

Towards a Temporal Latent Dirichlet Allocation Model

Kristopher W. Reese*, Patrick Shafto†

*Computer Science and Engineering, †Computational Cognitive Sciences

University of Louisville

Louisville, KY

{kwrees02, shafto}@louisville.edu

Abstract—Though most topics remain stationary over an infinitely long period of time, there are examples where topics and words within topics change. Latent Dirichlet Allocation (LDA) is used to capture information about topic models given a known number of Topics. This algorithm does not capture information about topics, which may temporally change. The proposed model in this paper attempts to modify the existing LDA model to allow it to capture temporal changes in topics while allowing those models that do not change to remain through infinite time. This is achieved by adding a variable to the existing model, K , and using this variable to calculate the probability of a change in the topic given the hyperparameters, two topics, and the words that make up the topics.

Keywords-Topic Model; Latent Dirichlet Allocation (LDA); Temporal Topic Models

I. INTRODUCTION

The Latent Dirichlet Allocation Model provides a mathematical framework for determining the topics of a document. This model proves to be a useful tool with possibilities for building future semantic web technologies and Information Retrieval Systems. One of the first things people do when attempting to determine whether a document is relevant to their current fields of interest. In scientific research, this is often done while reading the abstract of a paper [1]; however, even in other settings, people continuously approach reading documents by analyzing whether a document has any relevance to the topic in which they are interested in.

The Latent Dirichlet Allocation (LDA) Model is an often employed model to analyze document topics in an unsupervised manner. [1]–[4]. The model has proven extremely effective in determining the document of topics based on the frequency and occurrences of words, and has proven extremely versatile in determining polysemy in words. [3] Polysemy is simply the capturing of words which are the same but have different meanings. For example, the word belt may refer to a belt buckle in the case of clothing related topics and an asteroid belt in the case of astronomical topics. This flexibility would make it an invaluable tool in information retrieval systems and other semantic related technologies.

Despite how powerful LDA is in these technologies, LDA is not without its limitations. The primary limitation that this paper is focused with is its limitation to capture information about related topics which may be separated by a temporal

split. Humans have an ability to find these related temporal topics and be able to cluster them together despite possible extreme changes in the topic over time. One example of this might be a topic of medicine. In a medical document from the early 19th century we might expect to find a discussion of bleeding or leeches as a cure for medical diagnoses. However, we would not expect to find these topics in medical documents in today’s era. We would expect to find information about current medicines or information about various surgeries in exchange for these antiquated topics. Despite this change in topics, we are able to look at both and distinguish that these are both medically related documents.

This paper attempts to offer a model that pushes the research in LDA towards a Temporal LDA. Section 2 of this document will discuss the model in a general sense as well as discuss the mathematical reasoning behind the model. Section 3 will present an experiment using TLDA and the results that we received from the experiment. Section 4 concludes the paper by analyzing the results from the perspective of the possibilities for the algorithm from the experiments discussed in section 3. This section also discussion on future directions that could be taken with the Temporal LDA (TLDA).

II. TEMPORAL LATENT DIRICHLET ALLOCATION

The Temporal LDA model that we will discuss in this section is an extension of the LDA model and as such, an understanding of the this model is a precursor to extending the model to capture Temporal differences. The LDA model consist of a set of documents, D which contain a set of words w . Each of these words is assigned to a topic, z in that document. LDA also sets each word individually to a global topic, T , that lies outside of the document scope. Both the topic for a document and the global topics are assumed to be distributed using some type of distribution, θ and ϕ respectively. These distributions we assume to not know and are integrate out in the model. Instead we attempt use a set of hyperparameters, α and β in the model. Figure 1 shows the model visually.

The Temporal LDA model extends this by adding a K value which we assume overlaps all of the documents and all of the topics from the traditional LDA model. Figure 2

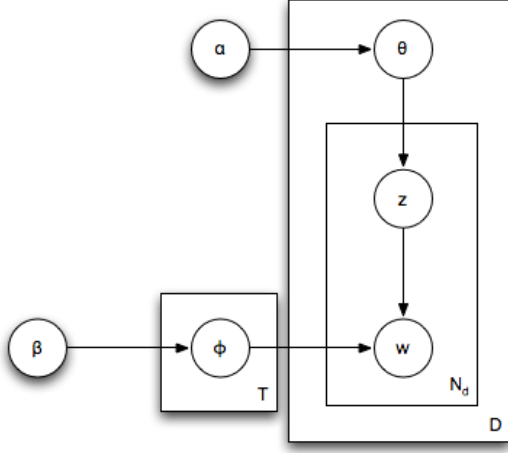


Figure 1. Latent Dirichlet Allocation Model

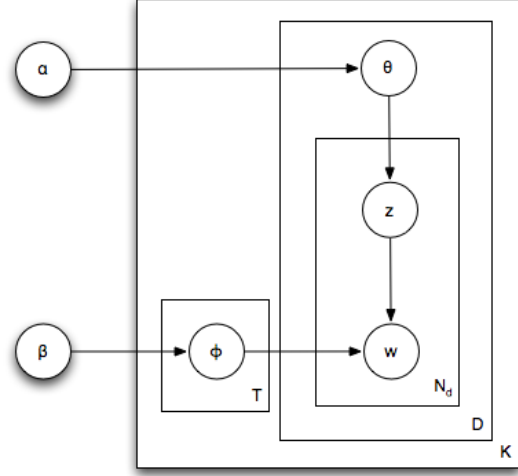


Figure 2. Temporal Latent Dirichlet Allocation Model

shows the Temporal LDA model visually. By looking at this model visually we can see that both the LDA and the TLDA models share primary characteristic with one another. This distinguishing feature in TLDA, K , we can think of as a vertical split in the LDA model, which is our temporal split in the document list. The limitation that the TLDA model faces is that the list must be sorted in temporal order. If the list does not contain dates or it's unknown if the documents are sorted in temporal order, the TLDA model will not be able to give accurate results and any temporal split that the TLDA model might find cannot be proven to be an accurate split in the document list.

Both figures 1 and 2 show the model in a more general sense. It is the mathematics behind the models that allow the models to infer topics in the document based on a set of hyperparameters and word frequency counts in the documents. Since TLDA is an extension, we will again look at LDA first to understand how we determine the topics of words in a document. The primary information that LDA attempts to determine is the topic of the word in the document. In probabilistic terms, LDA looks for $P(z_i = j | z_{-i}, w)$. This is found by sampling z_i with a Gibbs sampling algorithm. During each iteration we calculate the Probability using Equation 2 (which is derived from Probability in Equation 1), where n is $_{-i}$ and n is $_{-}$. We then determine whether to keep or discard the sampled value. Equations 1 and 2 are derived and proven by Blei et al. [2]

$$P(z_i = j | z_{-i}, w) \propto P(w_i | z_i = j, z_{-i}, w_{-i}) P(z_i = j | z_{-i}) \quad (1)$$

$$P(z | \alpha, \beta, w) = \frac{n_{-i,j}^{w_i} + \beta}{n_{-i,j}^{w_i} + W\beta} \frac{n_{-,j}^{d_i} + \alpha}{n_{-,j}^{d_i} + W\alpha} \quad (2)$$

TLDA uses the LDA model in order to help score the sampled location of K . We again use another Gibbs Sampler to sample a value of K and score it based on

$P(k | z_i, w_i, \alpha, \beta)$. We can define this probability as equation 3.

$$P(k | z, w, \alpha, \beta) \propto P(z_1 | k, w_1, \alpha, \beta) P(z_1 | k, w_1, \alpha, \beta) \quad (3)$$

Each of the probabilities on the right side are similar and can therefore be rewritten in a simpler form as equation 4.

$$P(k | z, w, \alpha, \beta) \propto \prod_{z,w} P(z | k, w, \alpha, \beta) \quad (4)$$

In order to find the probabilities on the right side, we can think of the problem as a production of all of the topics and documents as a function of their dirichlet scores. In order to find the dirichlet scores, we can again return to Blei et al. Blei et al. define the dirichlet score to be equation 5, where we can assume that $P(\theta) = 1$.

$$Dir(\theta | \alpha) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} P(\theta) \quad (5)$$

With the Dirichlet function, we want to iterate over all of the Topics and all of the Document separately from one another and find the product of all of the dirichlet scores. We define $P(z | k, w, \alpha, \beta)$ as equation 6, where ω is the number of times a word has been assigned to topic t , and psi is the number of words in a document assigned to topic t .

$$P(k | z, w, \alpha, \beta) \propto \prod_{z,w} \left[\prod_T \frac{Dir(\langle \omega \rangle + \beta)}{Dir(\langle \beta \rangle)} \prod_D \frac{Dir(\langle \psi \rangle + \alpha)}{Dir(\langle \alpha \rangle)} \right] \quad (6)$$

Equation 6 allows us to sample the temporal split K in the documents - topics list. We continue this for n iterations and we should see the documents list begin to have a split somewhere along the documents which we can define as our temporal split. The next section discusses our

TOPIC_1	0.59219	TOPIC_2	0.40781
Dynamics	0.01739	Foreign	0.02328
Organization	0.01739	Policy	0.02303
Environmental	0.01688	Practice	0.02303
Risk	0.01671	Development	0.02278
Evolutionary	0.01671	Effects	0.02253
Crisis	0.01637	Adverse	0.02229
National	0.01620	Policy	0.02229
Exchange	0.01552	Global	0.02179
Bribery	0.01552	Human	0.02179
Social	0.01552	Capitalism	0.02154
Offices	0.01535	Growth	0.02154
Global	0.01535	Industrial	0.02154
Sustainability	0.01535	Inflation	0.02154
Competition	0.01518	Functions	0.02154
Investment	0.01501	Firm-Specific	0.02130

Figure 3. A typical run of the TLDA algorithm and the left portion of the Document-Topic matrix with the words that are related to the topic.

TOPIC_1	0.42119	TOPIC_2	0.57881
Corporate	0.01027	Natural	0.01101
Business	0.01013	Firm-Self	0.01010
Distribution	0.01013	Routines	0.01000
Security	0.01013	Activists	0.00980
Economic	0.01000	Knowledge	0.00970
Decision-Making	0.01000	Capital	0.00960
Environmental	0.00986	Foreignness	0.00960
Environmental	0.00986	Competences	0.00960
Regulation	0.00986	Management	0.00960
Location	0.00972	Dependence	0.00960
Strategy	0.00972	Intervention	0.00950
Geography	0.00958	Public	0.00950
Disclosure	0.00958	Practices	0.00950
Organization	0.00944	Relationships	0.00950
Risk	0.00930	Political	0.00939

Figure 4. A typical run of the TLDA algorithm and the right portion of the Document-Topic matrix with the words that are related to the topic.

experiment which attempts to locate a temporal split in a set of documents.

III. TLDA EXPERIMENTATION

Our TLDA model was implemented in MATLAB and was based on the Topic Modeling Toolbox supplied by Mark Steyvers, from Univ. of California, Irvine, and Tom Griffiths, from Univ. of California, Berkeley. [5] The model was implemented by creating a Gibbs Sampler for the Temporal Split. We split the original document list and words list up into two separate lists and calculate their z scores. We then propose a move in the temporal split, K , and calculate the likelihood of the proposed K value. If we find the proposed value to be better, we will move the split. This entire Gibbs Sampler is run for a large number of iterations.

A. The Experiment

For the purposes of experimentation, we created a small toy dataset with a known split. In order to ensure that the split was there, we defined a split location at document ten in the toy dataset and forced the split to have a significantly different set of words for the topics.

The dataset was comprised of a total of 30 documents containing words from a list of business and economic terms. Because the dataset was small, we limited the total number of topics to two. This was done so that we could easily distinguish between the topics and ensure that the topics were comprised of similar but different words. This would be equivalent to the topic of medicine where we see a shift between early medicine and modern medicine.

The Gibbs sampler was run for a total of 200 iterations. During each iteration, each of the two document lists would be separated and their topics re-scored in order to better maintain an accurate scoring of each proposed move. The initial split of the Gibbs Sampler was started randomly on a normal distribution with a mean at half of the total documents, in our instance the mean was 15, and a standard deviation of the total number of topics, 2 in our case. This was done in order to test whether the algorithm would move towards the correct location or not.

B. The Results

During most of the runs, we found that the K value that was proposed would often move beyond the K value that was defined, ten in our experiment. And in rare cases, it would move far beyond the defined split. However, in all cases, we found that the algorithm did eventually move back towards the correct answer. The split that was found was rarely the exact predefined split. Instead we found that the document had a mean error of about 1.74 documents on this toy dataset. A further discussion on the implications of the results that were received can be found in the next section, IV.

We are able to look at the various topics that are returned by the TLDA split. Figures 3 and 4 show a typical run of the TLDA algorithm and the split between the words that are returned for the topics before and after the split. Looking at the words that are related to the topics, we see that the topics that are discussed in TOPIC_1 on the left portion of the document-topic matrix (Figure 3) are similar to those in TOPIC_1 on the right portion of the document-topic matrix (Figure 4). In fact, we see some of the same words, e.g. "Environmental" is found on both sides of the matrix in TOPIC_1.

We see the same in TOPIC_2 on both sides of the document-topic matrix. In this example, we do not see any of the same words. We do however see that there are words that are very similar on both sides. For example, since Foreign and Policy are the top 2 words in TOPIC_2 on the left side of the document-topic matrix, we can assume that this topic has something to do with politics. If we look at the right side of the document topic matrix, we see words like "Political" nearer to the end of the list. This shows that TOPIC_2 on both sides of the matrix are similar but because the words are different nearer the top of the lists, we can see that there was some kind of change in the topic.

C. Issues in the Code

While running the code, we ran into significant errors that prevented many of the runs from completing properly.

We believe that these errors were caused by the way in which MATLAB handles the way that we are deleting and moving items in the document-topic matrix and the way in which MATLAB handles garbage collection. After updating, MATLAB managed to more reliably run the code. Despite this, after MATLAB runs the code several times in one session, MATLAB will likely crash.

In order to avoid this, X11 on Mac OS X has to be restarted to clear the memory that MATLAB needs in order to run the algorithm. It is hard to tell whether the bug lies in OS X's X11 implementation or MATLAB, but after restarting X11 and MATLAB, we were able to run the algorithm several times before another crash. We did not have time to run the algorithm on another operating system which runs MATLAB natively, such as Ubuntu or Windows.

It would be interesting to implement this in another language where garbage collection could be better handled by the programmer. MATLAB does not allow the programmer much access to free the memory that has been used. This seems to have been the cause of much of the frustration that we suffered in running this algorithm.

IV. CONCLUSION & FUTURE WORK

The TLDA model that we've discussed in this paper has shown promising results on the experiment that we ran the algorithm with. The results that were gained show that the algorithm is able to locate and generate separate lists of topics where we defined a temporal split to be in the toy data. Since we split the documents using a massive change in the words that were used, we can assure that given a topic that has undergone massive changes, we can find the a temporal split in the document.

There is still however a large amount of work that is left in this algorithm, and as the title of the paper suggests, this is only a step in the direction towards a Temporal Latent Dirichlet Allocation Model. Future Directions that we would like to take on this algorithm are scalability. Currently the algorithm is not fast enough to be implemented in something such as a web based information retrieval system.

There is also an issue of the current algorithm only being capable of finding a single temporal split across all topics in the list. Another future direction of this algorithm would be to include multiple temporal splits in the matrix. We might also consider a temporal split within a single topic itself at some point. Yet, despite the significant amount of work that is left in solving topic modeling with a temporal layer, this is a step towards a temporal topic modeling algorithm as the results of the experiment showed.

REFERENCES

- [1] T. Griffiths and M. Steyvers, "Finding scientific topics," *PNAS*, vol. 101, no. 1, pp. 5228–5235, Apr 2004.
- [2] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Jan 2003.
- [3] M. Steyvers and T. Griffiths, "Probabilistic topic models," In *T. Landauer, D McNamara, S. Dennis, and W. Kintsch (eds), Latent Semantic Analysis: A Road to Meaning.*, 2007.
- [4] T. Griffiths, M. Steyvers, and J. Tenenbaum, "Topics in semantic representation," *Psychological Review*, vol. 114, no. 2, pp. 211–244, Mar 2007.
- [5] (2007, December). [Online]. Available: http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm