Hierarchical Topic Modelling
Mathematics Senior Thesis

Ziv Epstein
ziv.epstein@pomona.edu

November 5, 2016

Contents

1 Introduction to Topic Modeling

With the vast amount of digital text being generated across the internet, methods for understanding and processing corpora of human language become necessary. Across mathematics and computer science, many techniques have been put forward that allow one to understand a body of text far too large to read herself. A successful method in this domain is *topic modelling*, whereby semantically cohesive subgroups of words can be identified. In particular, let $\mathcal{C} = \{d_1, d_2, \cdots, d_n\}$ be a collection of documents with a vocabulary $\mathcal{V}$. A *topic* t_i is a vector over the words in the vocabulary that represents a coherent high level notion in the corpus:

$$t_i = \{v_1^i, v_2^i, \cdots, v_m^i\}$$

where m is the size of the vocabulary. Topic modelling offers a powerful tool for understanding large amounts of text because they can discover latent semantic structure within text.

There are two primary techniques for learning these topics t_i. The first is LDA, a generative Bayesian statistical model which views each document d_j as a mixture of various topics. The second is non-negative matrix factorization, which aims to factor the document/word matrix into a document/topic and a topic/word matrix [6]. The focus of this thesis will be NNMF, because of its relation to linear algebra, and its deep visual and conceptual intuition.

talk about high level structure of paper

2 Non Negative Matrix Factorization

2.1 How NNMF solves Topic Modeling

Non-negative matrix factorization was first employed by Paatero and Tapper [8] but was made popular by Lee and Seung [6] who suggested the importance of non-negative in human percetion and first linked it to topic modeling.

Given our n documents with vocabulary $\mathcal{V}$ of size m, we construct a matrix $X \in \mathbb{R}^{n \times m}$ where $X_{i,j}$ is the number of occurences of word j in document i. For a given inner dimension k, we seek to factor X into two matrices A and S such that

$$X \approx AS$$

where $A \in \mathbb{R}^{n \times k}$ is the document/topic matrix and $S \in \mathbb{R}^{k \times m}$ is the topic/word matrix. When we impose that A and S must be non-negative, a strong intuition emerges. In particular, the (i,j)th entry of A corresponds to the proportion of topic j in document i and the (i,j)th entry of S corresponds to the relevance of word j in topic i. The problem of finding such a factorization can be formulated as finding a non-negative A and S that minimize the error

$$F = ||X - AS||^2 \tag{1}$$

While this optization problem is not convex in both A and S, it is convex in one of them. So for a given, fixed S, we can find the optimal A by setting the gradient equal to zero. Since

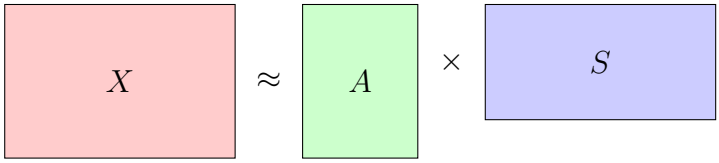

Figure 1: Figure 1: A visual representation of the non-negative matrix factorization

$||X - AS||^2 = \langle X - AS, X - AS\rangle = X^TX - 2X^TAS + (AS)^T(AS)$ we have

$$\frac{\partial F}{\partial A}(X^TX - 2X^TAS + (AS)^T(AS)) = 0$$
$$\text{implies } S^TAS = 2X^TS$$

which is to say $\frac{X^TS}{S^TAS} = 1$ at the optimal A. This equality gives us the below multiplicative update algorithm.

Input: k=0; Initialize A^0, S^0

repeat

$$A^{k+1} = A^k \circ \frac{XS^k}{A^k(S^k)^TA^k}$$
$$S^{k+1} = S^k \circ \frac{XA^{k+1}}{S^k(A^{k+1})^TA^{k+1}}$$
$$k = k + 1$$

until *Stopping condition*;

Algorithm 1: Multiplicative Update

This optimization scheme naturally leads to a convex optimization function, so the above algorithm can simply be iteratively applied (until for given T we have $k > T$ or for a given $\epsilon > 0$ we have $||X - AS||^2 \leq \epsilon$).

Theorem 1. *The Euclidean distance* $||X - AS||^2$ *is non-increasing under the updating rules of Algorithm 1.*

2.2 Hierarchical Topic Modelling

We then consider the work of Griffiths and Tenenbaum [3], who extened the notion of topic models to a hierarchical domain. We aim to replicate this structure but using an NNMF implementation instead of Latent Dirlichet Allocation (LDA).

3 Algorithms

We then discuss methods for using NNMF to generate a hierarchical topic model.

3.1 Single Linkage Graph Construction

For a given document matrix V, we use the python library `scikitlearn` to decompose V into document/topic matrix W and topic/word matrix H such that

$$V \approx WH.$$

The `scikitlearn` implementation uses alternating gradient descent with the following objective function to generate optimal guesses for W and H.

$$c(H, W) = \tfrac{1}{2}||X - WH||^2_{fro} + \alpha\lambda||W||_1 + \alpha\lambda||H||_1 + \tfrac{1}{2}\alpha(1-\lambda)||W||^2_{fro} + \tfrac{1}{2}\alpha(1-\lambda)||H||^2_{fro}$$

where $||\cdot||_{fro}$ is the Frobenius norm, $||\cdot||_1$ is the L1 norm, λ is the L1 ratio and α is a free parameter.

From the N topics t_n for $n \in \{1 \cdots N\}$[1], we populate an adjacency matrix A where

$$A_{i,j} = \frac{T_i \cdot T_j}{||T_i||\ ||T_j||}$$

is the cosine similarity between topics i and j. We then define a *threshold vector* σ by sorting all the elements of A.

$$\sigma = \{\sigma_1, \sigma_2, \cdots \sigma_{N^2} \mid 0 \leq \sigma_i \leq \sigma_j \leq 1 \forall i \leq j \text{ and } \sigma_k \in A\}$$

We then create an array of graphs $A^{(k)}$ thresholded using the values of σ, such that

$$A^{(k)}_{i,j} = \begin{cases} 1 & \text{if } A_{i,j} > \sigma_k \\ 0 & \text{otherwise.} \end{cases}$$

Observe that $A^{(1)}$ is the fully connected graph and $A^{(N^2)}$ is the completely disconnnected graph. By looking at the connected components of a given graph,

$$c(A^{(j)}) = \{c^j_1, c^j_2, \cdots, c^j_i, \cdots, c^j_N\}$$

where $c_i = k$ means that the ith vertex is in the kth order component, we can formulate a tree structure (see Figure 1). For example, say $N = 8$ and we have

$$c(A^{(j)}) = \{0, 0, 0, 0, 1, 1, 1, 1\}$$
$$c(A^{(j+1)}) = \{0, 0, 0, 0, 1, 1, 2, 2\}$$

This means that $A^{(j)}$ has two connected components, ordered 0 (with vertices 1,2,3,4) and 1 (with vertices 5,6,7,8) and that $A^{(j+1)}$ has three connected components, ordered 0 (with vertices 1,2,3,4), 1 (with vertices 5 and 6) and 2 (with vertices 7 and 8). Thus there is a branch from the connect component 1 in $A^{(j)}$ to the connected componenets 1 and 2 in $A^{(j+1)}$. By greedily repeating this iterative algorithm starting with $A^{(1)}$ [2] as the root, we produce the tree of topics. Observe that at this stage, all the leaf nodes correspond to actual topics t_n. We formulate the topic vectors for the parent nodes by additive percolating up the tree. That is, for a given parent topic τ with children $\tau_1, \cdots, \tau_k$ we simply have

$$\tau = \sum_i \tau_i$$

[1] observe that t_n is simply the nth row of H

[2] which has by definition only a single connected component and so $c(A^{(1)}) = \{0, \cdots, 0\}$

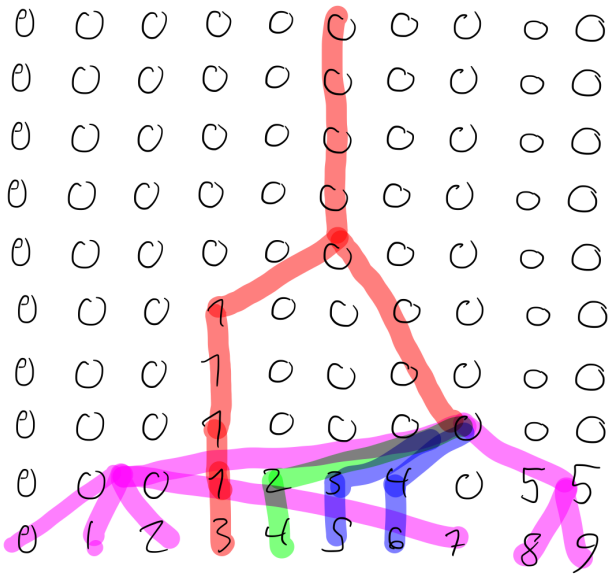

Figure 2: How the tree structure is formed for the connected componenet vectors

3.2 Semi-Supervised NNMF

In many cases, in addition to $X \in \mathbb{R}^{n \times m}$ we also have a label matrix $Y \in \mathbb{R}^{n \times k}$, where k is the number of cases and $Y_{i,j}$ is 1 if document i is in class j and 0 otherwise. When the data is labeled in this way, we can incorporate this information into the NNMF model to learn a more robust representation of the data. This *semi-supervised* NNMF learns a one-versus all separating hyperplane for the observations [7]. Given $B \in \mathbb{R}^{k \times r}$, a basis matrix for Y, and $L \in \mathbb{R}^{k \times n}$, a weight matrix to handle missing labels, then the energy function for SSNMF is as follows

$$E - ||(X - AS)||^2 + \lambda ||L \circ (Y - BS)||^2$$

where λ is a tradeoff parameter that governs the importance of the supervised term.

In the same vein of Algorithm 1, this energy function yields a convex optimization problem solved by the following algorithm.

Input: k=0; Initialize A^0, S^0, B^0
repeat

$$A^{k+1} = A^k \circ \frac{XS^k}{A^k(S^k)^T S^k}$$
$$B^{k+1} = B^k \circ \frac{(L \circ Y)S^k}{(L \circ B(S^k)^T)S^k}$$
$$S^{k+1} = S^k \circ \frac{(A^{k+1})^T X + \lambda B^T (L \circ Y)}{(A^{k+1})^T AS + \lambda B^T (L \circ BS)}$$
$$k = k + 1$$

until *Stopping condition*;

Algorithm 2: Multiplicative Update for Semi-Supervised NNMF

3.3 Deep Semi NMF

Semi-Supervised NNMF as discussed above can be thought of as representing A in a low dimensional representation as S. In this framework, A is the function that maps the low dimensional representation to the original high dimensional representation. However, as the data becomes increasingly complex, it may have many hierarchy of attributes, each of which requires its own mappings. With the motivation in mind, Trigeorgis et al put forward the notion of a Demi-Smi NNMF [9].

$$X \approx A_1 A_2 \cdots A_m S_m$$

This representation of the data can be achieved by recursively factorizing the low-dimensional representation at each level [9].

$$S_{m-1} = A_m S_m$$
$$\vdots$$
$$S_2 \approx A_3 \cdots A_m S_m$$
$$S_1 \approx A_2 \cdots A_m S_m$$

We then consider the algorithms and notions in the domain of Deep Semi NMF as a model for Hierarchical NMF [1, 9]

4 Visualization

I use the d3.js Sunburst implementation to visualize the hierarchical topic model. Arcs at the same level represent discrete topics. A topic on an inner layer that encompasses multiple outer topics represent a super-topic. For example, in Figure 3, the two outer green topics that represent European banking and Chinese banking representatively (with top words

{banks, loans, collateral, abs, ecb bank, lending, small, european, asset}

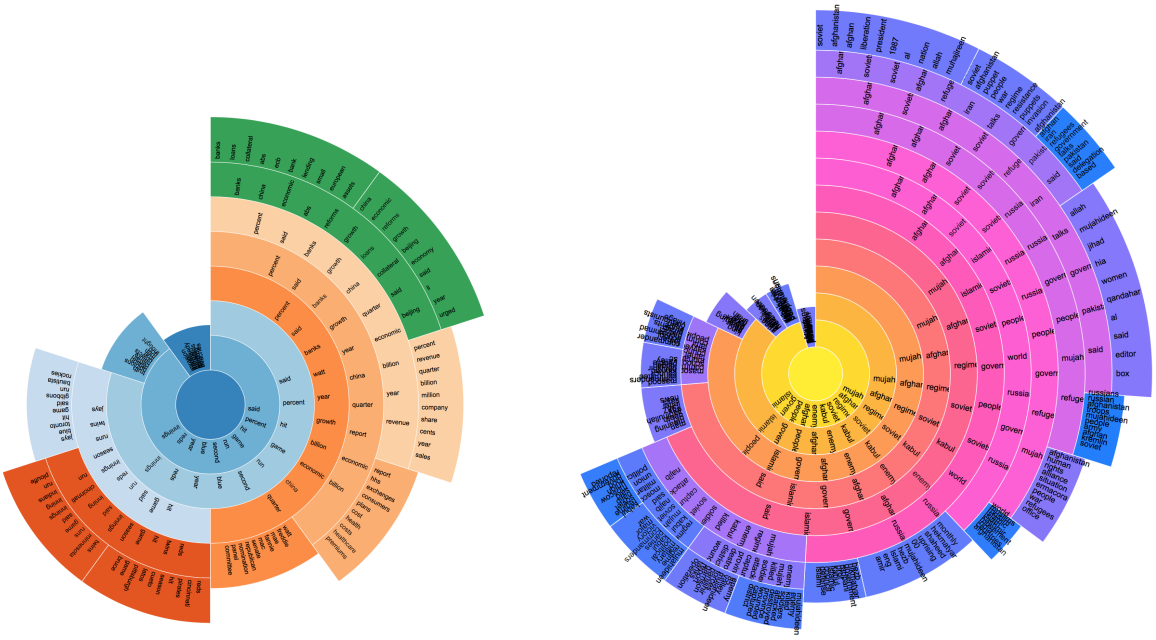

Figure 3: Left: Visualization of hierarchical topics in standard NMF; Right: Visualization of hierarchical topics in SSNMF;

and

{china, economic, reforms, growth, beijing, economy, said, li, year, urged})

merge into the inner super topic with top words

{banks, china, economic, abs, reforms, growth, loans, collateral, said, beijing}.

The visualization is responsive, dynamic and available at
`http://www1.cmc.edu/pages/faculty/BHunter/ziv.html`

Next, I extended this visualization to the Semi-Supervised domain using the Afghan Dataset. Here each document has associated with it a class $C \in \{1, \cdots, k\}$ which in this case $k = 3$. Recall that the matrix $B \in \mathbb{R}^{k \times r}$ in the semi-supervised NMF model is multiplied by S to obtain an approximation for Y, the label matrix. Thus the i, jth entry of B can be interpreted in the weighted importance of topic i in predicting class j. Thus I sum over the columns of B to capture how important topic i is predicting classes in general. After normalizing, we get a color value c_i for topic i, such that

$$c_i = \sum_j B_{i,j} / \sum_{i,j} B_{i,j}$$

where $c_i = 1$ corresponds to yellow and $c_i = 0$ corresponds to blue (see Figure 3 right)

5 Hierarchical Topic Model

We will then construct our own model for learning a hierarchy of topics within the model itself. This will be a blend of the Single Linkage graph construction model and the Deep Semi NMF.

6 Results

We will then apply the method of our Hierarchical Topic Model to several datasets, and compare our results with previous work.

6.1 Synthetic Data

We will generate synthetic data with hierarchical topics and verify that our algorithm succesfully extracts them

6.2 Standard Data

We will run our algorithm on data cannonically associated with this task. For our purposes, the 20 News Group Data set will probably be sufficient.

6.3 Afghan Data

A collaborator at NYU Abu-Dhabi has hand-curated a dataset of Afghani magazines, which is notable for sociologists. We will run our algorithm on this data set and see what happens

7 Discussion/Conlusion

Here we will conclude by situating this work within the field, compare its results with similar methods and propose future work.

References

[1] Jennifer Flenner and Blake Hunter. A deep nonnegative matrix factorization.

[2] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[3] DMBTL Griffiths and MIJJB Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16:17, 2004.

[4] Ngoc-Diep Ho. *Nonnegative matrix factorization algorithms and applications*. PhD thesis, ÉCOLE POLYTECHNIQUE, 2008.

[5] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[6] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[7] Hyekyoung Lee, Jiho Yoo, and Seungjin Choi. Semi-supervised nonnegative matrix factorization. *IEEE Signal Processing Letters*, 17(1):4–7, 2010.

[8] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[9] George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Bjoern Schuller. A deep semi-nmf model for learning hidden representations. In *ICML*, pages 1692–1700, 2014.