Efficient Tensor Decomposition with Application to Topic Modelling

Fengyu Cai
Department of Computer Science
Swiss Federal Institute of Technology Lausanne
Lausanne, Switzerland
fengyu.cai@epfl.ch

Armin Eftekhari
Department of Electrical Engineering
Swiss Federal Institute of Technology Lausanne
Lausanne, Switzerland
armin.eftekhari@gmail.com

***Abstract*—Nowadays, extracting part of text as information summarization for a collection of documents has become one of popular tasks in the fields of text mining. In this paper, we are going to present a new model for the task of topic extraction. We build up a three-dimensional tensor to describe the collection of documents. And through the mathematical illustration, we decompose the tensor in order to get the mapping relationship between topics and documents with different solver. Based on the decomposed matrix, we rank topics for each document to find its tag.**

***Index Terms*—Topic Selection, Tensor Decomposition**

Code Repository:
https://github.com/TRUMANCFY/SemesterProject2019

I. Introduction

Document summarization has become an important research area as a intersection of Information Retrieval (IR), Text Mining (TM), and Natural Language Processing (NLP). It helps a lot in the process of textual repository accessing and data indexing. Based on the huge size of data collection, we put document summerization under the scenario of multiple documents, and the summary of documents will be generated as a form of clusters. Therefore, it is one of key issues about how to present each document and the whole document set. On one hand, the modelling should describe the key features about the documents. On the other hand, the modelling should be general and doable for the further operation on the relationship mapping.

In this paper, we will build a tensor model to describe the document set with the dimentions of documents, terms and potential topics, and decompose it to get the relationship between the document and its topic. Referring to the previously published methods, we generated the topics by clustering the similar sentence. Instead of the traditional high-order Singular Value Decomposition (HOSVD) which has been used in the previous work, we try to solve this using other different solvers. And in the future, we may check whether the process of clustering can be combined into the solvers. (This direction have not thought about.)

This paper will be organized as following: in Section II, we will introduce some related work in the field of textual summarization; in Section III, we will describe the establishment of tensor modelling on the document set; in Section IV, we will discuss how we use different solvers with or without constrains to implement tensor decomposition; Section V will describe the experimental setup and analysis of the result; the final section will contain our conclusion and future work.

II. Related Worked

In the past, the most common method for the document labelling is the computation of the *TF-IDF* score. Even nowadays, around 83% percent of digitized recommender systems are still based on this methodologyadd ref. It is very straightforward method and shows a stable performance, therefore, we would like to continue to use this as the scoring method. In 2003, Latent Dirichlet allocation (LDA) has been proposed as a generative statistical method for topic labelling. With Dirichlet distribution as prior, it is able to label the document with the probability that it belongs to a specific class. However, the main disadvantage of this method is that the number of classes should be a given parameter. Also, the textual summarization problem has also been transferred to the maximum coverage problemadd ref. This methods can extract the sentence to cover as much as key information as possible. As the maximum converge problem is NP-hard, some attempt to find near-optimum solution through a greedy approach. Meanwhile, some other try to apply Integer Linear Programming to find a more precise result.

Tensor is able to well represent high dimensional relationship compactly. And tensor decomposition has been used widely in the bi-relationship extraction from multiple dimensions in many applicable areas. In add ref, the researchers applied tensor decomposition to summary the comments-oriented documents. They aimed to extract the sentences containing the highly-biased keywords from the collection of comments, based on the rank result from tensor decomposition.

In addition, add ref provides us the heuristics inspiration about the setup of the tensor. Instead of finding the topics of the document, they relaxed the constrain to find some of the sentences as the summary in a collection of multi-topic documents. They clustered the sentences first and generated the topics. And then, they built up a tensor with dimensions of terms, topics and documents, and got the ranking of the topics from the decomposition. After that, they selected the sentences out of the clusters based on the order of the topics until reach the maximum length of terms.

In both of two papers mentioned above, they both decomposed the tensor with the method of High Order Singular Value Decomposition (HOSVD). In our project, we intended to use different solver which can provide us more freedom in constrain setting, like Augmented Lagrangian Methods (ALM) or Alternating Direction Method of Multipliers (ADMM). Also, besides the Lagrangian family of methods, we also use get idea from coordinate descent of the non-negative matrix factorization. The previous work [add ref] provides us the existing mathematical tools to simplify our work.

III. Our Methods

A. Problem Setting

The main goal of our task to label the document with the topic. Without knowing the topic set, we generate topics first from the set of sentences, based on the assumption that the high dimensional topics generated from sentences is likely to be similar with the topics generated from the documents. After that, we match the document to its closest generated topic, which is most semantically related.

B. Overviews

1) Pre-processing: Our pre-processing method follows the standard procedure of the textual data pre-processing, including tokenization, stop-word removal, and stemming.

2) Generate Topics: We split the documents into sentences. And we clustered the sentences to find out the potential topics and take them as the document topics.

3) Tensor Setup: Given the generated topics, we built up a three-dimensional tensor representing terms, topics and documents.

4) Tensor Decomposition: After the setup of the tensor, we get the relationship between topics and terms using different solvers, including the coordinate descent methods and the constrained optimization methods like augmented Lagrangian Methods.

C. Topic Generation

Just Current Methods, will be improved later

Generally speaking, we have a basic assumption that the topics of document is very similar to the topics generated from the sentence. Therefore, we firstly tokenized the document, removed the stopwords, and stemmed the words to the original form. Based on the pre-set number of topics, we use the Latent Dirichlet allocation to generate the topics from the collection of reformed sentences. And also, in this process, we can also get the collections of sentences belonging to the specific topic(s).

D. Tensor Representation Model

Our tensor contains three dimensions. The first dimension of tensor represents the set of terms $T_1 ... T_m$, the second dimension represents the cluster of topics $C_1 ... C_k$, and the third dimension presents the collection of documents $D_1 ... D_n$. Therefore, each entry $\mathbf{T}_{i,j,k}$ in the tensor contains the information of $i-th$ term related to $j-th$ topic in the $k-th$ document. Based on the last step *Topic Generation*, we can find out that some words belong to the specific topics, therefore, it is an advantage that we only need to consider these kind of entries instead of the whole tensor.

$$\mathbf{T}_{i,j,k} = \begin{cases} tf-idf_{i,j,k} & T_i \in \mathrm{C}_j \text{ and } T_i \in \mathrm{D}_k \\ 0 & \text{otherwise} \end{cases}$$

There will be a figure to show the tensor representative

E. Tensor Decomposition Methods

After getting the tensor, we tried to decompose it to get the relational mapping between documents and topics. Generally speaking, we categorized the methods to unconstrained and constrained problem.

The process of tensor decomposition could be interpreted to find three matrices U, V and W, such that

$$\mathbf{T}_{i,j,k} \approx \mathrm{U}_{i,r} \times \mathrm{V}_{j,r} \times \mathrm{W}_{k,r}$$

Therefore, we can let

$$\mathcal{L}(\mathbf{T}; \mathrm{U}, \mathrm{V}, \mathrm{W}) = \frac{1}{2} \sum_{i,j,k}^{I,J,K} (\mathbf{T}_{ijk} - \sum_{r=1}^{R} \mathrm{U}_{ir} \times \mathrm{V}_{jr} \times \mathrm{W}_{kr})$$

to be the loss function. Here, R denote the rank of the reconstructed tensor,I denotes the total number of documents, J denotes the total number of topics, and K denotes the total number of terms. Therefore, U contains the information about documents, V contains the information about topics, and W will be related to terms. In order to get the matrix representing the relationship between documents and topics, we set the rank R equal to the number of topics. Our task will be transfer to

$$\min_{U,V,W} \mathcal{L}(\mathbf{T}; \mathrm{U}, \mathrm{V}, \mathrm{W})$$

and we will mainly focus on our target U and add constrain on it in the further.

1) Alternative Gradient Descent:
As we know, the gradient descent method is described as followed:

$$u^{t+1} = u^t - \alpha \frac{\partial \mathcal{L}}{\partial u}(u^t; x^t)$$

if the loss function is $\mathcal{L}(x; u)$, where t is the step of iteration and α is the learning rate.

The main difficulty of this problem is to get the gradient. The paper (add ref) provides the mathematical tools to derive the gradient. Firstly, we define a matrix $\mathbf{T}(\cdot, \mathrm{V}, \mathrm{W}) \in \mathrm{Real}^{I\times R}$, from $\mathbf{T}$, V and W:

$$\mathbf{T}(\cdot,\mathrm{V},\mathrm{W})_{ir} := \sum_{jk} \mathbf{T}_{ijk}\mathrm{V}_{jr}\mathrm{W}_{kr}$$

Therefore, according to **Theorem 1** mentioned in the paper, CP gradient of loss function can be written as:

$$\frac{\partial\mathcal{L}}{\partial \mathrm{U}}(\mathbf{T};\mathrm{U},\mathrm{V},\mathrm{W}) = -\mathbf{T}(\cdot,\mathrm{V},\mathrm{W}) + \mathrm{U}\Gamma((V),\mathrm{W})$$

where $\Gamma(A,B)$ is the Hadamard product of A^TA and B^TB, i.e., $\Gamma(A,B)_{ij} = (A^TA)_{ij}(B^TB)_{ij}$. And the formulas of $\partial\mathcal{L}/\partial\mathbf{V}$ and $\partial\mathcal{L}/\partial\mathbf{W}$.

Therefore, we can update U, V and W iteratively, setting two of them as constant variable and update the other. also cyclic variant version: use U^{t+1} to update V^{t+1} and W^{t+1}

$$\mathrm{U}^{t+1} = \mathrm{U}^{t} - \alpha\frac{\partial\mathcal{L}}{\partial \mathrm{U}}(\mathbf{T};\mathrm{U}^t,\mathrm{V}^t,\mathrm{W}^t)$$

$$\mathrm{V}^{t+1} = \mathrm{V}^{t} - \alpha\frac{\partial\mathcal{L}}{\partial \mathrm{V}}(\mathbf{T};\mathrm{U}^t,\mathrm{V}^t,\mathrm{W}^t)$$

$$\mathrm{W}^{t+1} = \mathrm{W}^{t} - \alpha\frac{\partial\mathcal{L}}{\partial \mathrm{W}}(\mathbf{T};\mathrm{U}^t,\mathrm{V}^t,\mathrm{W}^t)$$

2) Alternative Gradient Descent with regularization:

However, as the optimization is non-convex and 'bilinear', it may be stuck in ill-condition. There exit a lot of local minimum due to the scale of three values. For example, $\mathcal{L}(\mathbf{T};\mathrm{U},\mathrm{V},\mathrm{W}) = \mathcal{L}(\mathbf{T};\mathrm{aU},\mathrm{bV},\mathrm{W/ab})$

Therefore, we add the regularization terms to the loss function:

$$\mathcal{L}_\rho(\mathbf{T};\mathrm{U},\mathrm{V},\mathrm{W}) = \mathcal{L}(\mathbf{T};\mathrm{U},\mathrm{V},\mathrm{W}) + \frac{1}{2}(\|\mathrm{U}\|^2 + \|\mathrm{V}\|^2 + \|\mathrm{W}\|^2)$$

The gradient of regularized loss function is very similar to the original one:

$$\frac{\partial\mathcal{L}_\rho}{\partial \mathrm{U}}(\mathbf{T};\mathrm{U},\mathrm{V},\mathrm{W}) = -\mathbf{T}(\cdot,\mathrm{V},\mathrm{W}) + \mathrm{U}\Gamma_\rho((V),\mathrm{W})$$

where $\Gamma_\rho(A,B)_{ij} = \Gamma(A,B) + \rho I$. Also $\partial\mathcal{L}/\partial\mathrm{W}$ and $\partial\mathcal{L}/\partial\mathrm{W}$ are in the similar formulas. And the process of iterative gradient descent can be referred to last non-regularized part.

3) Second-order Gradient Descent:

Compared with the first-order gradient descent, the second-order is able to converge in fewer iterations. may need ref The format is defined as:

$$u^{t+1} = u^t - \alpha\mathcal{H}^t\frac{\partial\mathcal{L}}{\partial u}(u^t;x^t)$$

where H is the approximate inverse Hessian matrix.

Based on the derivative (may need more explanation and references, when increasing the rank, it does not work), we could update U in the following formula:

$$\begin{aligned}\mathrm{U}^{t+1} &= \mathrm{U}^t - \alpha\frac{\partial\mathcal{L}}{\partial \mathrm{U}}\Gamma_\rho(\mathrm{V}^t,\mathrm{W}^t)^{-1} \\ &= \mathrm{U}^t(1-\alpha) + \alpha\mathbf{T}(\cdot,\mathrm{V}^t,\mathrm{W}^t)\Gamma_\rho(\mathrm{V}^t,\mathrm{W}^t)^{-1}\end{aligned}$$

4) Linearized Augmented Lagrangian Method (LALM):

Till now, three methods mentioned above are all non-constrained. Therefore, we can add some constrain to our optimization problem. For example, we could let the entry U_{ir} to be the probability that the topic of *i-th* document is the *r-th* topic. Therefore, the following constrain should be satisfied for *i-th* row in U:

$$\sum_{r=1}^{R}\mathrm{U}_{ij} = 1$$

If we vectorize the equation above, we set up the constrain:

$$\mathrm{U}\mathbb{1} = \mathbb{1}$$

where $\mathbb{1}$ is a vector filled by one with suitable dimension.

And then, we can introduce Lagrangian multiplier $\mathcal{Y}$ to combine the original loss function and the constrain. Also, in order to guarantee the smoothness of the dual function, we can add the penalty term $\frac{\mu}{2}\|\mathrm{U}\mathbb{1}-\mathbb{1}\|^2$ into the new loss function. Therefore, considering two additional terms, we define the loss function of ALM as followed:

$$\mathcal{L}_\mu(\mathbf{T};\mathrm{U},\mathrm{V},\mathrm{W},\mathcal{Y}) = \mathcal{L}(\mathbf{T};\mathrm{U},\mathrm{V},\mathrm{W}) + \langle\mathcal{Y},\mathrm{U}\mathbb{1}-\mathbb{1}\rangle + \frac{\mu}{2}\|\mathrm{U}\mathbb{1}-\mathbb{1}\|^2$$

According to the procedure of Lagragian method, given a Y, we need to find out the optimized U:

$$\mathrm{U}^{t+1} = \min_{\mathrm{U}}\mathcal{L}_\mu(\mathbf{T};\mathrm{U}^t,\mathrm{V}^t,\mathrm{W}^t,\mathcal{Y}^t)$$

However, it is very difficult to get U^{t+1} directly, therefore, we need to linearize it:

$$\begin{aligned}\mathrm{U}^{t+1} &= \min_{\mathrm{U}}\mathcal{L}_\mu(\mathbf{T};\mathrm{U}^t,\mathrm{V}^t,\mathrm{W}^t,\mathcal{Y}^t) \\ &\leq \min_{\mathrm{U}}\mathcal{L}(\mathbf{T};\mathrm{U}^t,\mathrm{V}^t,\mathrm{W}^t) + \frac{\mu}{2}\|\mathrm{U}^t\mathbb{1}-\mathbb{1}\|^2 \\ &\quad + \mu\langle(\mathrm{U}^t\mathbb{1}-\mathbb{1})\mathbb{1}^T,\mathrm{U}-\mathrm{U}^t\rangle \\ &\quad + \langle\mathcal{Y}^t,\mathrm{U}-\mathrm{U}^t\rangle + \frac{\mu}{2}\|\mathrm{U}-\mathrm{U}^t\|^2 \\ &= \mathbf{F}_{\mu,\mathcal{Y}^t}(\mathrm{U})\end{aligned}$$

In order to find the optimized U in each step, we let $\frac{\partial\mathbf{F}}{\partial\mathrm{U}} = 0$. In the other words,

$$\begin{aligned}\frac{\partial\mathbf{F}}{\partial\mathrm{U}} &= -\mathbf{T}(\cdot,\mathrm{U}^t,\mathrm{V}^t) + \mathrm{U}\Gamma(\mathrm{V}^t,\mathrm{W}^t) \\ &\quad + \mu(\mathrm{U}-\mathrm{U}^t) + \mu(\mathrm{U}^t\mathbb{1}-\mathbb{1})\mathbb{1}^T + \mathcal{Y} = 0\end{aligned}$$

Therefore, we can update U as followed:

$$\mathbf{U}^{t+1} = \frac{1}{\mu}(\mathbf{T}(\cdot,\mathrm{U}^t,\mathrm{V}^t) - \mathcal{Y} - \mu(\mathrm{U}^t\mathbb{1}-\mathbb{1})\mathbb{1}^T + \mu\mathrm{U}^t)(I + \Gamma(\mathrm{V}^t,\mathrm{W}^t))^{-1}$$

And then, like general Lagrangian methods, we do the gradient ascent on $\mathcal{Y}$ to update it.