid oversampling from any candidate; resulting in a tensor of size $8 \times 10 \times 12721$.

In Table II, we display the relative reconstruction loss on the Twitter political dataset for all models. We see that Neural NCPD significantly outperforms Standard HNCPD, slightly outperforms Standard NCPD while additionally producing a hierarchical topic structure, and outperforms all HNTF-$i$, for which loss varies significantly based on the arrangement of the tensor. In Figure 3, we show the topic keywords and factor matrices of a rank 8, 4, and 2 hierarchical NCPD approximation computed by Neural NCPD. Note that in the rank 8 candidates mode factor and keywords we see that nearly every topic is identified with a single candidate. Topic two of the rank 8 approximation aligns with political issues (the Zika virus and the Venezuelan government) rather than a single candidate, and is temporally most present in May to July 2016 (during the Zika outbreak and the Venezualan state of emergency). Topics one and eight, corresponding to candidates Clinton and Trump, are most present in the months immediately leading up to the election. At rank 4, we see that topics one and four are inherited from the rank 8 approximation, topic two combines the rank 8 topics of candidates Trump and Clinton (final

TABLE II: Relative reconstruction loss, $C_{\text{rel}}$, on the Twitter political dataset for Neural NCPD, Standard NCPD, Standard HNCPD, and HNTF at ranks $r = 8$, $r^{(0)} = 4$, and $r^{(1)} = 2$. For HNTF we display the loss given the three possible arrangements of the tensor.

| Method | $r = 8$ | $r^{(0)} = 4$ | $r^{(1)} = 2$ |
|---|---|---|---|
| Neural NCPD | 0.834 | **0.883** | **0.918** |
| Standard NCPD | 0.834 | 0.889 | 0.919 |
| Standard HNCPD | 0.834 | 0.931 | 0.950 |
| HNTF-1 [12] | 0.834 | 0.890 | 0.927 |
| HNTF-2 [12] | 0.834 | 0.909 | 0.956 |
| HNTF-3 [12] | 0.834 | 0.895 | 0.942 |

Rank 8 Topics

| Topic 1 | Topic 2 | Topic 3 | Topic 4 |
|---|---|---|---|
| trump | senate | martinomalley | berniesanders |
| hillary | florida | hillaryclinton | people |
| donald | zika | realdonaldtrump | bernie |
| president | venezuela | campaigning | must |
| timkaine | nicolasmaduro | maryland | change |
| **Topic 5** | **Topic 6** | **Topic 7** | **Topic 8** |
| tedcruz | johnkasich | marcorubio | crooked |
| cruz | kasich | teammarco | hillary |
| ted | ohio | vote | thank |
| internet | john | flsen | great |
| choosecruz | gov | click | clinton |

Rank 4 Topics

| Topic 1 | Topic 2 |
|---|---|
| marcorubio | trump |
| teammarco | hillary |
| vote | donald |
| flsen | people |
| click | vote |
| **Topic 3** | **Topic 4** |
| tedcruz | senate |
| cruz | florida |
| ted | zika |
| johnkasich | venezuela |
| kasich | nicolasmaduro |

Rank 2 Topics

| Topic 1 |
|---|
| trump |
| hillary |
| vote |
| people |
| donald |
| **Topic 2** |
| tedcruz |
| cruz |
| ted |
| johnkasich |
| kasich |

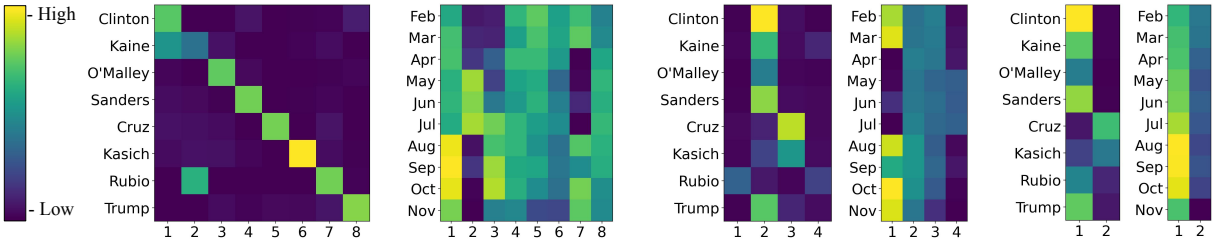


Fig. 3: A three-layer Neural NCPD on the Twitter dataset at ranks $r = 8$, $r^{(0)} = 4$ and $r^{(1)} = 2$. At each rank, we display the top keywords and topic heatmaps for candidate and temporal modes.

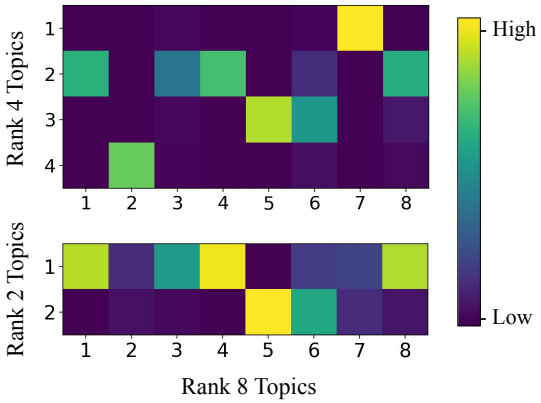


Fig. 4: The $\boldsymbol{S}_3^{(0)}$ (top) and $\boldsymbol{S}_3^{(1)}$ (bottom) matrices produced by Neural NCPD on the Twitter dataset.

candidates), and topic 3 combined the topics of candidates Cruz and Kasich (Republicans). Meanwhile, the rank 2 NCPD topics are nearly identical to rank 4 NCPD topics 2 and 3.

In Figure 4, we display the $\boldsymbol{S}_3^{(0)}$ (top) and $\boldsymbol{S}_3^{(1)}$ (bottom) matrices produced by Neural NCPD on the Twitter dataset, which illustrate how topics collect at each rank. We see topics 5 and 6 from the rank 8 factorization combine to form topic 3 at rank 4 and topic 2 at rank 2. This is expected as both topics include keywords from Cruz and Kasich, who had high presence in topics 5 and 6 respectively.

In Figure 5, we display the results of performing separate NCPD decomposition of ranks 4 and 2 on the Twitter dataset. We see that the results are similar to those of Neural NCPD,

Rank 4 Topics

| Topic 1 | Topic 2 |
|---|---|
| trump | berniesanders |
| hillary | people |
| johnkasich | bernie |
| ohio | mustt |
| kasich | vote |
| **Topic 3** | **Topic 4** |
| crooked | tedcruz |
| hillary | cruz |
| thank | ted |
| marcorubio | internet |
| great | choosecruz |

Rank 2 Topics

| Topic 1 |
|---|
| trump |
| hillary |
| vote |
| people |
| donald |
| **Topic 2** |
| tedcruz |
| cruz |
| ted |
| johnkasich |
| kasich |

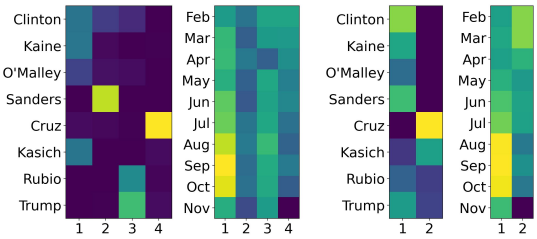


Fig. 5: Ranks 4 and 2 Standard NCPD of the Twitter dataset. At each rank, we display the top five keywords and candidate and temporal mode heatmaps.

but these independent decompositions lack the clear hierarchical structure provided by Neural NCPD. While the topics corresponding to Kasich and Clinton combine in the rank 4 NCPD, these candidates are present in *different* topics in the

rank 2 NCPD; Neural NCPD prevents this breach of hierarchy.

## IV. Conclusions

In this paper, we introduced the hierarchical NCPD model and presented a novel method, Neural NCPD, to train this decomposition. We empirically demonstrate the promise of this method on both real and synthetic datasets; in particular, this model reveals the hierarchy of topics learned at different NCPD ranks, which is not available to standard NCPD or NMF-based approaches.

## References


[1] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

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

[3] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[4] R. A. Harshman *et al.*, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis," 1970.

[5] J. Flenner and B. Hunter, "A deep non-negative matrix factorization neural network," 2018. Unpublished.

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

[7] J. Le Roux, J. R. Hershey, and F. Weninger, "Deep NMF for speech separation," in *Proc. Int. Conf. Acoust. Spee.*, pp. 66–70, IEEE, 2015.

[8] X. Sun, N. M. Nasrabadi, and T. D. Tran, "Supervised multilayer sparse coding networks for image classification," *CoRR*, vol. abs/1701.08349, 2017.

[9] M. Gao, J. Haddock, D. Molitor, D. Needell, E. Sadovnik, T. Will, and R. Zhang, "Neural nonnegative matrix factorization for hierarchical multilayer topic modeling," in *Proc. Int. Workshop on Comp. Adv. in Multi-Sensor Adaptive Process.*, 2019.

[10] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT Press, 2016.

[11] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[12] A. Cichocki, R. Zdunek, and S.-i. Amari, "Hierarchical als algorithms for nonnegative matrix and 3d tensor factorization," in *Int. Conf. on Indep. Component Anal. and Sig. Separ.*, pp. 169–176, Springer, 2007.

[13] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 267–273, 2003.

[14] E. Gaussier and C. Goutte, "Relation between PLSA and NMF and implications," in *Proc. 28th ACM SIGIR Conf. Res. Develop. Infor. Retriev.*, pp. 601–602, 2005.

[15] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Information Processing & Management*, vol. 42, no. 2, pp. 373–386, 2006.

[16] D. Guillamet and J. Vitria, "Non-negative matrix factorization for face recognition," in *Proc. Catalonian Conf. on Artif. Intell.*, pp. 336–344, Springer, 2002.

[17] P. O. Hoyer, "Non-negative sparse coding," in *Proc. 12th IEEE Workshop on Neural Networks for Signal Processing*, pp. 557–565, IEEE, 2002.

[18] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neur. In.*, pp. 556–562, 2001.

[19] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *IEEE Signal Proc. Let.*, vol. 17, no. 1, pp. 4–7, 2009.

[20] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, 2007.

[21] D. Kim, S. Sra, and I. S. Dhillon, "Fast projection-based methods for the least squares nonnegative matrix approximation problem," *Stat. Anal. Data Min.*, vol. 1, no. 1, pp. 38–51, 2008.

[22] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM J. Matrix Anal. A.*, vol. 30, no. 2, pp. 713–730, 2008.

[23] A. Cichocki, R. Zdunek, and S.-i. Amari, "Nonnegative matrix and tensor factorization [lecture notes]," *IEEE Signal Proc. Mag.*, vol. 25, no. 1, pp. 142–145, 2007.

[24] A. Traoré, M. Berar, and A. Rakotomamonjy, "Non-negative tensor dictionary learning," 2018.

[25] A. Saha and V. Sindhwani, "Learning evolving and emerging topics in social media: a dynamic NMF approach with temporal regularization," in *Proc. ACM Int. Conf. Web Search Data Min.*, pp. 693–702, 2012.

[26] M. Welling and M. Weber, "Positive tensor factorization," *Pattern Recogn. Lett.*, vol. 22, no. 12, pp. 1255–1261, 2001.

[27] J. Kim, Y. He, and H. Park, "Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework," *J. Global Optim.*, vol. 58, no. 2, pp. 285–319, 2014.

[28] M. A. O. Vasilescu and E. Kim, "Compositional hierarchical tensor factorization: Representing hierarchical intrinsic and extrinsic causal factors," *arXiv preprint arXiv:1911.04180*, 2019.

[29] L. Song, M. Ishteva, A. Parikh, E. Xing, and H. Park, "Hierarchical tensor decomposition of latent tree graphical models," in *Proc. Int. Conf. Mach. Learn.*, pp. 334–342, 2013.

[30] L. Grasedyck, "Hierarchical singular value decomposition of tensors," *SIAM J. Matrix Anal. A.*, vol. 31, no. 4, pp. 2029–2054, 2010.

[31] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. SIAM, 1995.

[32] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, pp. 400–407, 1951.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] J. Littman, L. Wrubel, and D. Kerchner, "2016 United States Presidential Election Tweet Ids," 2016.

[35] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic, "TensorLy: Tensor learning in Python," *CoRR*, vol. abs/1610.09555, 2018.