User-directed Non-Disruptive Topic Model Update for Effective Exploration of Dynamic Content

Yi Yang  
Northwestern University  
yiyang@u.northwestern.edu

Shimei Pan  
University of Maryland, Baltimore County  
shimei@umbc.edu

Yangqiu Song  
University of Illinois at Urbana-Champaign  
yqsong@illinois.edu

Jie Lu  
IBM T. J. Watson Research Center  
jielu@us.ibm.com

Mercan Topkara  
JW Player  
mercan@jwplayer.com

ABSTRACT

Statistical topic models have become a useful and ubiquitous text analysis tool for large corpora. One common application of statistical topic models is to support topic-centric navigation and exploration of document collections at the user interface by automatically grouping documents into coherent topics. For today’s constantly expanding document collections, topic models need to be updated when new documents become available. Existing work on topic model update focuses on how to best fit the model to the data, and ignores an important aspect that is closely related to the end user experience: topic model stability. When the model is updated with new documents, the topics previously assigned to old documents may change, which may result in a disruption of end users’ mental maps between documents and topics, thus undermining the usability of the applications. In this paper, we describe a user-directed non-disruptive topic model update system, nTMU, that balances the tradeoff between finding the model that fits the data and maintaining the stability of the model from end users’ perspective. It employs a novel constrained LDA algorithm (cLDA) to incorporate pair-wise document constraints, which are converted from user feedback about topics, to achieve topic model stability. Evaluation results demonstrate advantages of our approach over previous methods.

Author Keywords

Statistical Topic Model; Non-disruptive Topic Model Update; User Input; Dynamic Content

ACM Classification Keywords

H.5.m Information Interfaces and Presentation (e.g. HCI): Miscellaneous; I.2.7 Artificial Intelligence: Natural Language Processing-Text analysis

INTRODUCTION

In an age of information abundance/explosion, providing interactive interfaces to guide users through the navigation and exploration of the vast information space becomes a necessity for information applications [19, 12]. To achieve this goal, applications commonly employ statistical topic modeling techniques to structureize content collections by grouping documents into coherent topics, and utilize interactive visualizations at the interfaces to support topic-centric navigation and exploration.

Due to the highly dynamic nature of many content collections where new documents are frequently added, updates to existing topic models are necessary in order to capture the changes to topics (e.g. the emergence of new topics and the evolution of existing topics) over time. When topic model updates introduce substantial changes in the topic assignments of existing documents, the users’ mental maps between documents and topics may be disrupted, leading to increased cognitive load of the users and negatively impacting the usability of the applications. Therefore, how to maintain topic model stability during updates becomes a critical issue.

Latent Dirichlet Allocation (LDA) is one of the most commonly used approaches to topic modeling due to its capability to uncover hidden thematic patterns in text with little supervision. To handle update, standard (batch) LDA requires the topic model to be regenerated from scratch based on the updated dataset (including old and new content) [25], while online LDA incrementally updates the topic model when new content arrives without requiring a full scan of the entire dataset [9]. Because neither standard LDA nor online LDA explicitly constrains the topic model, the topic assignments of individual documents may vary significantly from one run to the next.

Let us use an example to illustrate why the lack of topic model stability using LDA may pose a serious problem. A user named Jane frequently needs to explore publications in certain fields for her work. She has been using a software tool that trains an LDA topic model on all the papers from a publication database to classify the collection of papers into three topics: Natural Language Processing (NLP), Speech Processing (SP), and Information Management (IM). Realizing that the topic model was created a year ago, Jane requested an update. The tool updated the topic model based on the current
In this paper, we describe a novel user-directed non-disruptive topic model update system (nTMU) to explicitly address topic model stability. It directly incorporates topic stability constraints to minimize the changes to the topic assignments of the same documents before and after model update. As a result, it can effectively maintain the stability of document-to-topic mappings and minimize the disruption to the mental maps of end users. The core module of nTMU runs a novel constrained topic modeling algorithm cLDA. It allows LDA to incorporate pair-wise document constraints such as document must-links and cannot-links. Evaluation on a benchmark dataset as well as a user study have demonstrated the effectiveness of our approach. We believe that both the nTMU system and the cLDA algorithm are significant contributions to the field since as far as we know, there isn’t any other system that addresses the topic model stability issue during topic model update, and no existing LDA algorithm can effectively handle document must-link and cannot-link constraints simultaneously.

The rest of the paper is organized as follows. Following a brief review of related work, we use two example scenarios to further illustrate the topic model stability issue and how our system addresses this issue. Then we describe notations used throughout the paper and provides a formal definition of topic model stability. We then present the nTMU system and introduce the cLDA algorithm, which incorporates user feedback into standard LDA as pair-wise document constraints. An algorithmic evaluation and a formal user study are described in the following sections. Lastly, we provide a discussion section and then conclude the paper.

RELATED WORK

Our work is closely related to three areas of research: user interface design, topic model update and topic model with prior knowledge.

User Interface (UI) Design

Usability is critical to the development of a user-friendly interface because it helps users to work in an effective, efficient, and manageable way [6]. Learnability, which is the ease with which a software application or product can be picked up and understood by users, is a key attribute of usability [17]. Because consistency and predictability are considered design factors that contributed to increased learnability [7], they have become two of the most important principles in UI design [18].

Despite being well-established principles in UI design, consistency and predictability have received little attention in the machine learning and data mining communities. New machine learning/data mining algorithms are rarely invented to specifically address these factors. However, for any interactive machine learning/data mining system, without proper backend support, it is impossible to achieve consistency and predictability at the UI level. For example, if the backend algorithm produces inconsistent results at different time intervals due to updates, the UI designed to surface these results will inevitably suffer from low consistency and predictability. More attention is needed during the development of backend algorithms to address these basic usability issues. Our work is the first in this direction.

Topic Model Update

Online LDA has been the main solution to topic model update. It incrementally updates the topic model when new documents arrive. Among existing online LDA algorithms, some use Variational Bayes (VB) [9, 26] to infer latent variables, others employ sampling methods [21, 3, 5]. Although online LDA sometimes can perform as well as batch LDA, it suffers from the same topic model stability problem as batch LDA since it mainly focuses on trying to approximate batch LDA’s performance without requiring a full scan of the entire dataset. In this paper, we propose a new LDA-based algorithm that employs a novel constrained topic modeling approach to topic model update. With this approach, the algorithm can maintain topic model stability and minimize user disruption by directly incorporating stability constraints in topic modeling.

Topic Model with Prior Knowledge

Recently, several algorithms have been developed to extend the standard LDA algorithm to incorporate user feedback or domain knowledge. Among them, some support seed constraints (e.g., document topic labels) [4, 20] while others support pair-wise word constraints (e.g., word must-links and cannot-links) [1, 2, 10]. [14] presents a topic modeling framework with network regularization, but it cannot easily handle cannot-links. To ensure topic model stability and at the same time to avoid over-constraining the new topic model, it is desirable to encode the stability constraints as pair-wise document constraints (e.g., document must-link and cannot-link constraints). However, none of these algorithms support general pair-wise document constraints without further extension.

There have also been works focusing on incorporating specific link relations (e.g., citations between documents) to facilitate social network-based topic analysis [23, 16, 13]. Since all of them employ generative models, they cannot directly formulate cannot-link constraints between documents. Our constrained LDA algorithm is more general and it can handle both types of document constraints (must-links and cannot-links).

EXAMPLE SCENARIOS

In this section, we provide two example scenarios to illustrate the topic model stability problem during dynamic topic-centric exploration of online news articles. For both scenarios, we also show how the nTMU system we have developed
aids the user, named Alice, by interactively incorporating her feedback during topic model update to keep the topic model stable.

In the first scenario, news articles related to Fiscal Cliff and Obamacare are put under two different topics based on the initial topic model created. Alice explores the articles under both topics but focuses more on Obamacare which is more closely related to her personal interests. A few weeks later, as more news articles become available, the topic model needs to be updated. When standard LDA is used to update the model, the Fiscal Cliff and Obamacare articles are put under a single topic as they are both related to government issues, which means that the articles about two previously separate topics are now mixed together. After this update, Alice is unable to easily relocate the old Obamacare articles and has difficulty identifying the changes happening to Obamacare between several weeks ago and now. In contrast, using the nTMU system which conducts constrained topic model update, Alice can indicate that she wants to keep the fiscal cliff and Obamacare articles separate since she wants to follow the news on these two events separately. nTMU incorporates this feedback during topic model update, so the articles about fiscal cliff and Obamacare remain under separate topics after the update to satisfy Alice’s preference.

In the second scenario, the initial model includes (among other topics) two topics, one about Hurricane Sandy and the other about Fiscal Cliff. Alice reads several articles related to Hurricane Sandy. A few weeks later, the topic model is updated to include a new set of news articles. In this new article set, some Hurricane Sandy articles discuss government funding situations to aid recovery after the hurricane, which happen to share many common keywords as the articles about Fiscal Cliff. As a result, some previous Hurricane Sandy articles are put under the same topic as the articles related to Fiscal Cliff when standard LDA is used to create the updated model. This totally confuses Alice as she does not understand why those articles about Sandy are not with other Sandy articles but mixed with those about Fiscal Cliff. As a result, she questions the accuracy of the system and deems the system not useful. In comparison, the nTMU system will ensure that all Hurricane Sandy articles remain together and are not mixed with Fiscal Cliff articles after the update.

NOTATIONS AND DEFINITIONS
In this section, we first briefly review topic model representation and topic label assignment based on the standard LDA algorithm (the same notations are also used in later sections). Then we formally define topic model stability.

**Topic Model Representation**
Statistical topic models derive latent topics from a text collection consisting of a set of documents \(d \in D\). Each document is a sequence of words \(w_d\) drawn from a vocabulary \(V\) of size \(W\). LDA provides a generative topic model where topics are modeled as latent variables \(z_n \in K\) for each word \(w_n\), given a set of topics \(K\). The number of hidden topics in LDA is \(K\).

In LDA, \(\theta_d\) is the topic distribution for document \(d\) which is drawn from a Dirichlet distribution, \(\theta_d \sim \text{Dir} (\alpha)\); \(\phi_k\) is the topic-specific word distribution which is also drawn from a Dirichlet distribution, \(\phi_k \sim \text{Dir} (\beta)\). \(\alpha\) and \(\beta\) are the hyper-parameters for the two Dirichlet distributions.

We focus on LDA inference using Gibbs sampling. The collapsed Gibbs Sampling algorithm iterates through and updates all the latent variables based on the following formulas:

\[
P(z_i = j | z_{-i}, w) = \frac{(n_{d,i} + \alpha) n_{w,j}^{-1} + \beta}{n_d^{-1} + K\alpha n_j^{-1} + W\beta} \tag{1}
\]

where \(n_{d,i}^{-1}\) is a count that does not include the current assignment of \(z_i\). The variable \(n_{d,j}\) is the count of topic \(j\) in document \(d\), and \(n_{w,j}\) is the count of word \(w\) being assigned to topic \(j\), where \(n_d = \sum_j n_{d,j}\), and \(n_j = \sum_w n_{w,j}\).

Finally, we can estimate document \(d\)'s \(j\)th topic probability \(\hat{\theta}_{j|d}\) and topic \(j\)'s word-specific probability of \(w\) \(\hat{\phi}_{w|j}\) from the topic samples \(z\) by

\[
\hat{\theta}_{j|d} = \frac{n_{d,j} + \alpha}{n_d + K\alpha} \quad \hat{\phi}_{w|j} = \frac{n_{w,j} + \beta}{n_j + W\beta} \tag{2}
\]

**Topic Label Assignment**
Given document \(d\)'s topic distribution \(\hat{\theta}_d\), we can assign \(d\)'s topic label \(l_d\) to be the topic that has the maximum value in \(\hat{\theta}_d\). However, it is possible that some topics learned by LDA are only “background” topics which have significant non-trivial probabilities over many documents [22]. Since background topics are often uninteresting ones, a weighted topic distribution could be used to filter them. Therefore, we normalize \(\hat{\theta}_{dk}\) by the sum of the \(k\)th topic value over all documents. This idea is similar to the inverse document frequency used in tf-idf, which normalizes the weight of a word by its overall occurrences in all documents. A weighted \(k\)th component \(\hat{\theta}_{dk}'\) for document \(d\) is computed as follows:

\[
\hat{\theta}_{dk}' = \frac{\hat{\theta}_{dk}}{\sum_{i=0}^{D} \hat{\theta}_{ik}}. \tag{3}
\]

Then, instead of assigning the document label to the topic that has the maximum value, we choose the topic that has the maximum normalized value. Therefore, the topic label of document \(d\) is \(l_d = \arg \max_{k \in K} \hat{\theta}_{dk}'\).

Note that there are other methods of assigning a topic label to a document. For example, we can use the per-document topic distribution as a feature vector in an unsupervised clustering algorithm such as k-means clustering, where the number of clusters equals to the number of topics in LDA. In our work, for simplicity, we utilize the straightforward method described above for topic label assignment, which is commonly used in real-world topic modeling systems.

**Topic Model Stability**
The topic model stability problem happens when there is new data coming in and the existing topic model needs to be updated. With standard LDA, if an existing model is refreshed with new documents, it is unavoidable that the model’s parameters will change. As a result, the top keywords of the
same topic and the topic label assignments of old documents may change. Given this, there are two different ways to define topic model stability: (1) keep the top keywords associated with each topic unchanged; (2) keep the topic label assigned to each old document unchanged.

We believe the first definition is too rigid since it does not allow a topic to evolve. For example, early papers on Natural Language Processing (NLP) focused on linguistic methods. As a result, the top keywords of the old NLP topic may include “grammar, syntax, semantics, subject, verb.” Nowadays, more and more NLP systems employ data-driven approaches. Statistical and machine learning based NLP methods have become mainstream. Thus, after updating the topic model with new NLP publications, we expect that the top NLP keywords may include “statistics, learning, unigram, frequency, corpus.” Since the change of top topic keywords is the result of topic evolution, minimizing the change of top topic keywords does not seem to be a good solution. In this work, we adopt the second definition which allows topics to evolve with new content. With this definition, maintaining topic model stability means minimizing the change of topic labels assigned to the same documents before and after an update. Specifically, we use the following formula to define topic model stability. Here, we denote $D_1$ as the dataset used to train the old topic model, $D_2$ as the new dataset and $D_3 = D_1 \cup D_2$ as the total dataset used to train the new topic model. $l_{1i}$ is the topic label assigned to the $i$th document based on the old topic model, and $l_{2i}$ is the topic label assigned to the $i$th document after the model is updated to the new topic model. Figure 1 illustrates the flow of new data.

$$S = (1 - \frac{\sum_{d_i \in D_3} I(l_{1i} \neq l_{2i})}{|D_1|}) * 100\%,$$  

where $I(\cdot)$ is the indication function. $S$ equals to 1 when the topic labels for all the documents in $D_1$ remain the same after the update.

**NON-DISRUPTIVE TOPIC MODEL UPDATE (NTMU)**

The standard topic model update systems do not have users in the loop. When new data are coming in, based on different topic model update strategies [25], topic models are retrained on either both old data and new data, or only the new data. As we will show in the System Evaluation section, such topic model update systems have poor topic model stability since their main objective is only to fit the data into the model. Figure 2 illustrates our system nTMU (user-directed non-disruptive Topic Model Update). It consists of four main components and includes users as a critical part. Below we describe these components in counter-clockwise order as shown in the figure.

Data, such as text collections, arrive at the nTMU system in an online fashion. The topic model is created and updated by the Constrained LDA component, which employs the cLDA algorithm (described in detail in the next section). Unlike unsupervised standard LDA, cLDA is capable of taking document-level user feedback encoded as document must-links and cannot-links. In the first round when the nTMU system is newly started with an initial set of data and no user feedback, cLDA behaves the same as standard LDA without any constraints. The system does allow users to provide prior domain knowledge/labels, but we assume that there is no prior information available in the current setting.

Given the topic modeling results from the Constrained LDA component, the Topic Assignor component assigns a topic label to each document using the method discussed in previous section. Since Gibbs sampler randomly initialize topic samples for each document, the topic label of the same topic before and after update can be different, and we need to align the topic label to make the labels consistent. Therefore, Hungarian algorithm is applied to achieve this task by maximizing the label matching. Basically, it will find an optimal matching that maximizes the number of the same document label before and after the update.

Next, the User Interface component presents the labeled results to the end users. It displays for each topic the most probable words based on topic-word distributions (Equation 2), along with a treemap-like visualization showing the twenty most representative documents that are assigned to this topic. Users can also click on the “Show More button” to see more representative documents. The title of each document is displayed, and if a user clicks on a document, a summary window will pop out to show the detailed information of this document, including time, author and highlights from the content. Figure 3 provides a screen shot of the user interface.

![Diagram](http://en.wikipedia.org/wiki/Hungarian_algorithm)
The treemap-like visualization has been demonstrated to be effective to support topic-centric navigation and exploration of document collections [11].

The system allows the user to provide feedback about the topics through the interface. In particular, for each topic, the system asks the user whether the topic is coherent and whether s/he wants to keep it. For example, if the content of the representative documents of the topic is consistent with this topic, the user would respond with Yes to the system. However, if the user decides that s/he does not care about the topic or the topic has inconsistent keywords and representative documents, he/she would respond No to the system.

After the feedback is collected, the Feedback Processor component converts the user’s Yes or No feedback to document must-link and cannot-link constraints. A must-link constraint between two documents indicates that they should belong to the same topic, and a cannot-link constraint indicates that the documents should belong to different topics. For a topic that the user responses with Yes, must-link constraints are added for each pair of the documents that have this topic label, and cannot-link constraints are added between documents with this topic label and documents with a different topic label. In contrast, for a topic that the user responses with No, neither must-links nor cannot-links are added. All the must-links and cannot-links are fed to the Constrained LDA component for constrained topic model update, which completes the loop.

**CONSTRANDED LDA (CLDA)**

In previous section, we introduce nTMU, the Non-Disruptive Topic Model Update system, that is capable of incorporating user feedback into standard LDA, but we treat the Constrained LDA module as a black box. In this section, we will present Constrained LDA algorithm in details.

We focus on two types of document level constraints: must-links and cannot-links. Specifically, we denote $\mathcal{M}_i \in \mathcal{D}$ as the set of documents sharing must-links with document $d_i$, and $\mathcal{C}_i \in \mathcal{D}$ as the set of documents sharing cannot-links with document $d_i$. The idea behind cLDA is that, we want to make documents share must-links close to each other, while documents share cannot-links far away from each other, in the $K$-dimensional space, by controlling documents’ topic distribution prior.

**The Role of Concentration Parameters**

In LDA, the prior of $\theta$ is a Dirichlet distribution, which is denoted by $\text{Dir}(\tilde{\alpha}_g) = \text{Dir}(\alpha_{g,1}, ..., \alpha_{g,K})$. $\tilde{\alpha}_g$ is the global hyperparameter, and $\alpha_{g,i}$ determines how “concentrated” the probability mass of a sampled $\theta$ is likely to be on topic $i$. If all the $\alpha_{g,i}$ are less than one, the probability mass tends to be concentrated on a few topics. If all the $\alpha_{g,i}$ are greater than one, the probability mass tends to be more uniformly distributed. Meanwhile, smaller $\alpha_{g,i}$ attracts more concentration on topic $i$. A simple and commonly used Dirichlet distribution is the symmetric Dirichlet distribution, where $\alpha_{g,1} = \alpha_{g,2} = ... = \alpha_{g,K} = \alpha$. If there is no information about relationships of the documents, the concentration parameters are normally set to be equal. Then the model can update the likelihood by learning the posterior of $\theta_i$. When we have knowledge about a document’s topic distribution, e.g., two documents have similar topic distribution, the topic distributions of documents cannot be assumed to be independently sampled. To achieve this, we manipulate the Dirichlet prior over document-topic distribution such that document must-link and cannot-link constraints can be incorporated into the topic model. In such a way, we use an asymmetric Dirichlet prior over document-topic while keep the Dirichlet prior over topic-word symmetric [24]. In the following, we explain how to incorporate the prior knowledge encoded in document must-links and cannot-links into an LDA topic model.

**Must-link Constraint**

A must-link between two documents $d_1$ and $d_2$ suggests that $d_1$ and $d_2$ should share the same topics, e.g., both are Sports news. Thus $\hat{\theta}_1$ should be similar to $\hat{\theta}_2$, and both distributions on the $K$-dimensional space should be close to each other.

Given the documents in $\mathcal{M}_i$, we introduce an auxiliary variable $\hat{\alpha}_i^\mathcal{M}$:

$$\hat{\alpha}_i^\mathcal{M} = T \star \frac{1}{|\mathcal{M}_i|} \sum_{j \in \mathcal{M}_i} \hat{\theta}_j,$$  

where $T$ controls the concentration parameters. The larger the value of $T$ is, the closer $\hat{\theta}_i$ is to the average of $\hat{\theta}_j$’s. $\mathcal{M}_i \in \mathcal{D}$ is the set of documents sharing must-links with document $d_i$.

**Cannot-link Constraint**

A cannot-link between documents $d_1$ and $d_2$ suggests that $d_1$ and $d_2$ should not have the same topics, for example, one is about Sports and the other is about Politics. Thus, $\hat{\theta}_1$ should not be similar to $\hat{\theta}_2$, and both distributions on the $K$-dimensional space should be really far away from each other.
Given the documents in $C_i$, we introduce the following auxiliary variable:

$$\tilde{\alpha}_i^C = T \arg \max_{j \in C_i} \min_{j \in C_i} KL(\tilde{\theta}_i, \tilde{\theta}_j),$$  

where $KL(\tilde{\theta}_i, \tilde{\theta}_j)$ is the KL-divergence between two distributions $\tilde{\theta}_i$ and $\tilde{\theta}_j$. This means we choose a vector that is maximally far away from $C_i$, in terms of KL divergence to its nearest neighbor in $C_i$. $C_i \in D$ is the set of documents sharing cannot-links with document $d_i$.

Then in each iteration, we draw a $\tilde{\theta}_i$ from the following distribution:

$$\tilde{\theta}_i \sim Dir(\eta_g \tilde{\alpha}_g + \eta_M \tilde{\alpha}_M^M + \eta_C \tilde{\alpha}_C^C) = Dir(\tilde{\alpha}_i).$$

Here, $\eta_g$, $\eta_M$ and $\eta_C$ are the weights to control the trade-off among the three terms. Note that in the first iteration of learning, it is possible that all the $\tilde{\theta}_i$'s are initialized by drawing from the global $\alpha_g$ solely. In our experiment, we choose $T = 100, \eta_g = \eta_M = \eta_C = 1$.

### Inference with Gibbs Sampling

Given a set of document must-links $M$ and cannot-links $C$, we infer the values of the hidden variables $z$ using collapsed Gibbs sampling as in LDA [8]. In each iteration, a topic assignment $z_i$ of word $w_i$ in document $d$ is sampled based on all the other variables. We directly present the conditional probability as following distribution:

$$P(z_i = j | z_{-i}, w, M_d, C_d) = \frac{(n_{-d,j} + \alpha_{d,j}) n_{w,j} + \beta}{n_{d,j}^{T} + \sum_{k=max}^{K} \alpha_{d,k} n_{j}^{T} + W\beta},$$

$$\propto (n_{-d,j} + \alpha_{d,j}) n_{w,j}^{T} + \beta$$

It is worth comparing Equation 8 with Equation 1. In LDA, the hyperparameter $\tilde{\alpha}$ is not updated during the training. Moreover, in most of the real world LDA system, for simplicity reason, $\tilde{\alpha}$ is a uniformed vector with the same value for each vector component over all documents. However, in cLDA, $\tilde{\alpha}$ is not a fixed value but is updated in every iteration based on the constraint sets, and different documents will have different hyperparameter which also reflects the relationship among documents.

The new iterative Gibbs sampling process is shown in Algorithm 2.

**Algorithm 1** cLDA Gibbs Sampling for document $d$ with constraint set $M_d, C_d$

1. $\tilde{\alpha}_d = \eta_g \tilde{\alpha}_g + \eta_M \tilde{\alpha}_d^M + \eta_C \tilde{\alpha}_d^C$
2. for $w_i$ in $d$ do
   3. for $j \leftarrow 0$ to $K$ do
      4. $f(z_i = j | z_{-i}, w) = (n_{-d,j} + \alpha_{d,j}) \frac{n_{w,j} + \beta}{n_{w,j} + \beta}$
   5. end for
   6. sample new topic assignment for $w_i$ from $f(z_i)$
3. end for

The difference between the cLDA's Gibbs sampling and the standard LDA's Gibbs sampling lies in line 1. In standard LDA, document $d$'s topic distribution prior $\tilde{\alpha}_d$ is a vector with fixed values. However, in cLDA, $\tilde{\alpha}_d$ is updated in every iteration based on $d$'s constraints set $M_d, C_d$. If $d$'s constraints set $M_d, C_d$ are empty sets, then the cLDA Gibbs sampling for $d$ is reduced to LDA Gibbs sampling.

### SYSTEM EVALUATION

In this section, we evaluate our nTMU system’s performance from two aspects, topic model quality and topic model stability. From a user’s standpoint of view, a good topic model update system should not only be able to fit data into model well but can also maintain a stable topic model transition.

For evaluation, we use NIPS dataset, a benchmark dataset commonly use in text analysis. It contains 1,740 papers from the NIPS conferences between 1987 and 1999. Documents in the dataset are sorted based on their timestamps. We use the first 844 documents (from 1987 to 1993) as $D_1$, next 896 documents (from 1994 to 1999) as $D_2$. Note that the documents in this dataset do not have ground-truth labels. After removing stopwords and stemming, the dataset has 6,448 unique words. For topic modeling parameters, we choose 20 as the number of topics since it is commonly reported in literatures that use NIPS dataset. We also set the global hyperparameter $\alpha_g = 1$, and $\beta = 0.01$. To make sure the Gibbs sampling chain converges, we average over 10 runs for each experiment, and for each run, 500 iterations. The first two experiments are conducted without having user in the loop. Thus, the feedback processor component automatically generates document constraints. Must-links are added for documents with the same topic label, while cannot-links are added between documents with different topic labels. In the experiments, for each document, 10 must-links and 10 cannot-links are randomly sampled.

#### Experiment 1: Topic Stability

In the first experiment, we test nTMU’s capability in maintaining topic stability. We compared nTMU against three baseline systems:  

1. Standard LDA: it jointly resamples all topics for the entire collection, including both the old and the new documents, and old documents’ topic samples are not reused. Therefore, it retracts the whole topic model on the entire collections from scratch.

2. Fixed Fold-in: it jointly resamples topics for all the new documents, but it keeps old documents’ topic samples fixed. Therefore, old documents’ topic samples are used to initialize the topic model, but they are not updated. Note that although this method keeps all old documents’ topic sample fixed, it does not necessarily mean that the topic labels for the documents will be the same before and after the update. There are two reasons that the topic label would be changed. First, document’s topic distribution $\tilde{\theta}$ is

2http://psiexp.ss.uci.edu/research/programsdata/toolbox.htm
a weighted distribution (Equation 3), and its value is normalized over all the documents, including the new documents. Second, since topic labels are aligned by Hungarian algorithm which looks for a “maximum matching” on all documents, it is likely that the old documents’ label indexes are swapped, which changes the topic labels for old documents.

3. Rejuvenated Fold-in: in addition to jointly resampling topics for all new documents, it also updates old documents’ topic samples. Therefore, after being used to initialize topic model, old documents’ topic samples will be updated.

Both (2) and (3) were inspired by the ideas behind sampling-based inference strategies for new documents discussed in [25]. In fact, the difference among three baseline topic model update methods is how to initialize new topic model given old topic model results. In standard LDA, old documents’ topic samples have no impact on the new topic model, while in Fixed Fold-in and Rejuvenated Fold-in, old documents’ topic samples will have different impacts.

Table 1 shows the topic model stability results on three baseline systems and our nTMU. Among all the topic update methods, nTMU significantly outperforms the other methods in maintaining high topic stability. In particular, it improves the standard LDA’s topic model stability by 103%, and it also improves the second best Fixed fold-in method by 46.3%. Note that the results are averaged over 10 runs for each topic model update method.

<table>
<thead>
<tr>
<th>Model update method</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard LDA</td>
<td>43.3%</td>
</tr>
<tr>
<td>Fixed fold-in</td>
<td>60.2%</td>
</tr>
<tr>
<td>Rejuvenated fold-in</td>
<td>57.8%</td>
</tr>
<tr>
<td>nTMU</td>
<td>88.1%</td>
</tr>
</tbody>
</table>

Table 1. Topic model stability performance of different model update methods.

**Experiment 2: Topic Coherence**

In addition to topic stability metric, we also evaluate our nTMU using topic coherence metric. Recent research [15] has shown that topic coherence is highly consistent with human judgment than held-out dataset likelihood. Thus, here we use topic coherence to assess a topic model’s quality. Following [15], topic $t$’s coherence is defined as $C(t : V^{(t)}) = \sum_{m=2}^{M} \sum_{l=1}^{n-1} \log \frac{F(v_l^{(t)}, v_{l+1}^{(t)\prime}) + 1}{F(v_l^{(t)})}$, where $F(v)$ is the document frequency of word type $v$, $F(v_l^{(t)}, v_{l+1}^{(t)\prime})$ is the co-document frequency of word type $v$ and $v'$, and $V^{(t)} = (v_1^{(t)}, ..., v_M^{(t)})$ is a list of the $M$ most probable words in topic $t$. In our experiments, we choose the 20 most probable words to compute topic coherence, i.e., $M = 20$. In addition, since LDA usually generates common background topics which appear in many documents and thus un-interesting, we also filtered those topics based on the method proposed in [22] before we compute the coherence scores for all the methods.

As shown in Figure 4, nTMU and Standard LDA achieve similar topic coherence score, and both methods perform better than Fixed Fold-in and Rejuvenated Fold-in methods with statistical significance at $p = 0.05$, using Chi-Square test. Although nTMU is slight worse than standard LDA, but there is no statistical significance. Since this is a simulation experiment, and the constraints are added automatically within each topic label, it is inevitable that 1). mistakes are introduced to the system for inconsistent or uncorrelated topics; 2). the model is over-constrained by the constraints. For example, Fixed Fold-in has the worst topic coherence performance because it does not allow topic samples in old documents to update. Therefore, this system is over-constrained and has a poor topic model quality. However, we can see from the figure that the nTMU system can still achieve topic coherence nearly as good as standard LDA update system, which does not have any constraints applied and have the most freedom to fit new data.

![Figure 4. Topic Coherence of different topic model update methods.](image)

To sum up, unlike Fixed Fold-in and Rejuvenated Fold-in who achieve higher topic stability than standard LDA at the cost of lower topic coherence, nTMU achieves the highest topic stability score among all the methods without sacrificing any topic coherence.

**Experiment 3: Incorporating User Guidance**

In the above two experiments, nTMU system simulates user interaction by adding constraints for each topic automatically. In this experiment, instead of the simulations, we demonstrate the benefit of keeping users in the loop and allowing them to guide topic model update.

Since nTMU employs a constrained topic modeling framework, it needs to satisfy two goals simultaneously: (1) to satisfy as many constraints as possible (2) to derive a topic model that fits the data. In general, if goal (1) matches goal (2), the constraints will help the system to meet goal (2) (e.g., the ground-truth labels). But if goal (1) is competing with goal (2) (e.g., minimizing end user disruption in nTMU), goal (1) may distract the system from meeting goal (2). Here is an example on how goal (1) in nTMU might distract the system from achieving goal (2): With an old topic model, a paper on language modeling was categorized as an SP paper since the method was first adopted in the speech recognition community. Over the years, language modeling became very popular in the NLP community for tasks such as machine translation, POS tagging and parsing. As a result, now, it is more appropriate to categorize this as an NLP paper. But due to the
incorporating user guidance in managing topic model update. This result has clearly demonstrated the benefit of stability constraints on all the topics had the worst statistical significance. Finally, the default model that enforces stability of incoherent topics deteriorated noticeably with stability, as the metric to assess the topic model’s quality. As shown in Figure 5, nTMU that uses coherent topic stability constraints performed very well. At the end of the 2000 iterations, there is no difference between the perplexity of this model and the perplexity of a model that has no constraint. In contrast, the performance of the model that enforces the stability of incoherent topics deteriorated noticeably with statistical significance. Finally, the default model that enforces the stability constraints on all the topics had the worst performance. This result has clearly demonstrated the benefit of incorporating user guidance in managing topic model update.

To validate the above concerns and to demonstrate the benefit of keeping users in the loop, we designed the following experiment. First, we allowed a user to choose a subset of the topics (e.g., 3) that are important to him. Frequently, user selected topics are coherent topics. In the experiment, this user is one of the authors who is familiar with the topics in this dataset. We also compared this model with two alternatives: (1) a model which kept three incoherent topics stable (b) a model which kept all the topics stable. With this setting, we wanted to verify that (1) keeping incoherent topic stable will hurt the system’s capability to adapt to the data (2) keeping all the topics stable will over-constrain the system and prevent it from fitting the data. Here, since we want to measure how well the learned model fits the test dataset, rather than the consistence with human judgement, we use perplexity, not topic coherence, as the metric to assess the topic model’s quality. As shown in Figure 5, nTMU that uses coherent topic stability constraints performed very well. At the end of the 2000 iterations, there is no difference between the perplexity of this model and the perplexity of a model that has no constraint. In contrast, the performance of the model that enforces the stability of incoherent topics deteriorated noticeably with statistical significance. Finally, the default model that enforces the stability constraints on all the topics had the worst performance. This result has clearly demonstrated the benefit of incorporating user guidance in managing topic model update.

Setting Model Parameters
In our current system, we need to set multiple model parameters, such as the number of topics, the value of concentration parameters, and the weights of must-link and cannot-link constraints. Although we can use heuristics to estimate the parameters for a dataset based on its characteristics (e.g., size of the data set), in reality, the parameters will still need to be tuned empirically for each new dataset. Optimizing such parameters for each dataset automatically is an active research topic in Machine Learning and beyond the scope of the paper.

USER STUDY

Experiment Design
We also conducted a user study to evaluate the impact of topic stability in an automated document exploration system. We used Amazon Mechanical Turk (MTurk) as our study platform. In order to design tasks that are not too difficult for the workers (called Turkers), we decided to use a general news dataset instead of a scientific publication dataset. Our news dataset includes CNN news articles from October, 2012 to November, 2013, covering five prominent topics at the time including “Fiscal Cliff”, “Hurricane Sandy”, “Violence in Iraq”, “Obamacare”, and “Syrian Civil War”. Since it is impossible for a Turker to explore a very large dataset during the time given at the study, we limited the dataset to include 320 articles.

We sorted the articles based on their timestamps and then split the dataset into two halves. The first half was used to train an initial topic model using LDA. Then we added the second half when updating the topic model. We employed a between-subject design, testing two different update algorithms, one used standard LDA, the other used nTMU. Due to the similarities of some of the topics in the dataset (e.g., both “Obamacare” and “Fiscal Cliff” are about US politics), the topic model was not very stable during update. For example, before update, “Fiscal Cliff” and “Obamacare” were two different topics. After the update using LDA, they merged into one general topic of “US Government Affairs”. With nTMU, the system still maintained two separate topics in the updated topic model.

We designed two Human Intelligence tasks (HITs), one for each test condition. Each HIT was divided into two parts: before-update tasks and after-update tasks. Before-update tasks were designed to help a Turker to build a mental model of the system. The before-update tasks were exactly the same as the after-update tasks. For example, before an update, a Turker was asked to choose the best label for a given topic by inspecting the top topic keywords and the top articles on that topic, both inferred by the system. S/he was also asked to select a few articles s/he would like to read and write down a few details from each article such as its title. The after-update tasks were designed to test how topic stability affects (a) a Turker’s comprehension of the topics in the updated model (2) a Turker’s ability to recall and relocate the articles s/he has chosen before the update.

Figure 5. The benefit of human guidance.
We used both subjective and objective metrics to evaluate the effectiveness of these methods. Our objective evaluation metrics included

1. **KTLA_BU** (Keyword-based topic label accuracy before update). This measures whether a Turker is able to pick a correct topic label based on the top topic keywords before model update.

2. **ATLA_BU** (Article-based topic label accuracy before update). This measures whether a Turker is able to pick a correct topic label based on the top articles in a topic before model update. The top articles were automatically inferred by the system.

3. **BTLA_BU** (Topic label accuracy based on both topic keywords and top articles before model update). This measures whether a Turker is able to pick a correct topic label based on both the topic keywords and the top articles in a topic before model update.

4. **KTLA_AU** (Keyword-based topic label accuracy after model update). This measures whether a Turker is able to pick a correct topic label based on the top topic keywords after model update.

5. **ATLA_AU** (Article-based topic label accuracy after model update). This measures whether a Turker is able to pick a correct topic label based on the top articles in a topic after model update.

6. **BTLA_AU** (Topic label accuracy based on both topic keywords and top articles after model update). This measures whether a Turker is able to pick a correct topic label based on both the topic keywords and the top articles in a topic after the model update.

7. **ARSR_AU** (Article relocating success rate after update). This measures whether a Turker is able to relocate the articles s/he found before an update.

In addition, we included four subjective evaluation metrics:

1. **TD** (Topic difficulty) This measures how difficult it is to understand the system-derived topics.

2. **ARD** (Article relocating difficulty) This measures how difficult it is to relocate the articles a user found before.

3. **UA** (Use Again) This measures the likelihood of a user to use the system if it is available.

4. **RF** (Recommend Friends) This measures the likelihood of a user to recommend the system to others.

All the survey questions are rated on a 5-point Likert scale with 1 being the least desirable and 5 the most desirable.

**Experiment Results**

Overall, we have collected data from 80 Turkers, 40 for each test condition. Figure 6 shows the before-update objective metrics. Since the before-update tasks are exactly the same in the two test conditions, we did not expect much differences. Moreover, any significant difference in these scores may indicate an irregularity of the test conditions and thus require further investigation.

As shown in Figure 6, since there isn’t any statistically significant differences between the test conditions, we are assured of the validity of the data. In contrast, since the after-update metrics are design to capture the impact of topic stability on a user’s ability to comprehend topics and recall articles, if maintaining topic stability is important and our system is effective, we expect that our method will result in better scores than those using the LDA method. Our study results have also confirmed this. With our method, the topic keywords are more coherent, which makes it easier for a user to understand a topic and select a correct topic label for it (0.688 with nTMU v.s. 0.338 with LDA). Since the top articles selected by the system are also more topic-relevant, which makes it easier for a user to pick the correct topic label based on top articles (0.875 with nTMU versus 0.388 with LDA). Finally, keeping the topic model stable can also significantly improve a user’s chance to relocate an article (0.775 with nTMU v.s. 0.525 with LDA). As shown in Figure 7, our system outperformed the baseline in all the after-update evaluation dimensions. The differences are all statistically significant with p < 0.001 based on independent sample t-test. Therefore, it again demonstrated that our nTMU system can provide users a non-disruptive topic model update process, compared to the real world system that based on standard LDA.

In addition, as shown in Figure 8, our system also outperformed the baseline in three out of the four subjective evaluation dimensions. For example, with our system, the users think that they can understand the topics better. It is also easier for them to relocate an article they found before. They are also more likely to recommend the system to their friends.

![Objective Metrics Before Update](image)

**DISCUSSION**

**Scalability**

For today’s constantly expanding document collections, a topic modeling system that supports topic-centric navigations and explorations of the collections needs to be quickly updated when new documents arrive in an online fashion. Therefore, scalability is a critical issue for any topic modeling update system. For example, online LDA [9] incrementally updates the topic model without requiring a full scan of the entire dataset. However, as we mentioned before, online...
LDA suffers from topic stability problem that undermines the system’s usability. Unlike online LDA, our nTMU system can not only update the topic model without a full scan of the entire dataset when new content arrives, but also maintain a good topic model stability based on user feedback. A user has both the freedom to choose a topic that he/she wants to keep stable and specify a set of documents that are supposed to have the same label after the model is updated. Thus, the user can control the size of old documents that are trained with the new data. The Feedback Processor in nTMU system can consume users’ feedback and generate document-level constraints accordingly. Therefore, at the expense of adding a subset of old documents and their constraints, the nTMU system can achieve the same scalability as online LDA, but with a better stability performance. The only issue is that online LDA uses variational inference method while nTMU’s cLDA component utilizes Gibbs sampling for inference. It is not very straightforward to develop a variational inference version of cLDA to incorporate document-level constraints. Therefore, it needs further investigation for cLDA to achieve same converge speed as online LDA.

Personalization
In addition to a non-disruptive topic model update, the proposed constrained topic modeling, cLDA, can also be used for personalized topic modeling. In practice, different users may have different expertise or interests. For example, let us assume two users want to explore a newsgroup dataset, such as New York Times news article dataset. The first user who is a sports fan may be looking for a fine-grained topic exploring all sports articles. As a result, this user will likely add a cannot-link between Baseball documents and Hockey documents. However, the second user who has little interest in sports may just want a coarse-grained topic for sports articles. Therefore, this user will likely add a must-link between Baseball and Hockey documents. Although these two users are training the same dataset with cLDA, they will get different topic summarizations based on their different feedback. The personalized topic model would be very useful in practice by providing individual users the freedom to explore the dataset to match their own tastes and preferences. We also believe personalization along with user interaction is an important future research direction.

CONCLUSION
Existing topic model update methods (e.g., online LDA) have long neglected an important issue which has a significant impact on the usability of a topic modeling system: How to maintain topic model stability to minimize end user disruption? In this paper, we presented an approach to topic model update which directly addresses topic model stability. Included in this approach are 1) a novel constrained LDA algorithm cLDA which enables LDA to incorporate general pairwise document constraints that none of the existing methods are capable of handling effectively, and 2) a new user-directed non-disruptive topic model update system nTMU which collects user feedback, converts them to pair-wise document constraints and employs cLDA to achieve a smooth topic model transition. Evaluation results on both simulation experiments and user studies indicated that our approach significantly outperforms baseline systems in achieving high topic model stability while still maintaining high topic model quality.

ACKNOWLEDGMENTS
This work was supported by DARPA contract D11AP00268.

REFERENCES