
Hierarchical Data Representation Model - Multi-layer NMF

Hyun Ah Song

Department of Electrical Engineering
KAIST
Daejeon, 305-701
hyunahsong@kaist.ac.kr

Soo-Young Lee

Department of Electrical Engineering
KAIST
Daejeon, 305-701
sylee@kaist.ac.kr

Abstract

In this paper, we propose a data representation model that demonstrates hierarchical feature learning using nsNMF. We extend unit algorithm into several layers. Experiments with document and image data successfully discovered feature hierarchies. We also prove that proposed method results in much better classification and reconstruction performance, especially for small number of features.

1 Introduction

In order to understand complex data, hierarchical feature extraction strategy has been used [1]. One best known algorithm is Deep Belief Network (DBN) introduced in 2006 [2]. With the success of training deep architectures, several variants of deep learning have been introduced [3]. Although these multi-layered algorithms take hierarchical approaches in feature extraction and provide efficient solution to complex problems, they do not provide us the relationships of features in form of hierarchies that are learned throughout the hierarchical structure.

In this paper, we propose a hierarchical data representation model, hierarchical multi-layer non-negative matrix factorization. (Similar approach has been introduced in [4].) We extend a variant of NMF algorithm [5], nsNMF [6] into several layers for hierarchical learning. Here, we demonstrate intuitive feature hierarchies present in the data set by learning relationships between features across layers. We also prove that instead of one step learning, hierarchical approach learns more meaningful and helpful features, which leads to better distributed representations, and results in better performance in classification and reconstruction for small number of features, which guarantees reduced loss of performance, even when representing data in small dimensions.

2 Non-smooth non-negative matrix factorization (nsNMF)

Proposed network is constructed by stacking nsNMF [6] into several layers. Non-smooth non-negative matrix factorization (nsNMF) is a variant of NMF that restricts sparsity constraint. Basic NMF decomposes non-negative input data \mathbf{X} into non-negative \mathbf{W} and \mathbf{H} , which are features and corresponding coefficients or data representation respectively. It aims to reduce error between original data \mathbf{X} and its reconstruction \mathbf{WH} : $C = \frac{1}{2} \|\mathbf{X} - \mathbf{WH}\|^2 = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n (X_{ij} - \sum_{k=1}^l W_{ik} H_{kj})^2$.

To apply sparsity constraint to standard NMF, a sparsity matrix \mathbf{S} is introduced in [6]: $\mathbf{S} = (1 - \theta)\mathbf{I}(k) + \frac{\theta}{k}\mathbf{ones}(k)$. k is number of features, and θ is parameter for smoothing effect, in range of 0 to 1. $\mathbf{I}(k)$ is identity matrix of size $k \times k$, and $\mathbf{ones}(k)$ is a matrix of size $k \times k$ with all components of 1s. We smooth a matrix by multiplying it with \mathbf{S} . The closer θ is to 1, more smoothing effect is applied. During alternative update, we smooth \mathbf{H} matrix by multiplying \mathbf{S} and \mathbf{H} during iterations as $\mathbf{H}=\mathbf{SH}$. To compensate the loss of sparsity, \mathbf{W} becomes sparse.

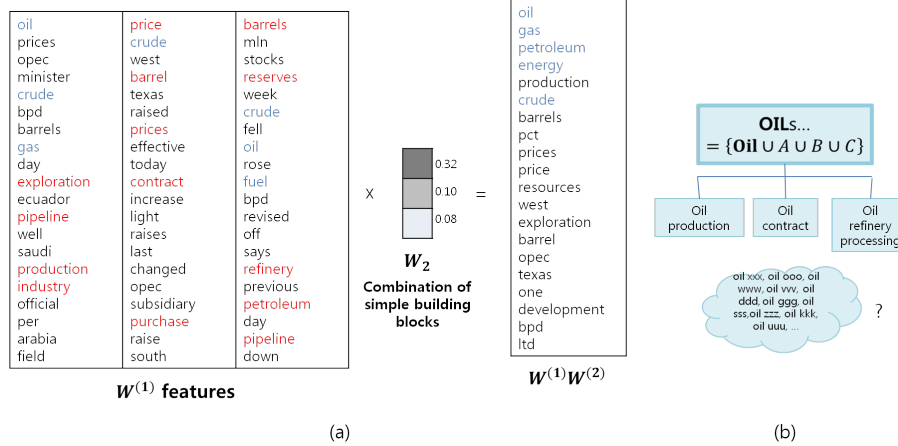


Figure 1: Concept hierarchies in Reuters. (a) Experimental results, and (b) diagram of result in (a).

3 Multi-layer architecture

The proposed hierarchical multi-layer NMF structure comprise of several layers of unit algorithm. We first train each layer separately. We process outcome of each layer $\mathbf{H}^{(l)}$ to get $\mathbf{K}^{(l)}$. $K_{kj}^{(l)} = f\left(\frac{H_{kj}^{(l)}}{M_{kj}^{(l)}}\right)$, where $M_{kj}^{(l)} = \sum_{j'=1}^n \frac{H_{kj'}^{(l)}}{n}$, $f(\cdot)$ is nonlinear function, and l denotes index of layer, $l = 1, 2, \dots, L$. The superscript of each term denotes layer index. Processed data representation of $\mathbf{K}^{(l)}$ is used as input to next layer. Using nsNMF, $\mathbf{K}^{(l)}$ is decomposed into $\mathbf{W}^{(l+1)}$ and $\mathbf{H}^{(l+1)}$: $\mathbf{K}^{(l)} \approx \mathbf{W}^{(l+1)}\mathbf{H}^{(l+1)}$. Then, we use outcome of separate training as initialization, and train the whole network jointly. The cost function for joint training is described: $C = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n (X_{ij} - \sum_{k=1}^l W_{ik}^{(1)} \widetilde{H}_{kj}^{(l)})^2$, where $\widetilde{H}_{kj}^{(l)}$ is the reconstruction of $H_{kj}^{(l)}$, which can be computed via back propagation of errors from the last layer to the l^{th} layer: $\widetilde{\mathbf{H}}^{(L-1)} \approx \mathbf{M}^{(L-1)} \odot f^{-1}\left(\mathbf{W}^{(L)}\widetilde{\mathbf{H}}^{(L)}\right), \dots, \widetilde{\mathbf{H}}^{(1)} \approx \mathbf{M}^{(1)} \odot f^{-1}\left(\mathbf{W}^{(2)}\widetilde{\mathbf{H}}^{(2)}\right)$, where $\widetilde{\mathbf{H}}^{(L)} = \mathbf{H}^{(L)}$. $f^{-1}(\cdot)$ is inverse nonlinear function. (more details on the actual update computation is described in Appendix A). After training until the last layer, final data representation $\mathbf{H}^{(L)}$ is acquired. This is the activation information of complex features, which is the integration of features throughout the layers, $\mathbf{W}^{(1)}\mathbf{W}^{(2)} \dots \mathbf{W}^{(L)}$.

For more detailed explanation, refer to the pseudo-code for the training procedure in Appendix B.

4 Document data feature hierarchies

We applied our proposed network to document database. We used "Reuters-21578 collection, distribution 1.0"¹ as database. We sorted top 10 categories from ModApte split, conducted pre-processing of removing stop-words, and reduced dimension to 1000. There are 5786 and 2587 document samples for training data, and test data. We constructed two-layered network with number of hidden neurons as 160.

We observed how concepts form hierarchies in document data in Figure 1 (a). First, second, and third $\mathbf{W}^{(1)}$ features contain words related to 'oil production' (exploration, pipeline, production, industry), 'oil contract' (contract, purchase, barrel, prices), and 'oil refinery processing' (refinery, reserves, pipeline, petroleum), respectively. These sub-class topic features are combined together and develop into one broader topic 'oil.' With this combination relationship of features, we can figure out that those three seemingly independent features can be re-categorized under the same broader topic. (The

¹The Reuters-21578, Distribution 1.0 test collection is available from David D. Lewis professional home page, currently: <http://www.research.att.com/~lewis>

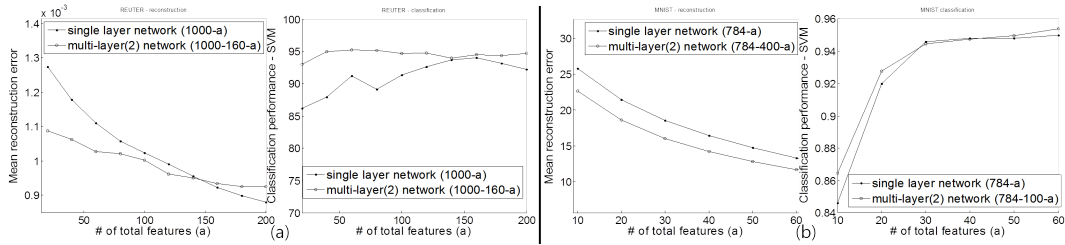


Figure 2: Reconstruction error (*left*) and classification rate (*right*) of (a) Reuters and (b) MNIST.

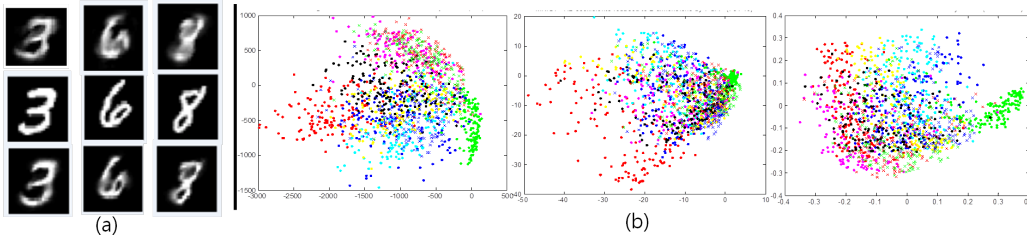


Figure 3: (a) Reconstruction by shallow network (first row) and proposed network (third row), with original input (second row). (b) Final data representation (visualized by PCA) comparison of shallow network (*middle*) and proposed network (*right*), in comparison to raw data (*left*)

concept hierarchy learned in Reuters: sub-categories of 'oil production', 'contract', and 'refinery processing' exist under 'oil' category.)

Furthermore we analyzed reconstruction and classification performance as shown in Figure 2 (a). The proposed hierarchical feature extraction method results in much better classification and reconstruction, especially for small number of features, compared to extracting features at one step. This proves the efficiency and effectiveness of our proposed approach in learning of features.

We also applied our network to handwritten digit image MNIST². The final data representation \mathbf{H}^L displayed distinct activation patterns for samples of the different classes, as a result of successful learning of feature hierarchy, which determines the combination of low level features in forming of distinct class features. In Figure 2 (b), the reconstruction error and classification performance also demonstrate better performance of our proposed method in small number of dimensions. In Figure 3 (a), we can observe sparser and clear reconstruction of our proposed network. The Fisher discriminant values of final data representation of the shallow network and our proposed network were 0.51 and 0.61 respectively. We can infer that proposed network learns more meaningful and helpful features so that it results in better distributed (clustered) representation of data. We can also check this via the visualization of $\mathbf{H}^{(L)}$ to 2-D domain shown in Figure 3 (b).

5 Conclusion

In this paper, we proposed a hierarchical data representation model, hierarchical multi-layer NMF by stacking nsNMF into several layers. We demonstrated hierarchical approach in learning of the features. There are mainly two findings of our research. Taking hierarchical learning by stacking NMFs: 1.reveals intuitive feature hierarchies (subcategories) by learning feature relationships throughout the layers, and 2.learns more meaningful features compared to one-step learning. (as a result, our proposed method results in much better classification and reconstruction performance, provided small number of dimensions for data representation.) We expect our proposed method to be applied to various types of data for discovering underlying feature hierarchies and at the same time, maintain reconstruction and classification performance even with small number of features for data representation.

²Available at: <http://yann.lecun.com/exdb/mnist/>

References

- [1] Bengio, Y. (2007). Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2(1), 1-127.
- [2] Hinton, G. E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 1527-1554.
- [3] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy Layer-Wise Training of Deep Networks. *NIPS*, 153-160.
- [4] Ahn, J., Choi, S., Oh, J. (2004). A Multiplicative Up-Propagation Algorithm. *Proceedings of the 21st International Conference on Machine Learning*.
- [5] Lee, D. D., and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 788-791.
- [6] Pascual-Montano, A., Carazo, J. M., Kochi, K., Lehmann, D., and Pascual-Marqui, R. D. (2006). Non-Smooth Nonnegative Matrix Factorization (nsNMF). *IEEE Trans. Pattern Anal. Machine Intell*, 403-415.

A Appendix: Actual update computation

Continued from Section 3, the actual computation is done as described in (1).

$$W_{ik}^{(l)} \leftarrow W_{ik}^{(l)} \frac{\left(\mathbf{Nu}^{(l)} \widetilde{\mathbf{H}}^{(l)T} \right)_{ik}}{\left(\mathbf{De}^{(l)} \widetilde{\mathbf{H}}^{(l)T} \right)_{ik}}, \text{ and } H_{kj}^{(l)} \leftarrow H_{kj}^{(l)} \frac{\left(\mathbf{W}^{(l)T} \mathbf{Nu}^{(l)} \right)_{kj}}{\left(\mathbf{W}^{(l)T} \mathbf{De}^{(l)} \right)_{kj}}, \text{ where} \quad (1a)$$

$$\mathbf{Nu}^{(l)} = \begin{cases} \mathbf{X} & \text{if } l = 1 \\ \left(\mathbf{W}^{(l-1)T} \mathbf{Nu}^{(l-1)} \right) \odot \left(\mathbf{M}^{(l-1)} f^{-1'} \left(\mathbf{W}^{(l)} \mathbf{H}^{(l)} \right) \right) & \text{otherwise} \end{cases} \quad (1b)$$

$$\mathbf{De}^{(l)} = \begin{cases} \widetilde{\mathbf{X}} & \text{if } l = 1 \\ \left(\mathbf{W}^{(l-1)T} \mathbf{De}^{(l-1)} \right) \odot \left(\mathbf{M}^{(l-1)} f^{-1'} \left(\mathbf{W}^{(l)} \mathbf{H}^{(l)} \right) \right) & \text{otherwise} \end{cases} \quad (1c)$$

Here, $\widetilde{\mathbf{X}} = \mathbf{W}^{(1)} \widetilde{\mathbf{H}}^{(1)}$. $\widetilde{H}_{kj}^{(l)}$ is the reconstruction of $H_{kj}^{(l)}$, which can be computed via back propagation of errors from the last layer to the l^{th} layer as shown in (2).

$$\widetilde{\mathbf{H}}^{(l)} = \begin{cases} \mathbf{H}^{(l)} & \text{if } l = L \\ \mathbf{M}^{(l)} \odot f^{-1} \left(\mathbf{W}^{(l+1)} \widetilde{\mathbf{H}}^{(l+1)} \right) & \text{if } l = L - 1, \dots, 1 \end{cases} \quad (2)$$

$\mathbf{M}^{(l)}$ is a matrix of column-wise mean of $\mathbf{H}^{(l)}$, and $f^{-1}(\cdot)$ is inverse nonlinear function.

B Appendix: Pseudo-code for training procedure of proposed network

```

%% Separate training of layers in extending mode
for l = 1 : L do
  Randomly initialize  $\mathbf{W}^{(l)}$  and  $\mathbf{H}^{(l)}$ 
  if l = 1 then
     $\mathbf{K}^{(l-1)} = \mathbf{X}$ 
  end if
  for iteration = 1 : (untilconvergence) do
     $W_{ik}^{(l)} \leftarrow W_{ik}^{(l)} \frac{(\mathbf{K}^{(l-1)} \mathbf{H}^{(l)T})_{ik}}{(\mathbf{W}^{(l)} \mathbf{H}^{(l)} \mathbf{H}^{(l)T})_{ik}}$ 
     $H_{kj}^{(l)} \leftarrow H_{kj}^{(l)} \frac{(\mathbf{W}^{(l)T} \mathbf{K}^{(l-1)})_{kj}}{(\mathbf{W}^{(l)T} \mathbf{W}^{(l)} \mathbf{H}^{(l)})_{kj}}$ 
  end for
   $M_{kj}^{(l)} = \sum_{j'=1}^n H_{kj'}^{(l)} / n$ 
   $K_{kj}^{(l)} = f \left( \frac{H_{kj}^{(l)}}{M_{kj}^{(l)}} \right)$ 
end for

%% Joint training the whole network
Use  $\mathbf{W}^{(l)}$  and  $\mathbf{H}^{(l)}$ , and use  $\mathbf{M}^{(l)}$  acquired from above
for iteration = 1 : (untilconvergence) do
  for l = 1 : L do
    if l = L then
       $\widetilde{\mathbf{H}}^{(l)} = \mathbf{H}^{(l)}$ 
    else
       $\widetilde{\mathbf{H}}^{(l)} = \mathbf{M}^{(l)} \odot f^{-1} \left( \mathbf{W}^{(l+1)} \widetilde{\mathbf{H}}^{(l+1)} \right)$ 
      which can be written in full length as:
       $\widetilde{\mathbf{H}}^{(l)} = \mathbf{M}^{(l)} \odot f^{-1} \left( \mathbf{W}^{(l+1)} \left( \mathbf{M}^{(l+1)} \odot f^{-1} \left( \mathbf{W}^{(l+2)} \left( \dots \left( \mathbf{M}^{(L-1)} \odot f^{-1} \left( \mathbf{W}^{(L)} \mathbf{H}^{(L)} \right) \right) \right) \right) \right) \right)$ 
    end if
    if l = 1 then
       $\widetilde{\mathbf{N}}\mathbf{u}^{(l)} = \widetilde{\mathbf{X}}$ 
       $\widetilde{\mathbf{D}}\mathbf{e}^{(l)} = \widetilde{\mathbf{X}}$ 
      which can be written in full length as:
       $\widetilde{\mathbf{X}} = \mathbf{W}^{(1)} \widetilde{\mathbf{H}}^{(1)} = \mathbf{W}^{(1)} \left( \mathbf{M}^{(1)} \odot f^{-1} \left( \mathbf{W}^{(2)} \left( \mathbf{M}^{(2)} \odot f^{-1} \left( \mathbf{W}^{(3)} \left( \dots \right) \right) \right) \right) \odot f^{-1} \left( \mathbf{W}^{(L)} \mathbf{H}^{(L)} \right) \right)$ 
    else
       $\mathbf{N}\mathbf{u}^{(l)} = \left( \mathbf{W}^{(l-1)T} \mathbf{N}\mathbf{u}^{(l-1)} \right) \odot \left( \mathbf{M}^{(l-1)} f^{-1'} \left( \mathbf{W}^{(l)} \mathbf{H}^{(l)} \right) \right)$ 
       $\mathbf{D}\mathbf{e}^{(l)} = \left( \mathbf{W}^{(l-1)T} \mathbf{D}\mathbf{e}^{(l-1)} \right) \odot \left( \mathbf{M}^{(l-1)} f^{-1'} \left( \mathbf{W}^{(l)} \mathbf{H}^{(l)} \right) \right)$ 
    end if
     $W_{ik}^{(l)} \leftarrow W_{ik}^{(l)} \frac{(\mathbf{N}\mathbf{u}^{(l)} \widetilde{\mathbf{H}}^{(l)T})_{ik}}{(\mathbf{D}\mathbf{e}^{(l)} \mathbf{H}^{(l)T})_{ik}}$ 
     $H_{kj}^{(l)} \leftarrow H_{kj}^{(l)} \frac{(\mathbf{W}^{(l)T} \mathbf{N}\mathbf{u}^{(l)})_{kj}}{(\mathbf{W}^{(l)T} \mathbf{D}\mathbf{e}^{(l)})_{kj}}$ 
  end for
end for

```