Write up

1 Introduction

Our aim is to develop an algorithm that will effectively analyze and summarize an incoming stream of short texts. In particular, the algorithm intends to

- discover what the topics are at each time instant (this is particularly challenging for short texts, and we will address this issue),
- identify whether the topics are evolving, emerging or fading,
- and tell a story based on all the incidents that take place during a given timeframe.

.

2 Model

In this section, we explain the model used to study our corpora.

2.1 Discovering the topics

Consider a documents-by-words matrix, X of dimension n-by-m. A non-negative factorization on this matrix can reveal what the latent topics are. The matrix X can be factorized as

$$X \approx PQ, \tag{1}$$

where P is a n-by-k matrix of documents-by-topics, and Q is a k-by-m matrix of topics-by-words. k is the number of topics.

When the documents under consideration are very short, X ends up being very sparse. Typically, only about than 4% [4] of all the entries in X are non-zero. This would make learning from X very inefficient.

For a corpus containing very short texts, the size of the vocabulary does not increase very much beyond a certain point [4]. This behavior is a contrasting one when compared with a corpus containing large text documents. Hence, for a corpus with short texts, one can say that all the documents are being generated from a relatively small vocabulary.

Keeping this in mind, let us think about the *word co-occurrence* matrix K. This is an m-by-m matrix built empirically from the co-occurrences of the word pairs. (This matrix could be built from any form of normalized co-occurrence counting, but Pointwise Mutual Information is widely used). It turns out that this matrix K is a fairly dense one. Typically, about 80% of its entries are non-zero. Given that the vocabulary size for short-text corpora is relatively small, it makes sense that the word co-occurrence matrix for such a corpora would be dense. The idea then is to learn the topics from this K matrix by performing a decomposition as below:

$$K \approx Q^T Q. \tag{2}$$

Equation 2 resembles the singular value decomposition, except that we require all the entries to be positive. Such a decomposition is referred to as a *symmetric non-negative* matrix factorization, and is similar to kernel K-means. Here, Q is an k-by-m matrix, where k is the number of topics. Q can be thought of as the *topics matrix*. Each row of Q, after normalization, is the distribution of words in that topic. Once, the topic matrix Q has been learnt, one can infer the P matrix from Equation 1, where X and Q are assumed known.

We can think of Equation 2 as the learning process, and Equation 1 as the inference process.

2.2 Discovering the evolution

Our aim is to know which topics are emerging, which are fading, and which are evolving. To this end, we model the data as an *incoming stream* of data matrices $\{X^t\}_{t=0}^{t=T}$. Between consecutive time steps, we model the change in the topics through an evolution matrix M as below:

$$K^t \approx {Q^t}^T M^t Q^{t-1}. \tag{3}$$

In Equation 3, K^t and Q^{t-1} are known. We can think of Equation 3 as such: the topics of the current time step (Q^t) are learnt from the current data (K^t), and past topics (Q^{t-1}). The product $M^t Q^{t-1}$ can be thought of as an approximation for Q^t. This approximation aims to model the temporal changes in the topics over time through the M^t. Studing the M^t in detail would give us insight about which topics are emerging, evolving or fading.

2.3 The Evolution Matrix: M

Consider the equation:

$$Q^t \approx M^t Q^{t-1}. \tag{4}$$

Equation 4 can be immediately deduced from Equation 3. Let $\mathbf{q}_i^t$ and $\mathbf{q}_i^{t-1}$ be the i^{th} row of $Q^t and Q^{t-1}$. Let m_{ij}^t represent the elements of M^t. Then, we have that

$$\mathbf{q}_i^t \approx \sum_r m_{ir}^t \mathbf{q}_k^{t-1}. \tag{5}$$

Note that all the entries of M^t are positive, since we have enforced all factorizations to be non-negative. From Equation 5 we can see that the i^{th} topic at time t is a weighted combination of the all the topics from time $t-1$. The weights are specified by m_{ir}^t. /

By studying the nature of M^t, the following conlusions can be made:

- if M^t is a permutation matrix, it implies that the topics have not changed very much from the previous time instant to the current.
- If the i^{th} row of M^t is all 0s, it implies that q_i is an emerging topic (since the model is unable to find a way to represent it in terms of the previous topics).
- If the j^{th} column of M^t is $\mathbf{0}$, it implies that topic q_j^{t-1} is fading, since it contributes no weight to the topics of the current time step.

We will utilize these properties of M^t to construct a "story" from the Ebola dataset.

3 Optimization

This section will detail the algorithm that will be used to find the current topics, and infer the document representations - essentially solving Equations 1 and 3. We start with Equation 3. The loss function can be formalized as below:

$$L_t = ||K^t - {Q^t}^T M^t Q^{t-1}||_F^2 + \alpha R. \tag{6}$$

We want to minimize the loss function in Equation 6 with respect to the variables Q^t and M^t. The term R encompasses all the regularization. We apply the following regularization for the variables:

$$R = \alpha(||Q^t||_1 + ||M^t||_1) + \lambda(||M^t - I||_F^2) \tag{7}$$

The $||.||_F$ is the Frobenius norm. The $||.||_1$ is the $l1$ norm. The regularization $||M - I||_F^2$ will decide how close or far from identity M needs to be. The closer to identity, the lesser the topics change over time. Such a loss function is not simultaneously convex in the variables under consideration. Hence, we resort to techniques which minimize the objective to attain the local minimum. One such technique, called multiplicative updates is described in [1]

The update equations to minimize the loss in Equation 6 is summarized below (derivation is straight forward):

$$\begin{aligned} Q^t &\leftarrow Q^t \odot \frac{M^t Q^{t-1} K^t - \alpha \mathbf{e}\mathbf{e}^T}{M^t Q^{t-1} {Q^{t-1}}^T M^t Q^t}, \\ M^t &\leftarrow M^t \odot \frac{Q^t K^t {Q^{t-1}}^T + \lambda I - \alpha \mathbf{e}\mathbf{e}^T}{Q^t {Q^t}^T M^t Q^{t-1} {Q^{t-1}}^T + \lambda M^t}. \end{aligned} \tag{8}$$

For Equation 1, we assume that the data matrix X, and the topics Q are given. The loss function we consider is as below:

$$||X^t - P^t Q^t||_F^2 + \gamma ||P||_F^2. \tag{9}$$

The solution to Equation 9 is just the traditional l-2 regularized least squates solution. The least squares solution involves taking the inverse of a large matrix which is computationally expensive. One can also use update equations as in Equation 8 to minimize the loss in Equation 9 to avoid such inverse computations. The update equations for Equation 9 are as below:

$$P^t \leftarrow P^t \odot \frac{X^t {Q^t}^T}{(Q^t {Q^t}^T + \lambda I) P^t}. \tag{10}$$

4 Notes

The idea takes inspiration from several ideas in literature like [4], [2], and [3]. In particular, the learning from the word cooccurrence matrix has been presented in [4], but they do not incorporte the temporal knowledge. So, their optimization is different.

The analysis of the evolution matrix, M^t has been studied in [2], but their analysis is on a news dataset containing a few hundred words. Our dataset contains about 4 - 5 words per document.

References

[1] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2000.

[2] Carmen K. Vaca, Amin Mantrach, Alejandro Jaimes, and Marco Saerens. A time-based collective factorization for topic discovery and monitoring in news. WWW '14, 2014.

[3] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1445–1456, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[4] Xiaohui Yan, Jiafeng Guo, Shenghua Liu, Xueqi Cheng, and Yanfeng Wang. *Learning Topics in Short Texts by Non-negative Matrix Factorization on Term Correlation Matrix*, chapter 82, pages 749–757.