Hierarchical Topic Modelling

Ziv Epstein
`ziv.epstein@pomona.edu`

October 1, 2016

Abstract

Here we explore an application of non-negative matrix factorization (NNMF) in topic modelling. In particular, we consider a Hierarchical topic models, where topics are nested in a tree-structure. In this document, we formalize this notion, propose an algorithm, and showcase an visualization engine for this doman.

1 Implementation

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.

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

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

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

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

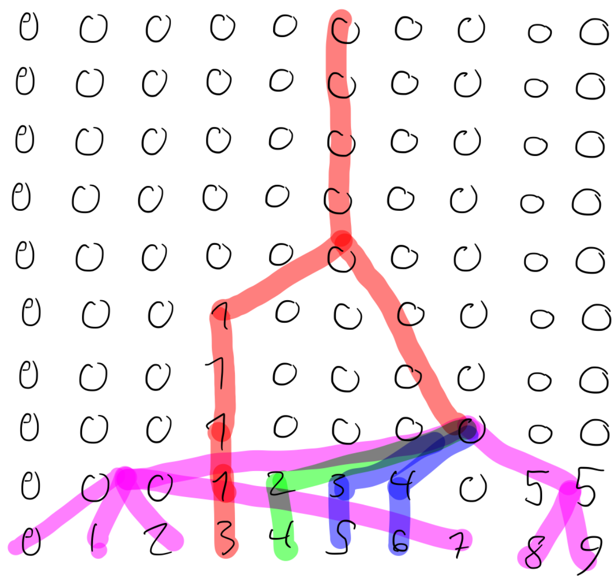

Figure 1: How the tree structure is formed for the connected componenet vectors

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,

$$c(A^{(j)}) = \{c_1^j, c_2^j, \cdots, c_i^j, \cdots, c_N^j\}$$

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

$$\tau = \sum_i \tau_i$$

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

Figure 2: Screen shot of hierarhical topic model application for sample data set