Incorporating Metadata into Dynamic Topic Analysis

Tianxi Li
Stanford University
Stanford, CA 94305
tianxili@stanford.edu

Branislav Kveton
Technicolor
Palo Alto, CA 94301
Branislav.Kveton@technicolor.com

Yu Wu
Stanford University
Stanford, CA 94305
yuw2@stanford.edu

Ashwin Kashyap
Technicolor
Palo Alto, CA 94301
Ashwin.Kashyap@technicolor.com

Abstract

Everyday millions of blogs and micro-blogs are posted on the Internet These posts usually come with useful metadata, such as tags, authors, locations, etc. Much of these data are highly specific or personalized. Tracking the evolution of these data helps us to discover trending topics and users' interests, which are key factors in recommendation and advertisement placement systems. In this paper, we use topic models to analyze topic evolution in social media corpora with the help of metadata. Specifically, we propose a flexible dynamic topic model which can easily incorporate various type of metadata. Since our model adds negligible computation cost on the top of Latent Dirichlet Allocation, it can be implemented very efficiently. We test our model on both Twitter data and NIPS paper collection. The results show that our approach provides better performance in terms of held-out likelihood, yet still retains good interpretability.

1 Introduction

Topic evolution analysis has become increasingly important in recent years. Such analysis on social media and webpages could help people understand information spreading better. In addition, it also provides ways to understand latent patterns of corpus, reduce effective dimensionality and classify documents and data. Meanwhile, reseachers manage to fit various types of data into the topic model. For example, image segmentations was modeled as topics in Feifei et al. [6]. User behaviors were also modeled as topics as in Ahmed et al. [1] In such circumstances, topic evolution gains other practical values. For example, knowing the evolution of people's behaviors could improve the performance of item recommendations and advertising strategy. In addition, dynamic feature extraction might also provide richer user profile.

In various applications, one might want to harness metadata for different purposes. When metadata contains useful information for the topic analysis, it can help enhance the precision of the model. For instance, authorship can be used as an indicator of the topics in scientific paper analyzing [14]. Citations can also help reveal the paper's topics [9]. In behavior modeling, metadata such as user_id could be used for personalized analysis.

In this paper, we propose topic evolution model incorporating metadata effects, named metadata-incorporated dynamic topic model (mDTM). This is a flexible model effective for various metadata types and evolution patterns. We demonstrate its applicability by modeling the topic evolution of Twitter data, where we use hashtags as the metadata. This problem is particularly challenging because of the limited length of tweets and their non-standard webish style. Later we use authors as the metadata to run a dynamic author-interest analysis on the NIPS corpus.

The paper is organized as following. Section 2 gives a brief description of backgrounds and prior work. Our model is introduced in Section 3. Finally, the illustrative examples of topic evolution analysis are presented in Section 4.

2 Notations and Related Work

In this paper, the corpus is denoted by D, and each document d in corpus consists of N_d words. Each word

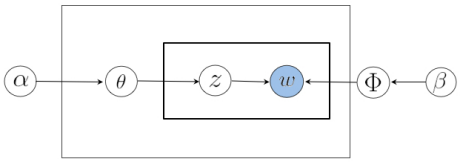
w is an element in the vocabulary of size V. There are K different topics associated with the corpus. Assume the words in the same document are exchangeable. The case of interests is when the documents have other special metadata. We use h to represent the metadata. Assume $h \in H$, where H is the domain of h. For instance, when h is hashtag of a tweet, H can be all the strings of hashtags. Let h_d be the instantiation of $h \in H$ at document d. Now with above notations, we can define the topics to be probability distributions over the vocabulary. Let $p(w|z)$ be the probability of word w appears when the topic is z, then topic z is represented by a V-vector corresponding to a multinomial distribution:

$$(p(1|z), p(2|z) \cdots, p(V|z)).$$

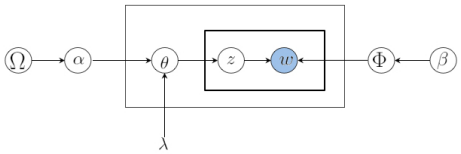
Latent Dirichlet Allocation proposed by Blei et al.[4], is one of the most popular models for topic analysis. LDA assumes the documents are generated by the following process:

(i) for each topic $k = 1, \cdots, K$:
Draw word distribution by $\phi_k \sim \text{Dir}(\beta)$.
(ii) for each document d in the corpus :
(a) Draw a vector of mixture proportion by $\theta_d \sim \text{Dir}(\alpha)$.
(b) for each word position j in d :
(b1): Draw a topic for the position by $z_{d,j} \sim \text{mult}(\theta_d)$.
(b2): Draw a word for the position by $w_{d,j} \sim \text{mult}(\phi_{z_{d,j}})$.

In the process, α is a K-vector and β is a V-vector. θ_d's are K-vectors characterizing a multinomial distribution of the topic mixture for each document d. α and β are called hyperparameters. Throughout this paper, we will use $w_{d,j}$ and $z_{d,j}$ to denote the word and topic in position j of document d respectively. $\text{Dir}(\alpha)$ denotes the Dirichlet distribution with parameter α, and $\text{mult}(\theta)$ denotes the 1-trial multinomial distribution.The model structure of LDA is shown in Figure 1(a), where we use Φ to represent the vectors $\{\phi_1 \cdots \phi_K\}$. In most cases, α and β are chosen to be symmetric vectors. There is work (Wallach et al.[16]) showing that LDA with asymmetric hyperparameters can outperform symmetric settings. For a K-vector $\Omega = (\Omega_1, \cdots, \Omega_K)$, they added the prior of α so as $\alpha \sim \text{Dir}(\Omega)$ can connect LDA to mixture model given by Hierarchical Dirichlet Process (HDP), which



(a) Original LDA graphical structure



(b) Asymmetric LDA with priors.

Figure 1: Graphical structures of LDA models.

is a nonparametric prior allocation process proposed in Teh et al.[15]. Adding the extra prior Ω, the graphical structure of LDA can be represent by Figure 1(b).

As mentioned in Section 1, we would like to take metadata into consideration as in [14]. Labeled-LDA (Ramage et al.[13]) provides another method to use metadata, requiring topics to be chosen from a subset of the label set, where the labels can incorporate certain kinds of metadata. Statistically speaking, this works like adding sparse mixture priors. In Ramage et al.[12], labeled-LDA is used for Twitter data. However, there is no natural way to create labels for different metadata. Such models assume specific generative process for metadata influences, which often limits the model to certain metadata. However, in our model, the impacts of metadata are modeled by empirical estimation rather than a specific probabilistic process, which makes it valid generally.

On the other hand, we need dynamic models to analyze topic evolution. The dynamic topic model (DTM) proposed by Blei works well on the example of *science* papers [3]. However, its logistic Gaussian assumption is no longer conjugate to multinomial distribution, which makes the computation inefficient. Moreover, it is an offline model that needs the entire corpus at one time, thus not suitable for stream data. Iwata et al.[10] uses multi-scale terms to incorporate time relation. This method can be very complicated in some cases and therefore infeasible for large scale datasets. But the

idea of modeling the relation by hyperparameter is really effective in many problems. In [1], a time-varying user model (TVUM) is proposed. It considers users' behaviors over time, and connects different users by the general sampling process. Here we can take a different viewpoint of TVUM. Note that when we take each user's identity as the metadata, TVUM is actually using metadata for interests evolution. In this aspect, it can be seen as a special case and also a starting point of our model.

In the next section, we begin from another view of LDA model, and generalize it to incorporate metadata.

3 Metadata-incorporated Dynamic Topic Model

3.1 Motivation: Define LDA via Markov Chains

The inference of LDA can be done through MCMC sampling. The sampling inference algorithm was proposed in Griffiths et al.[8]. But to understand how LDA works, we need to use the smoother version shown in Figure 1. It is shown in [15] that LDA in this case limits to a HDP mixture model as $K \to \infty$. Thus we will introduce a few more notations and start from HDP aspect of LDA. In the rest of the paper, we will use subscript d to denote relevant variables associated with document d, subscript k to denote the variable associated with topic k, and w to denote variables associated with word w. Following this style, m_k is defined as the number of documents containing words generated from topic k and $\mathbf{m} = (m_1, m_2, \cdots, m_K)$, while $n_{d,k,w}$ is the number of words w in document d that is from topic k. We further use $\cdot$ to denote summation over a specific variable, so $n_{\cdot,k,w}$ is the number of occurrence of words w being drawn from topic k and, $\mathbf{n}_k = (n_{\cdot,k,1}, n_{\cdot,k,2}, \cdots, n_{\cdot,k,V})$. In addition, $n_{d,k,\cdot}$ is the number of words in d which are associated with topic k. When we want to discuss the variables at time t, we use the superscript x^t to represent the variable x in the model of time t. So we have $\mathbf{m}^t = (m_1^t, m_2^t, \cdots, m_K^t)$,$\mathbf{n}_k^t = (n_{\cdot,k,1}^t, n_{\cdot,k,2}^t, \cdots, n_{\cdot,k,V}^t)$. When we focus on discussions at one time slice, which is clear in context, we will ignore the superscript t.

According to the discussion in [15] and the mechanism of Gibbs sampler, we can equivalently define the LDA inference of topic z (for each position of each document) as the stationary distribution of a Markov chain with the transition probability given in Formula (1), in which the superscript $-(d,j)$ refers to the originally defined variables without considering the position j of document d, and $\mathbf{w}^{-(d,j)}, \mathbf{z}^{-(d,j)}$ are the variables of the words and topics of the corpus except the ones at position j in document d, that is, $w_{d,j}$ and $z_{d,j}$ respectively.

The interpretation of this transition probability is that the Markov chain evolves with the following two patterns to arrive new topic states in the document. (i) Choose a topic proportional to the existing topics distribution within the document. This means it tends to keep the topic of each position consistent with the document contents. (ii) With certain probability, it might choose a topic ignoring the existing contents of the document. However, this choice is based on the popularity of topics over the entire corpus. This is a reasonable assumption in many circumstances, and we believe this could explain the power of LDA.

$$P(z_{d,j} = k|\mathbf{w}^{-(d,j)}, \mathbf{z}^{-(d,j)}) \propto (n_{d,k,\cdot}^{-(d,j)} + \lambda \frac{m_k + \Omega_k}{\sum m_k + \Omega_k}) P(w_{d,j}|\phi_k). \quad (1)$$

3.2 Generalization: mDTM

Assume the corpus has metadata h. Our basic assumption is that metadata is a good indicator of topics for each document. For example, a tweet with hashtag "#Microsoft" is much more likely to talk about technology rather than sports. Nearly all the previous works involving a certain type of metadata rely on this assumption. We first define the preferences of metadata over time as a vector function of t and h, $g(h,t) = (g_1(h,t), g_2(h,t), \cdots g_K(h,t))$. The kth element $g_k(h,t)$ is the preference of h to topic k at time t. Since we want to build a dynamic model for topic evolution, we can learn $g(h,t)$, and turn it into another impact on top of the evolutionary effects of β and Ω. Motivated by the definition of LDA given by (1), we define the mDTM inference at a fixed time slice to be the stationary distribution of a Markov chain with transition probability

$$P(z_{d,j} = k|\mathbf{w}^{-(d,j)}, \mathbf{z}^{-(d,j)} \propto (n_{d,k,\cdot}^{-(d,j)} + g_k(h_d,t) + \lambda \frac{m_k + \Omega_k^t}{\sum m_k + \Omega_k^t}) P(w_{d,j}|\phi_k^t). \quad (2)$$

The modification we make has exact effects that we want to incorporate into (1). In addition, this process

provided by mDTM is simple and does not incure too much computation, as shown in the Section 3.4. We only focus on the case where there is only one metadata variable in our discussion. There might be the case that more than one metadata variables are associated with the corpus. For instance, we might have timezone and browser for web log. In this case, we can simply model the effects as additive and estimate the function $g(h,t)$ separately for each metadata variable. Then everything we discuss here could be used for multiple metadata variable case. As we will propose different evolution patterns for the parameters in later sections, here we introduce notation f_Ω and f_β as the evolution functions of Ω and β. Now taking the time effects of evolution into consideration, the entire evolution process of mDTM is as follows:

(1) $t = 0$: initialize the model by LDA.
(2) For $t > 0$:
  (a) Draw Ω_t according to the model of $t-1$ by $\Omega_t = f_\Omega(t-1)$.
  (b) For each topic $k = 1, \cdots, K$: Draw β_k by $\beta_k^t = f_\beta(t-1)$.
  (c) With the current Ω_t and $\{\beta_k^t\}_{k=1}^K$, implement the inference for the process described by equation (2).

We model the evolution of all the effects by separable steps, so the model can be updated when data in new time slice arrives, which makes it possible for stream data processing and online inference. It is very flexible to adjust mDTM for different types of metadata, with different properties as we do not have to assume specific properties of the metadata. Notice that though we generalize the Markov chain definition of LDA to mDTM, we haven't shown the existence of the stationary distribution or limiting behavior of the chain. To address this issue, we can check mixing of the chain, so as to know if the inference is valid. In all of our experiments, such validity is observed. For details and methods about mixing behavior of Markov chains, we refer to Levin et al. (2009) [11].

The evolution patterns $f_\Omega(t)$, $f_\beta(t)$ and $g(h,t)$ are addressed in Section 3.3. Then we give the inference steps of mDTM in Section 3.4.

3.3 Evolution Patterns of mDTM

Now we describe how to model g. Assume metadata is categorical which is the case we normally encounter in applications. Similar methods can be used to choose f_Ω and f_β, so we will only discuss the evolution pattern for $g(h,t)$ in detail. We use $\tilde{n}_{k,h}^t$ to denote the number of the topic k that occurs in all documents having metadata h at time t.

3.3.1 Time-decay Weighted Evolution

We can just take g_k as the weighted average number of topics k appearing in documents with metadata h, using the weights decays over time. This represents our belief that the recent information is more useful to predict the preference. Thus,

$$g_k(h,t) = \sigma \sum_{s<t} \kappa^{t-s} \tilde{n}_{k,h}^s, \tag{3}$$

where σ is a scalar representing the influence of the metadata. This is a straightforward way to encode the evolution pattern, and the computation is very easy.

3.3.2 Bayesian Posterior Evolution

For each $h \in H$, we assume there is a preference vector for h to be $\mu_h^t = (\mu_{1,h}^t, \mu_{2,h}^t, \cdots \mu_{K,h}^t)$ which is a vector in the $K-1$ dimensional simplex, with $\mu_{k,h}^t \geq 0$ for $k = 1 \cdots K$. Then the realization of choosing topic for any $h \in H$ can be seen as $(\tilde{n}_{1,h}^t, \tilde{n}_{2,h}^t, \cdots \tilde{n}_{K,h}^t) \sim \text{Multinomial}(\tilde{n}_h^t, \mu_h^t)$, the $\tilde{n}_h^t$-trial multinomial distribution which is sum of $\tilde{n}_h^t$ independent trials from $\text{mult}(\mu_h^t)$, where $\tilde{n}_h^t$ is the total number of observations of h over the corpus. So we can take the Bayesian estimation by adding a Dirichlet prior by the process:

$$\mu_h^t \sim \text{Dir}(\zeta^{t-1} \cdot \hat{\mu}_h^{t-1})$$

$$(\tilde{n}_{1,h}^t, \tilde{n}_{2,h}^t, \cdots \tilde{n}_{K,h}^t) \sim \text{Multinomial}(\tilde{n}_h^t, \mu_h^t)$$

In such settings, we can choose the posterior expectation as the estimator, which is

$$\hat{\mu}_{k,h}^t = \frac{\tilde{n}_{k,h}^t + \zeta^{t-1} \cdot \hat{\mu}_{k,h}^{t-1}}{\sum \tilde{n}_{k,h}^t + \zeta^{t-1} \cdot \hat{\mu}_{k,h}^{t-1}}. \tag{4}$$

ζ is a scalar representing influence of the prior, which is the Bayesian estimator from previous time. Then let

$$g_k(h,t) = \sigma \hat{\mu}_{k,h}^t$$

in the process, where σ is a scalar representing the influence of the metadata. Such evolution pattern is very simple and smooth and it adds almost no additional computation cost.

This pattern actually also assumes there is a hyperparameter in each time t, which is μ_h^t. Rather than setting it beforehand, we impute the estimate for such hyperparameters by inference from the model. This is the idea of empirical Bayes method. In particular, one could notice that if there is no new data for h after time t, Bayesian posterior evolution would remain the same, while the time-decay evolution gradually shrinks g to zero.

3.3.3 Sparse Preference

In certain cases, we might constrain each document to only choose a small proportion of K topics. Our method to achieve this goal is to force sparsity on the topic choosing process. We can take the occasional appearance of most of the topics as noise, then implement a thresholding to denoise and get the true sparse preference. Define the function $S(a, \epsilon)$ as hard or soft thresholding operator where ϵ is the threshold. Then we can process each variate of the vector resulting from the previous evolution pattern by S, resulting a sparse vector. The soft and hard thresholding functions are defined respectively as

$$S_{\text{soft}}(a, \epsilon) = \text{sign}(a) \cdot \max\{|a| - \epsilon, 0\}$$

$$S_{\text{hard}}(a, \epsilon) = \text{sign}(a) \cdot I\{|a| > \epsilon\}$$

3.3.4 Choice of f_Ω and f_β

Similar evolution patterns for f_Ω and f_β can be chosen. With certain variable changed according to the settings. For f_Ω, one could use m_k^t to replace $\tilde{n}_{k,h}^t$ in (3) and (4). The evolution pattern of β can be derived via replacing $\tilde{n}_{k,h}^t$ in (3), (4) by $\mathbf{n}_k^t$.

3.4 Inference

As mentioned previously, mDTM can be seen as a generalization of TVUM. Suppose now we take user-ID as the only metadata which is categorical, and assume that each document belongs to a certain user-ID, then the parameters associated with each category of the metadata in mDTM become the parameters associated with a particular user. Furthermore, suppose that the documents are the browsing history of a user, then mDTM will be modeling the user's browsing behavior over time. In particular, if we use the time-decay average discussed in Section 3.3.1, the resulting model is equivalent to TVUM after some simple derivation [1]. This connection gives an vivid example about how to transform a specific problem into the settings of mDTM.

The time-varying relationship of mDTM can be represented by a separable term, thus we can incorporate the time-related term and the topic modeling for a fixed time separately. For a fixed time unit, the inference process by Gibbs sampling is easy to derive. Since the special case mentioned before is equivalent to TVUM, we derive the inference process by analogy to that shown in [1]. Suppose now we have the model in previous time $t-1$, the whole process for inference of t is as follows:

(i) Update the new hyperparameters Ω^t and β^t for time t according to the chosen evolution pattern.

(ii) Initially set the starting values. We could set the initial value of α as Ω^t. The initial values for the counts at time t, that is $m_k^t, n_{\cdot,k,w}^t, n_{d,k,\cdot}$, can be computed after randomly choosing topics for each documents and words.

(iii) For each document d, compute the $g(h_d, t)$ according to the chosen evolution pattern in Section 3.3. Then sample the topic for each word position j by the formula

$$P(z_{d,j} = k | w_{d,j}, others)$$
$$\propto (n_{d,k,\cdot}^{-(d,j)} + g_k(h_d, t) + \lambda\alpha_{d,k}) \cdot \frac{n_{\cdot,k,w_{d,j}}^{t,-(d,j)} + \beta_{k,w_{d,j}}^t}{\sum_{w=1}^V n_{\cdot,k,w}^{t,-(d,j)} + \beta_{k,w}^t}.$$

(iv) Sample m_k^t from the Antoniak distribution [2] for Dirichlet process.

(v) Sample α from $\text{Dir}(\mathbf{m}^t + \Omega^t)$. And repeat (iii)-(v).

4 Experiments

To illustrate the model, we conducted two experiments in which metadata is used for different purposes. We first use mDTM on Twitter data for topic analysis, in which we take hashtags as the metadata. In the second experiment, we fit our model on the NIPS paper corpus and try to find information for specific authors,

[1] Actually, TVUM has a slightly different way to define the evolution of Ω, which defines the average in different scales of time, such as daily, weekly and monthly average.

which we use as metadata. For conciseness, we mainly discuss the former in detail, because Twitter data is special and challenging for topic analysis. In the NIPS analysis, on top of the similar results as in previous dynamic models such as DTM, we can extract authors' interests evolution pattern, which would be the main result we present for that experiment.

4.1 Twitter Topic Analysis

4.1.1 Data and Model Settings

The Twitter data in the experiment is from the paper of Yang and Leskovec [18]. We use the English tweets from July 1st, 2009 to August 31st, 2009. For each of the first three days, we randomly sampled 200,000 tweets from the dataset. And around 100,000 tweets were sampled for each of the rest days. We considered the hashtags as the metadata in the experiment. After filtering stop words and ignoring all words appearing less than 10 times, a vocabulary of 12,000 words is selected by TF-IDF ranking. The number of topics was fixed at 50. In mDTM, time-decay weighted average was used for f_Ω and f_β. We simply set $\kappa = 0.3$. Bayesian posterior evolution was used for hashtag and soft-thresholding discussed in Section 3.3.3 was used for the evolution of $g(h_d, t)$. The parameters λ and ϵ are tuned according to the prediction performance in the first week, which is discussed in Section 4.1.4.

Our main interest is how topic popularity and contents change over time.

4.1.2 Topic Popularity Evolution

As can be seen from Equation (2), all the documents with different metadata share the common term $\mathbf{m}$, thus $\mathbf{m}$ can be interpreted as community popularity of topics, separated from the specific preference of metadata. This shows which topics are more popular on Twitter. Figure 2 gives popularity over the two months of some topics, which we labeled manually after checking the word distributions of the topics.

4.1.3 Topic Contents Evolution

Since each topic is represented by a multinomial distribution, one could find out the important words of the topics. Table 1 gives the content evolution of the topic *US politics*. It can be seen that *obama* and *tcot*[2] are very important words. However, words about "Sarah Palin" were mainly popular in July, while the words about "Kennedy" and "Glenn Beck" became popular only at the end of August, all of which roughly match the pattern of search frequencies given by Google Trends[3].

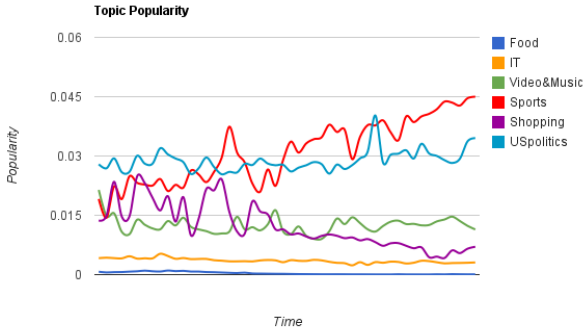


Figure 2: Topic Popularity on Twitter given by mDTM, over the period of July and August 2009.

Table 1: Content evolution of the topic *US politics*

Jul 4	Jul 27	Aug 12	Aug 30
palin	obama	health	kennedy
obama	palin	care	care
sarah	tcot	obama	ted
tcot	sarah	tcot	health
president	president	bill	obama
alaska	healthcare	healthcare	bill
al	health	reform	beck
honduras	obamas	insurance	glenn
governor	speech	president	public
palins	alaska	town	president

4.1.4 Generality Performance

There is no standard method to evaluate dynamic topic models, thus we take a similar approach as in [3] to show the prediction performance on the held-out data. In each day, we treat the next day's data as the held-out data and measure the prediction power of the model.

We compare mDTM with two LDA models without metadata as in [16] to illustrate the improvement provided by metadata modeling[4]. Without metadata, in the first model, we use LDA on the data of each day for inference, and call this model indLDA. The prob-

[2]The word *tcot represents "top conservatives on twitter"*.

[3]We don't provide the results from Google Trends due to the limited space. The search frequencies can be found at www.google.com/trends/

[4]We didn't compare directly with DTM. This is because DTM cannot be used in an online way, thus it cannot serve our purpose.

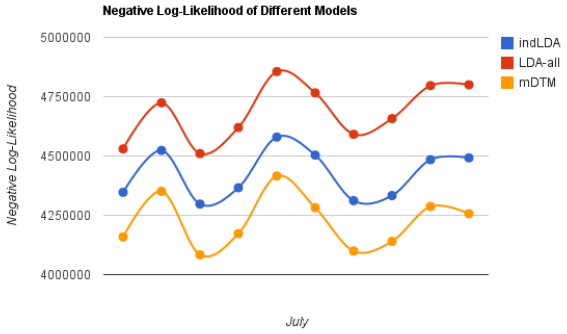


Figure 3: Negative log-likelihood during the early period (July 4th - 10th).

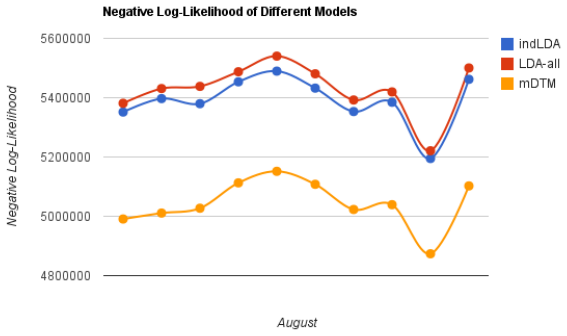


Figure 4: Negative log-likelihood during the end period (Aug 21st - 30th).

lem here is that there is no clear association for topics between days. In the second one, we try to overcame this drawback and take all the data of previous days for inference, which we call LDA-all. It would take nearly two months' data at the end of the period. This would be too much for computation. Thus we further subsampled the data from previous days for LDA-all in the end of the period to make it feasible. LDA-all will not serve for our purpose and so the main interests would be comparing indLDA and mDTM. We report the negative log-likelihood on the held-out data computed as discussed by Wallach et al[17] over the beginning period (July 4th - 10th) and the end period (Aug 21st - 30th). We estimate mDTM as discussed before, but computed the negative log-likelihood ignoring the metadata of the held-out data, thus this gives us an idea of how metadata can improve the modeling for general documents, even those without metadata. There is λ in all of the three models. We tune it and the thresholding parameter ϵ by achieving the best log-likelihood in the first week. Figure 3 and 4 illustrate the results.

As is shown, mDTM always performs better than the other two models. This is not surprising because mDTM has more flexible priors. It is interesting that LDA-all performs even worse than indLDA. This is different from the results of [3]. It might be explained by the differences between Twitter data and scientific paper data. Twitter's topic changes so frequently, but LDA-all takes all the previous days together, which undermines its power.

4.1.5 Effects of Metadata

In Twitter analysis, the topic preference of a specific hashtag is not of interests. However, incorporating hashtags can improve the preformance. On average, there are roughly 10 precent of the tweets having hashtags. But such a small proportion of metadata is able to provide important improvement of the whole corpus, even for the tweets without hashtags. We compute the held-out log-likelihood, for both the model inferred without using hashtags as metadata (called DTM_noTag) and the model mDTM using hashtags. mDTM_noTag can be seen as TVUM with one user. Note that when compute the held-out log-likelihood. We take the improvement of hashtags as the improvement of negative log-likelihood

$$(-\text{loglik})_{\text{DTM\_noTag}} - (-\text{loglik})_{\text{mDTM}}.$$

Figure 5 illustrates the improvement of negative log-ikelihood on the held-out data over the period. It can be seen that on average, incorporating hashtags as metadata does improve the performance. And this improvement tends to grow as time goes. This might results from the better estimation of most of the metadata preference.

4.1.6 Running Times

Here we present a comparison for timing of mDTM and indLDA. Both were implemented in C++, running under Ubuntu 10.04, with Quad core AMD Opteron Processor and 64 GB RAM. We list average running times (rounded) in Table 2. indLDA is the average time on 10 days (July 4th - July 13th) with 600 sampling iterations each day. mDTM-1 is the mDTM running on the same data with 600 sampling iterations. Since mDTM could inherit information from previous time, we found 300 iterations (or less) are enough for valid inference. Thus we use mDTM-2 to denote mDTM with 300 it-

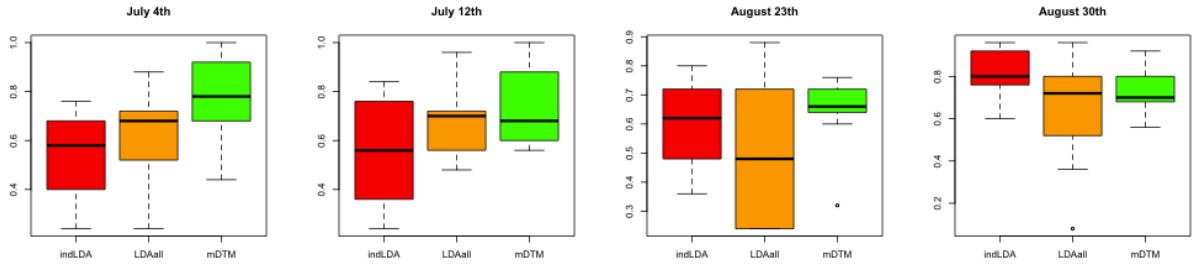


Figure 6: The human evaluation ACR for the three models. Each box is a value distribution of average correct ratios for 10 topics of the corresponding model on certain day.

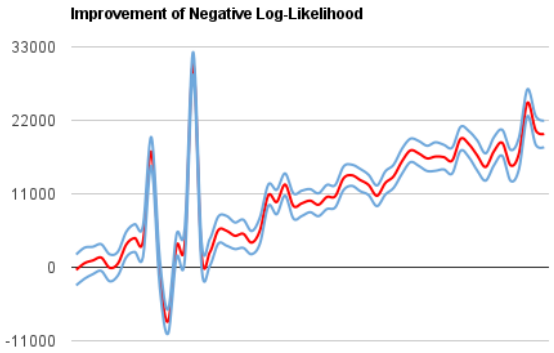


Figure 5: The improvement of negative log-likelihood via hashtags over the period. The red lines are the improvement of $-log(likelihood)$ computed by importance sampling. The blue lines are the intervals at each estimation point given by 2 standard deviations of the sampling.

erations. It can be seen that mDTM is much faster than LDA.

Table 2: Running times of three different models

indLDA	mDTM-1	mDTM-2
58min 41s	67min 13s	39min 24s

4.1.7 Interpretability

The previous sections show that mDTM is better than indLDA and LDA-all at generality. However, the interpretability of the topics is also of interests. Chang et al. [5] revealed that models with better performance on held-out likelihood might have poor interpretability. Here we use the method in [5] to ask humans to evaluate the interpretability. We choose July 4th (the first day after three initial days), July 11th (after one week of July 4th), August 30th (the last day) and August 23th (one week before the end) for experiments. However, news would be difficult for people to recognize after more than one year, so we only chose 10 stable topics from each model[5]. For every topic in each model, we construct the list by permuting top 15 words for that topic together with 5 intruder words which have low probability in that topic but high probability in some other topics. Suppose we have S subjects, then for each topic k, we compute the average correct ratio (ACR)

$$ACR(k) = \sum_{s=1}^{S} C(s,k)/(5S),$$

where $C(s,k)$ is the number of correct intruders chosen by subject s for topic k. We conducted a human evaluation experiment on Mechanical Turk with 150 subjects in total. Figure 6 shows the boxplot of ACR distribution within each model on each day.

It can be seen that mDTM does not lose much interpretability despite its better prediction performance, which is different from the observations in [5]. We hypothesize that this is due to the impacts of metadata.

4.2 NIPS Topic Analysis

In this section, we illustrate a different application of mDTM, that is, to extract specific information of metadata.

[5] We count the number of different words in the top 20 words list on two consecutive days, and sum such numbers during the whole period together. A larger sum number means that the topic word list changes frequently. Then we select 10 topics that are the most stable. The topics in different time are not associated for indLDA and LDA-all. We connect a pair of topics between two consecutive days if they have the most overlap on top 20 words.

4.2.1 Data and Model Settings

The dataset for this experiment contains the text file of NIPS conference from 1987 to 2003 in Globerson et al[7][6]. We only use the text of the paper and take the authors as the metadata. The papers in 1987-1990 were used for the first time unit to initiate the model, and each year after that was taken as a new time unit. The preprocessing details of the data can be found on the website. We further deleted all the numbers and a few stop words. The resulting vocabulary has 10,005 words. The number of topics K was set as 80. Bayesian posterior evolution was used for g and f_β. And f_Ω was set as time-decay weighted average with $\kappa = 0.3$. We don't use sparse preference in this example. The parameter λ is again tuned by log-likelihood as before.

4.2.2 Author-topic interests

As before, we could see the topic contents and popularity trends over time. Here, we only focus on the special information given by metadata in this experiment. When taking authors as metadata, an interesting information result provided by mDTM is the interests of authors, similar to the results of [14]. Figure 7 shows the results given by mDTM for author "Jordan_M". The height of the red bars represents the $\hat{\mu}_{k,h}$ from Equation (4) for h="Jordan_M", which can be interpreted as the topic interests according to the past information.

It can be seen that authors' favorite topics remained nearly the same during the three years, though the interest level for individual topics varied. When we know the topic interests of the author, we can further investigate the contents of the user's favorite topics, which is a way to detect the user's interests that would be useful in many applications. Table 3 shows the top 10 words for four topics of significant interests to "Jordan_M" in 1999, according to the result in Figure 7. We can roughly see they are mainly about "clustering methods", "common descriptive terms", "graphical models" and "mixture models & density estimation", which is a reasonable approximation.

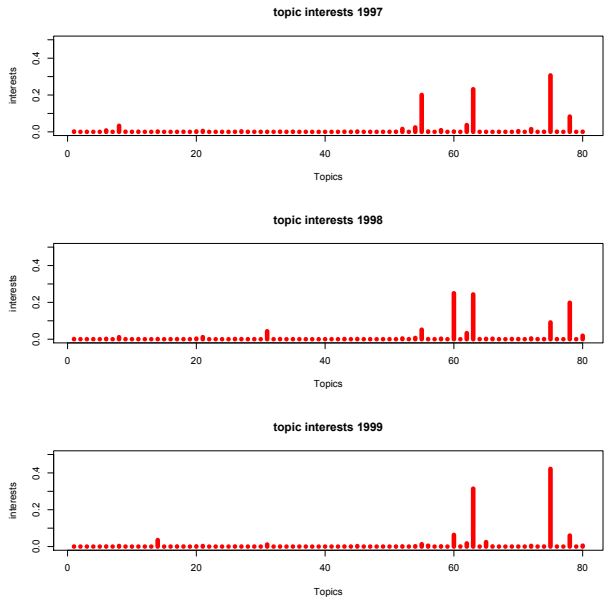


Figure 7: Topic preference from mDTM of 80 topics, for author "Jordan_M" in 1997, 1998 and 1999.

Topic 60	Topic 63	Topic 75	Topic 78
clustering	function	variational	model
clusters	number	nodes	data
information	figure	networks	models
data	results	inference	parameters
algorithm	set	gaussian	likelihood
cluster	data	graphical	mixture
feature	case	field	distribution
selection	based	conditional	log
risk	model	jordan	em
partition	problem	node	gaussian

Table 3: Four significant topics for "Jordan_M" selected from Figure 7 in 1999.

5 Conclusion

In this paper, we have developed a topic evolution model that incorporats metadata impacts. Flexible evolution patterns are proposed, which can be chosen according to properties of data and the applications. We also demonstrate the use of the model on Twitter data and NIPS data, revealing its advantage with respect to generality, computation and interpretability.

The work can be extended in many new ways. For the moment, it cannot model the birth and death of topics. One way to solve this problem is to use general prior allocation mechanism such as HDP. There has been work using this idea for static models. In addition, the generality and flexibility of mDTM make it possible to build other evolution patterns for hyperparameters, which might be more suitable for specific purposes of modeling.

[6]Data can be found at http://ai.stanford.edu/ gal/data.html

References

[1] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. J. Smola. Scalable distributed inference of dynamic user interests for behavioral targeting. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 114–122, New York, NY, USA, 2011. ACM.

[2] C. E. Antoniak. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

[3] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 113–120, New York, NY, USA, 2006. ACM.

[4] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[5] J. Chang, J. Boyd-Graber, C. Wang, S. Gerrish, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, 2009.

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *CVPR*, pages 524–531, 2005.

[7] A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean Embedding of Co-occurrence Data. *The Journal of Machine Learning Research*, 8:2265–2295, 2007.

[8] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, Apr. 2004.

[9] Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and L. Giles. Detecting topic evolution in scientific literature: how can citations help? In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 957–966, New York, NY, USA, 2009. ACM.

[10] T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda. Online multiscale dynamic topic models. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 663–672, New York, NY, USA, 2010. ACM.

[11] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2009.

[12] D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *ICWSM*, 2010.

[13] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256, Singapore, August 2009. Association for Computational Linguistics.

[14] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[15] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[16] H. Wallach, D. Mimno, and A. McCallum. Rethinking lda: Why priors matter. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981. 2009.

[17] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA, 2009. ACM.

[18] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 177–186, New York, NY, USA, 2011. ACM.