Topic Modeling using LDA with Feedback Mechanisms

Alireza Bagheri Garakani
Department of Computer Science
University of Washington
Seattle, WA, USA
abagheri@cs.washington.edu

***Abstract*—Topic models provide a way to identify the latent topics from a collection of documents. Although the identified topics often appear quite representative of the data; just as often, there are parts of the output that appear erroneous or otherwise difficult to interpret by humans. This is a limitation of topic models that can be remedied by user feedback mechanisms. In this paper, I discuss two feedback actions: removing a word from a topic and removing a topic from a document. I apply these two functionalities in the framework of Collapsed Gibbs Sampling of an LDA model using a subset of the Wikipedia dataset. A preliminary evaluation of this method shows that such feedback mechanisms can be useful and efficiently implemented.**

I. INTRODUCTION

Finding documents among a large collection – whether that collection is some digital repository of documents or the internet – is an increasingly difficult task to carry out in a fast and effective manner. Search engines continue to better address this problem using sophisticated ranking algorithms, performance metrics for feedback, and even personalized results using a user's social data. However, perhaps there is better way than repeatedly submitting queries and choosing among ranked results to discover content (i.e., a better way to refine, broaden, or redirect your search domain).

With the ability to identify the underlying themes of documents, topic models may provide a better way to represent and navigate among documents. One popular method for topic modeling is Latent Dirichlet Analysis (LDA) [1], which defines a generative model for documents where each word is determined by a latent underlying topic structure. The implications of such a model allow for a greater interaction with documents (and similarly words and topics). For example, imagine a document based on a distribution of topics and effectively narrowing down or around a topic of interest and discovering more relevant documents [2]. Yet, LDA have far-reaching applications beyond text and topic-modeling; viewed as a mixed-membership model of grouped data, it has been extended to find patterns in genetic data, images, and many other areas [2].

One less explored area of research is how to incorporate a user-triggered feedback mechanism to help the algorithm converge to a more desirable set of topic labels. This is essential because, based on a variety of factors, the unsupervised learning of topics may be poor (i.e., poor text normalization, improper choice of topics in corpus, and others). As such, it is often desirable from a user's perspective to tweak the results and have the algorithm dynamically adapt. Note that when viewing this feedback as coming from multiple users we have effectively provided a way for LDA to benefit from crowdsourcing, as many other applications have shown to be useful.

Since we are discussing a mechanism that involves user interaction, visualization is clearly a very important element. The proper user interface should not only illustrate the output of the topic model, but should also incorporate an intuitive design for performing the feedback actions.

In this paper, I explore two feedback functionalities – namely, removing word from a topic and removing a topic from a document. I will show that these functionalities are intuitive and efficient within the framework of LDA inference via the Collapsed Gibbs Sampling. A preliminary evaluation of this method using a subset of the Wikipedia dataset appears to show promising results.

II. RELATED WORK

A. Topic Modeling

Among choices for topic modeling techniques, LDA is one of many options (Buntine and Jakulin explore several alternatives and how they are related [3]). However, even within LDA model, there are a vast number of variations – both in terms of model formulation and inference techniques.

Among many others, variations of LDA include modeling the change in topics over time [4], removing the 'bag-of-words' assumption [5, 6], and even using non-textual input (like learning natural scene categories in a collection of images [7]). Blei provides a more thorough overview of these and other variations [2]. In this paper, I use the standard LDA model as described in [1].

Moreover, since the task of exact inference is not tractable, there are also many variations to the method of inference; this is well summarized by Asuncion et al. [8]. In this paper, given its simple implementation and ability to easily integrate feedback, I use the Collapsed Gibbs Sampling (CGS) method, which is briefly discussed in the next section [9, 10].

B. Visualization

There has been a large and diverse amount of visualization techniques for topic model output, especially with extensions to the basic LDA model [4, 13].

With the standard LDA model, it is relatively simple to display many different types of information beyond document topic labels: similar documents (or topics or words), most relevant documents based on a particular topic, most relevant words from a topic, among many other things [9, 11, 14].

Chaney et al. implements a topic model browser that displays this information and others [11]. However, despite the various types of information and clean interface, I believe that quantitative information on the UI is important for understanding the data. Further, as it relates to providing feedback, it will be important to see how distributions change as a result of different actions.

There has also been a lot of work on different ways to navigate from one document to the next. Gerrish's "disciple browser" allows users to refine a topic distribution that is used to match against other documents [12]. For example, this allows for refining a query on 'Iran' more towards the topic of 'archaeology' and away from the topic of 'political science'.

C. Feedback

There is less work on methods to provide feedback to the algorithm. Some very recent work by Hu et al. explores this idea using Gibbs Sampling and a tree-based topic model that incorporates a metric of correlation between words [15]. In other words, adding a positive correlation between a pair of words would encourage this pair to form a topic (i.e. merge); while adding negative correlation between this pair would encourage the pair to find different topics (i.e. split) [15]. The motivation for using correlation appears valid as it enforces a soft constraint, instead of explicitly fixating or banning a word from a topic; however, one downside is that this may require many more samples in order to again converge (at least 100 for techniques described in [15]). Using the standard LDA model, I explore a simpler solution by creating a hard constraint on whether a word is welcomed within a topic or not. I show that it meets expectations and briefly discuss its limitations.

One important result from Hu et al. is that each round of user-feedback via Mechanical Turk showed improvement in relative accuracy [15]. Here, accuracy was measured using document labels from their dataset. This shows promise for incorporating user-feedback and interacting with the algorithm.

III. Method

A. LDA

LDA is a generative model that is assumed for producing a collection of documents. Within this model, we can 'generate' a document by selecting a distribution over topics $\boldsymbol{\theta}^{(d)}$ and sampling a topic labeling $\mathbf{z}$ for each word. With each words' topic label $\mathbf{z}$ determined, we can proceed to selecting the word $\mathbf{w}$ itself; this is done by sampling the distribution over words for each respective topic $\boldsymbol{\phi}^{(z)}$. In LDA, we define $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ as symmetric Dirichlet distributions with hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively [9]. Furthermore, the sampled topic $\mathbf{z}$ and word $\mathbf{w}|\mathbf{z}$ (as described above) are represented as multinomial distributions. These choices for distributions are convenient as the Dirichlet distribution is a conjugate prior for the multinomial distribution [1, 9]. A summary of this generative process is illustrated using plate notation in Figure 1.

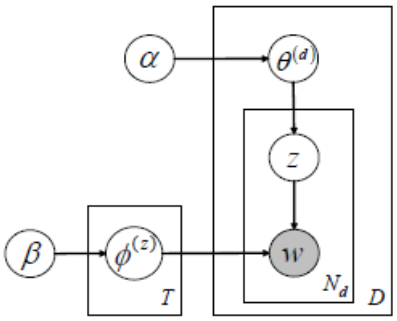

Figure 1. Plate Diagram for LDA showing how the latent variables (z, θ, ϕ) and constant hyperparameters (α, β) determine each word (w). Figure from [9]

The goal of LDA is not to generate words $\mathbf{w}$ based on known topic distributions; but rather, to work in the reserve direction – you are given a collection of documents and must find the latent topics assignments.

B. Posterior Approx via Gibbs Sampling

As discussed earlier, there are many different approaches to determining these latent variables. I use Collapsed Gibbs Sampling as described by Steyvers and Griffiths [9, 10]. This is an iterative sampling technique used to approximate $\mathbf{z}$, which can then be used to arrive at $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. This algorithm is summarized at a high-level in Algorithm 1.

1. Randomly assign topics to every word.
2. Repeat sample {
    a. For each token **t** in corpus {
        i. Remove topic assignment for **t**.
        ii. Approximate topic assignment for **t**, given all other topic assignments. (Equation 1)
        iii. If not initial burn-in period, accumulate sample **S**
    }
}
3. Using **S**, estimate $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. (Equation 2)

Algorithm 1.

Equations 1 and 2 are described in by Steyvers and Griffiths in [9] and are not replicated here.

C. Incorporating Feedback

I explore two different feedback operations; (1) removing a word from a topic, and (2) removing a topic from a document. I will first describe the former; however, I note in advance that the following explanation can be easily extended to the latter.

Intuitively, when a user marks a word as poorly representing a topic, this means that we should reassign our word to other topics. However, the question as to which new topic this assignment should be made needs to be addressed. Since a given word is represented by many topics, we will select an assignment based on these 'other' topics. Sampling a multinomial distribution based on the current weights of these

other topics has the effect of dispersing word assignment among other likely topic labels.

This can be viewed as running the sampling step defined in Algorithm 1 with two important modifications, as shown in Algorithm 2. Firstly, we only need to resample tokens that match the word and topic label of the feedback action. In other words, this can be seen as selective re-sampling of topic assignments. Further, there are substantial performance benefits here since we only perform on a small subset of a full sampling procedure.

For the second modification, we must remove topic assignment for all words in the first step before finding new assignments for any word; this will help ensure that the multinomial that we sample in subsequent steps appropriately represents alternative topics for assigning our word.

1. Let disperseSet be the set of all tokens that equal the chosen word and topic pair to be removed.
2. For each token **t** in disperseSet {
    a. Remove topic assignment for **t**

    }
3. For each token **t** in disperseSet {
    a. Approximate topic assignment for **t**, given all other topic assignments. (Equation 1)

    }

Algorithm 1.

This approach for removing a word from a topic can trivially be extended to removing a topic from a document. In this case, we select every token that matches the chosen topic and document pair to be part of our disperseSet. Having defined this set, we sample new topic labels from a multinomial distribution over other topics in the document.

IV. Dataset

As my corpus, I used a total of 4769 random articles from Wikipedia. The initial format of this data was in the form of an SQL dump with significant amount of unneeded metadata, like timestamp of article creation, username of last contributor, etc... Even further, the article text had to be reformatted from wiki markup language to plain text. I used an open-source tool, Wikipedia Extractor, to extract this plain text directly from the SQL dump [16].

The next step is text normalization. This step required some trial-and-error to work properly. In the end, my changes to the original text included removing stop words, non-alphanumeric tokens, no-alpha tokens, tokens with extremely high frequency (similar to stop words), tokens with frequency of only one, made tokens lowercase and a few other minor changes. These effects can be seen in **Figure 2**.

Anarchism is generally defined as a political philosophy which holds the state to be undesirable, unnecessary, or harmful, or, alternatively, as opposing authority or hierarchical organization in the conduct of human relations. Proponents of anarchism, known as "anarchists," advocate stateless societies based on non-hierarchical voluntary associations.	anarchism generally defined political philosophy holds state undesirable unnecessary harmful alternatively opposing authority

Figure 2. Running text normalization on the article text from the Wikipedia (left) prepared the input for LDA (right).

I. Evaluation and Discussion

As evaluation, I ran LDA with the Wikipedia normalized text as input and a small value for number of topics (T = 30); this small value is not representative of the vast number of topics in our corpus, however, it benefits from faster sampling and was preferable for evaluation on my desktop machine. Nonetheless, these values suffice for demonstrating the proposed feedback mechanisms. For choice of hyperparameters, I used values as suggested by [9], where $\alpha=50/T$ and $\beta=0.01$. I also chose the number of burn-in iterations to be 50, which appeared to be decent from prior experimentation.

We would like to observe the effect of removing a word from a topic. For this experiment, I used the same article as mentioned above in Figure 2 and I remove the word "man" from Topic 7 – at the time of action, this is the top topic for the document. Per our implementation, the topic distribution now missing that word should be updated accordingly (see Figure 3). Similarly, this word should be dispersed among remaining topics (see Figure 4). Both figures illustrate the effect of removing "man" from Topic 7 from the point of view of Topic 7 and Topic 19. There are no sampling iterations that occur within the snapshots shown below. Both figures are accompanied by percentages to illustrate the effect of this feedback action.

Top-10 words for Topic 7 (36.17% of document)	king	2.21%
	man	**1.31%**
	led	1.09%
	son	0.99%
	men	0.94%
	war	0.91%
	end	0.68%
	roman	0.67%
	empire	0.60%
	battle	0.55%
Action: Remove "man" from Topic 7		
Top-10 words for Topic 7 (39.34% of document)	king	2.16%
	men	1.08%
	led	1.06%
	war	0.94%

	son	0.90%
	end	0.64%
	roman	0.63%
	empire	0.60%
	battle	0.57%
	emperor	0.53%

Figure 4. Removal of word "man" causes it to be dispersed among other top words within topic. A new word, "emperor", emerges among the top-10 words for Topic 7.

Top-10 words for Topic 19 (9.73% of document)	men	1.74%
	man	**1.47%**
	red	0.96%
	led	0.80%
	india	0.67%
	china	0.64
	afghan	0.61%
	han	0.56%
	attack	0.46%
	chinese	0.45%
Action: Remove "man" from Topic 7		
Top-10 words for Topic 19 (10.77% of document)	men	1.79%
	man	**1.70%**
	red	1.16%
	led	0.82%
	china	0.78%
	soviet	0.64%
	india	0.62%
	afghan	0.62%
	han	0.61%
	chinese	0.58%

Figure 4. Removal of word "man" in Topic 7, encourages other topics (like Topic 19) to represent it more strongly.

From Figure 3 we observe the effect of removing "man" from Topic 7. As we would expect, it is quickly removed from the top-10 words defining Topic 7. Observing this effect on the second dominant topic of this document (Topic 19); we notice this topic tries to compensate for the occurrences of "man" within the document.

It should be restated that no sampling iterations took place after the action was taken; this is important because we would expect that with subsequent sampling these topics would converge to a better result. It should also be noted that, due to the insufficient number of topics defined and a lack of iterations prior to the action, topics do not appear to represent discrete ideas yet.

Furthermore, this evaluation is hardly sufficient for demonstrating the impact of the remove-word action on the entire set of documents, topics, and words. Taking a lesson from Hu et al. work, it would be interesting to use Mechanical Turk and use similar evaluation metrics on the resulting model to gauge its value [15].

music, album, song, band, record, play, bass, release, form, sing
computer, problem, work, science, research, machine, engineer, put, engineering, system, …
school, university, college, student, education, students, schools, art, year, program, …
character, game, series, story, novel, comic, book, fiction, stories, characters, …
car, engine, signal, design, cycle, cars, motor, engines, drive, amp, …
electron, energy, particle, atom, form, part, matter, field, mass, universe, …
cell, gene, protein, cells, dna, form, proteins, species, structure, organism, …
greek, son, god, name, myth, poet, poem, apollo, roman, athens, …

Figure 5. Subset of 50-topic LDA model on Wikipedia dataset. Each row represents a topic as represented by the 10 words listed (in order of representativeness within that topic).

II. Conclusion and Future work

Topic models provide an effective way to discover, navigate, and understand the content from a large collection of documents. With the proper choice of parameters and enough sampling, LDA with CGS inference performs surprising well on the Wikipedia dataset – several distinguishable topics can be identified based on article text (see Figure 5).

However, it is often desirable to tweak with the results of LDA and guide the system to reach better end results. I have demonstrated this interactivity by supporting two user feedback actions: removing a word from a topic and removing a topic from a document. These functions can be implemented as simple extensions to the CGS method. Moreover, since these actions do not require a full re-sampling within the algorithm to see its effects, operations can be implemented efficiently and are extendable feedback from multiple users.

Future work should address better evaluation of these mechanisms, explore other forms of providing feedback to the algorithm, and operate in an environment capable of running LDA with hundreds of topics for the entire Wikipedia dataset. I look forward to exploring these next steps as this work only scratches the surface of topic models with 'humans-in-the-loop'.

Acknowledgment

For guidance on this project, special thanks to Professor Carlos Guestrin (U. Wash), Yucheng Lo (U. Wash), and the entire teaching staff of CS229 led by Professor Andrew Ng (Stanford).

References

[1] Blei, D., Ng, A., Jordan, M. Latent Dirichlet allocation. J. Mach. Learn. Res. 3 (January 2003), 993–1022.

[2] D. Blei. Introduction to probabilistic topic models. Communications of the ACM, 2011.

[3] W. Buntine and A. Jakulin. Discrete component analysis. Lecture Notes in Computer Science, 3940:1, 2006.

[4] Blei, D., Lafferty, J. Dynamic topic models. In International Conference on Machine Learning (2006), ACM, New York, NY, USA, 113–120.

[5] Griffiths, T., Steyvers, M., Blei, D., Tenenbaum, J. Integrating topics and syntax. Advances in Neural Information Processing Systems 17. L. K. Saul, Y. Weiss, and L. Bottou, eds. MIT Press, Cambridge, MA, 2005, 537–544.

[6] Wang, C., Thiesson, B., Meek, C., Blei, D. Markov topic models. In Artificial Intelligence and Statistics (2009).

[7] Fei-Fei, L., Perona, P. A Bayesian hierarchical model for learning natural scene categories. In IEEE Computer Vision and Pattern Recognition (2005), 524–531.

[8] Asuncion, A., Welling, M., Smyth, P., Teh, Y. On smoothing and inference for topic models. In Uncertainty in Artificial Intelligence (2009).

[9] Steyvers, M., Griffiths, T. Probabilistic topic models. Latent Semantic Analysis: A Road to Meaning. T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, eds. Lawrence Erlbaum, 2006.

[10] Griffiths, T. and Steyvers, M. (2004). Finding scientific topics. Proceedings of the National Academy of Science, 101:5228--5235.

[11] Allison J.B. Chaney and David M. Blei. Visualizing topic models. In Intl. AAAI Conference on Social Media and Weblogs, Department of Computer Science, Princeton University, Princeton, NJ, USA, March 2012.

[12] http://dbrowser.jstor.org/browser.cgi

[13] http://topics.cs.princeton.edu/Science/

[14] http://vis.stanford.edu/papers/termite

[15] Hu, Y., Boyd-Graber, J., Satinoff, B. Interactive topic modeling, Interactive Topic Modeling. Machine Learning Journal, Under Review, 2013.

[16] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor