Online Topic Evolution Modeling Based on Hierarchical Dirichlet Process

Tao Ma[1], Dacheng Qu[2], Rui Ma[1], Wei Feng[2] and Kan Li[2]
[1]School of Software, Beijing Institute of Technology, Beijing, China
[2]School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China
Email: {2120141108,qudc,mary,20141001,likan}@bit.edu.cn

***Abstract*—This paper presents a model based on Hierarchical Dirichlet Process (HDP), that automatically captures the evolutionary thematic patterns in texts. Our approach allows HDP to work in an online fashion, such that it can build an up-to-date model for new documents given the old model, without accessing historic data. Since exact calculation is infeasible, we turn to Gibbs sampling to carry out approximate posterior inference. After the topics are found, we can analyze the evolution relationships between time-adjacent topics. Experiments on a real world dataset (Reuters-21578) validate the effectiveness of the model quantitatively, showing its advantage over both OLDA and plain HDP in modeling topic evolution.**

I. Introduction

With the prospect of various social media, there are massive data generated every day, whose contents show strong inherent temporal ordering. These data convey valuable information, for example, we are interested in the topics underlying the documents. Compared with ignoring time information in the data, incorporating it will give more meaningful topics. With time dependency modeled in, evolution relationships between time-adjacent topics can be revealed, and we can better understand how a topic comes into being, develops and diminishes. Besides, in the applications of evolutionary topics, for some limitations such as storage/computing cost or application requirements, sometimes it is more practical and reasonable to process data in an online fashion, and update the model based on current data and previous models, instead of fitting the model on the whole closed dataset.

In topic modeling, probabilistic topic models such as probabilistic latent semantic indexing (PLSI) [1] and latent Dirichlet allocation (LDA) [2], [3] are good approaches to find topics underlying documents. Compared with PLSI, LDA alleviates the problem of overfitting, can be used to generate unseen documents and easily incorporated into other models. The idea behind LDA is that each document is a mixture of topics, and a topic is a multinomial distribution over the vocabulary. One main assumption underlying LDA is the exchangeability of both documents and words in a document. In LDA, each document is generated as follows: (1) select a multinomial distribution over topics for the document; (2) for each word in the document, first select the topic which it comes from, then generate the word. Both the document-topic distribution and topic-term distribution are placed a Dirichlet distribution prior.

Dirichlet process (DP) [4] is a nonparametric Bayesian model often used to do clustering. A Dirichlet process mixture model can be represented as an infinite mixture model, where the number of mixture components is taken to infinity. In DP mixture model, there is no need to give the number of components ahead, and the model will automatically select the number of components that best describes the data. Theoretically there can be infinite components, however, only enough will be exhibited. Generally there are three equivalents of DP, that is, Polya urn scheme, the stick-breaking construction and the Chinese restaurant process, and using these representations, efficient algorithms can be developed to do posterior inference. Though Dirichlet Process is useful, it will meet difficulty when sharing mixture components among groups, where each group is modeled as a Dirichlet process mixture model. Hierarchical Dirichlet Process (HDP) [5] is proposed to solve this problem by introducing another Dirichlet process, that is, let the base measure for the child Dirichlet processes itself be distributed according to a Dirichlet process, hence the base measure is discrete, and mixture components can be shared among groups.

Topic evolution modeling is still a tricky problem, though much research has been done. There are three challenges in this problem. The first is how to model topics at a time slice. There are topics shared by all the documents, and each document exhibits one or more topics. The second challenge is how to model the time dependencies between topics. There is strict ordering between topics, for example, a topic may be developed from a previous one (or merged from a few topics), and it can proceed just like how it got here. The last challenge is how to determine the number of topics at each time slice. We know that topics are emerging, evolving and dying all the time, so there doesn't exist a fixed number for topics. Assuming fewer or more topics will prohibit us from discovering the real underlying topics, and the best way is to automatically determine the number of topics.

In this paper we present a model based on HDP that can automatically find all the topics at each time slice and capture their relationships. HDP mixture model can be used to do topic modeling if we treat each document as a group. Benefiting from the nonparametric nature of HDP mixture model, we are exempted to specify the number of topics at each time slice which we don't know actually. Since previous topics reflect the contents and trends, therefore, in our approach

978-1-5090-1192-6/16 $31.00 © 2016 IEEE
DOI 10.1109/DSC.2016.60

we build the current model directed by the previous one, using previous topics to construct the prior. We use Gibbs sampling to conduct posterior inference by sampling indicators for words and parameters for mixture components.

Our model differs from some related work in three aspects. First, not like some algorithms that work in an offline fashion, our method processes documents in an online fashion, which brings two advantages. The first advantage is lower storage cost and time complexity since we needn't store historic data and the data is processed each time a slice. The second advantage is enabled by the first, since this will make more applications feasible. Second, our model can find the appropriate number of mixture components automatically, which we have stressed several times. Third, compared with some similar research, our model allows topics to have slight changes in real time. In addition, we can easily track the evolution of a topic, and analyze the evolutionary pattern assisted by this.

We evaluate our model on a real world dataset Reuters-21578 quantitatively. In the experiment, we select perplexity and normalized mutual information as criteria. The results show that our model gives lower perplexity and normalized mutual information than OLDA and plain HDP, since our model can select appropriate number of mixture components even without prior knowledge, while LDA behaves not so good if not provided the right number of components. This illustrates the effectiveness and implies practical applications.

The rest of the paper is organized as follows. In section II we carry out a short review of the related work. Both our model and its posterior inference procedures are presented in detail in section IV, following some preliminaries in section III. We present the experiments in section V, followed by conclusions and future directions in section VI.

II. Related Work

Research on topic evolution can be derived from TDT study [6], and now statistical models using PLSI, LDA or HDP take the lead.

Much work studies topic evolution in an offline fashion. Griffiths and Steyvers [2] first fit a model on the whole data without considering time information, then, they slice the documents over time, and inspect the strength of topics over time. Blei and Lafferty propose a dynamic topic model [7] where they chain natural parameters for adjacent topics over time by a Gaussian distribution, and for a topic the previous natural parameter is the mean for the next one. They first divide the data over time into slices, then fit a topic model with K-components for each slice. However, normal distribution is not a conjugate prior of multinomial distribution, which makes it difficult to inference. TOT [8] takes a different view. Unlike [2], [7], TOT doesn't assume the Markov property, nor pre- or post-discretize the data over time, instead, they treat time as a continuous variable and incorporate it into the model explicitly. Besides a distribution over terms, each topic is also associated with a distribution over time. If a topic spans a long time, it will be associated with a broad time distribution; if a topic just appears momently, a narrow time distribution will be associated. In TOT, topics never change in fact, and they just appear with higher or lower probability over time, which is another characteristic that distinguishes it from other models. Methods using offline fashion train the model using all the documents, and better fitness will be obtained overall, but there is no easy way to extend to new documents consistently.

There is also some work processing the data in an online fashion, where the data is divided into time slices. One observation underlying OLDA [9] is that the hyperparameter of a topic-term distribution can be treated as pseudo-counts of terms' occurrence. So a reasonable way is to directly use the topic-term counts as the hyperparameter of next time slice for the same topic. Furthermore, the hyperparameter can be constructed according to not just the very previous model but in a more complex way if necessary. TopicMonitor [10] is a model based on PLSI. When coming to a new time slice, TopicMoniter will remove both old documents and words that don't appear in new documents, and then fold-in new documents and words into the previous model to obtain an updated model for the current time slice. The feasibility of this method is that the model can be transformed between its two forms, document-based and word-based.

All these above models are based on LDA except [10], and assume a fixed number of topics. In some cases this is acceptable, while sometimes it is not appropriate. The trouble can be avoided by exploiting a nonparametric method, just like what we'll do in this paper. The work most related to ours is [11], [12]. In their work, documents at different time slices share the same infinite set of topics, and the difference lies in the mixing weights. In this way, the topic (its mathematical representation) appearing in time slice 1, may appear in time slice 2 without any change. But in fact, a topic changes slightly at any time, and allowing this capability will model the evolution of topics more finely. Instead of constructing the discrete distribution of topics, we move up a layer, and adapt the base measure. Since new topics are drawn from the base measure, we can make the base measure act as we like. Besides, their models are fitted on the whole data, while ours is updated slice by slice.

III. Dirichlet Process and Hierarchical DP

In this section we'll give a brief introduction to DP and HDP as the preliminaries of our model.

Dirichlet process is a distribution over distributions, and in fact, it is Dirichlet distribution's generalization to infinite dimension. If G is distributed according to a DP, then we write $G \sim DP(\alpha, G_0)$, where G_0 is the base measure, and α is the concentration parameter. The base measure is the mean of G, and the concentration parameter controls how much G varies around the mean. [13] shows that draws from a DP are discrete, and we can represent DP using the so called stick-

breaking construction as:

$$\begin{aligned}\beta_k &\sim \text{Beta}(1,\alpha)\\ \theta_k^* &\sim G_0\\ \pi_k &= \beta_k \prod_{l=1}^{k-1}(1-\beta_l)\\ G &= \sum_{k=1}^{\infty}\pi_k \delta_{\theta_k^*}\end{aligned} \tag{1}$$

From this equation we see that mixture model using DP is an infinite mixture model, and it can be proved that the number of mixture components typically exhibited is approximately $O(\alpha \log n)$, where n is the number of data items observed.

If $\theta_1, \cdots, \theta_n$ are an i.i.d. sequence drawn from G, then the posterior of G can be evaluated as

$$G \mid \theta_1, \cdots, \theta_n \sim \text{DP}\left(\alpha + n, \frac{\alpha}{\alpha+n}G_0 + \frac{\alpha}{\alpha+n}\frac{\sum_{i=1}^{n}\delta_{\theta_i}}{n}\right) \tag{2}$$

We see that the posterior distribution over G is a DP as well. Now we can compute the conditional distribution of θ_{n+1} given $\theta_1, \cdots, \theta_n$ as

$$\theta_{n+1} \mid \theta_1, \cdots, \theta_n \sim \frac{1}{\alpha+n}\left(\alpha G_0 + \sum_{i=1}^{n}\delta_{\theta_i}\right) \tag{3}$$

DP works well until it comes to share mixture components among groups. In such cases, no components would be shared since the base measure G_0 is continuous. The solution is to extend DP to hierarchical DP (HDP), that is, let the base measure G_0 itself be distributed according to a DP. HDP can be formulated as follows:

$$\begin{aligned}G_0 &\sim \text{DP}(\gamma, H)\\ G_j &\sim \text{DP}(\alpha, G_0)\end{aligned} \tag{4}$$

HDP has similar properties to DP, since our model is based on HDP, more details will be seen in quickly.

IV. Our Model for Topic Evolution

A. Model

We assume that the documents arrive in ascending order of their publication date, and process them in slice. At each time slice we model the documents via a HDP mixture model and establish connections between successive topics. The graphical model is shown in Fig. 1. Here for clarity, we omit the superscripts t of the variables for current time slice if no ambiguity, while superscripts for previous time will be kept. For example, when we write G_0, we mean $G_0^{(t)}$; if $G_0^{(t-1)}$ intended, it will be expressed verbatim. One exception is H, which always means the base measure at time slice 1, and any later base measure would be denoted with a t superscript explicitly.

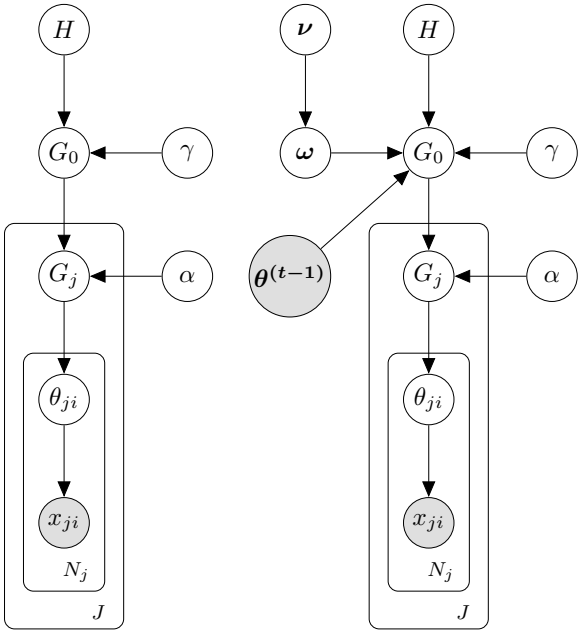

Fig. 1. (a) The left is the model at time slice 1; (2) The right is the model for any later time slice.

At time slice 1, J documents arrive, and document j consists of N_j words, represented as $\{x_{ji}\}_{i=1}^{N_j}$. The HDP mixture model used here is:

$$\begin{aligned}G_0 &\sim \text{DP}(\gamma, H)\\ G_j &\sim \text{DP}(\alpha, G_0)\\ \theta_{ji} &\sim G_j\\ x_{ji} &\sim F(\theta_{ji})\end{aligned} \tag{5}$$

where G_j is the DP for document j, θ_{ji} is the component (topic) parameter for word x_{ji}, and $F(\theta_{ji})$ is the distribution of a topic parametrized by θ_{ji}. Using the Chinese restaurant franchise metaphor, we can understand the HDP mixture model better. In the metaphor, there are J restaurants sharing the same menu. At restaurant j, when customer i comes in, he sits at a table t_{ji} and has the dish k_{ji} with probability proportional to $n_{jt\cdot}$, where n_{jtk} is the number of customers in restaurant j sitting at table t eating dish k, and $n_{jt\cdot}$ is the marginal count of n_{jtk}, representing the number of all the customers in restaurant j sitting at table t. Or he chooses a new table to sit down with probability proportional to α and chooses a dish to be served with probability proportional to $\sum_{i=1}^{K}\frac{m_{\cdot k}}{m_{\cdot\cdot}+\gamma}\delta_{\phi_k} + \frac{\gamma}{m_{\cdot\cdot}+\gamma}H$, where K is the number of unique dishes being served in all restaurants, ϕ_k denotes dish k, m_{jk} is the number of tables in restaurant j serving dish k, $m_{\cdot k}$ and $m_{\cdot\cdot}$ are its marginal count.

At any subsequent time slice t, we use the previous model to construct the base measure. We let the base measure $H^{(t)}$ itself be the weighted mixture of distributions centering around

previous topics and the uninformative prior used in time slice 1. We present this mixture model as follows:

$$\begin{aligned} \boldsymbol{\omega} &\sim \text{Dirichlet}(\boldsymbol{\nu}) \\ H^{(t)} &= \sum_{k=1}^{K^{(t-1)}} \omega_k \text{Dirichlet}(\phi_k^{t-1}) + \omega_{K^{(t-1)}+1} H \\ G_0 &\sim \text{DP}(\gamma, H^{(t)}) \\ G_j &\sim \text{DP}(\alpha, G_0) \\ \theta_{ji} &\sim G_j \\ x_{ji} &\sim F(\theta_{ji}) \end{aligned} \tag{6}$$

There is careful consideration why we select this scheme to establish time dependencies. First, it will enable slight changes in topics if we let the continuous base measure itself be a mixture but not the G_0 which is a draw from DP. The weights of these Dirichlet priors are controlled by $\boldsymbol{\omega}$. Second, the reason we incorporate H into $H^{(t)}$ is that we don't know about the documents, and retaining the uninformative prior keeps the model robust in case the contents change dramatically. Last, a mixture of conjugate priors is the same mixture of posteriors. This gives us the flexibility we want, at the same time keeps the posterior easily computed.

B. Posterior Inference

Posterior inference will be easily carried out with Gibbs sampling [14] if we use the Chinese restaurant franchise metaphor [5]. In the metaphor, latent variables $\boldsymbol{t}$, $\boldsymbol{k}$, $\boldsymbol{\phi}$, $\boldsymbol{\omega}$ appear, and their posteriors need to be estimated. For simplicity, we assume fixed values for all the hyperparameters γ, α and $\boldsymbol{\nu}$. Here we don't integrate out $\boldsymbol{\phi}$, since this will make the computation difficult though not intractable.

First of all, Let's introduce some useful notations and quantities. We let $F(\theta)$ have density $f(\cdot \mid \theta)$ and H have density $h(\cdot \mid \eta)$, where η is the parameter of H. Let z_{ji} denote the indicator of the mixture component associated the observation x_{ji}, which is simply a shorthand of $k_{jt_{ji}}$ and doesn't appear in the inference. let $\boldsymbol{x}_k = \{x_{ji} : \text{all } j, i \text{ with } z_{ji} = k\}$, $\boldsymbol{x}_{jt} = \{x_{ji} : \text{all } i \text{ with } t_{ji} = t\}$. And variables like $\boldsymbol{x}^{-ji}$,$\boldsymbol{k}^{-jt}$,$\boldsymbol{\phi}^{-k}$ denote the remainders after removing x_{ji},k_{jt},ϕ_k from corresponding sets of variables. We denote the conditional density of x_{ji} under existing topic (mixture component) k given all words except x_{ji} as

$$f_k^{-x_{ji}}(x_{ji}) = f(x_{ji} \mid \phi_k) \tag{7}$$

Similarly $f_k^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt})$ denotes the conditional density of $\boldsymbol{x}^{jt}$ under mixture component k.

Sampling $\boldsymbol{t}$. The full conditional of t_{ji} is obtained by combining the conditional prior for t_{ji} with the likelihood of generating x_{ji}. We compute the conditional prior for t_{ji} by exploiting the exchangeability and treating t_{ji} as the last variable sampled and have

$$p(t_{ji} \mid \boldsymbol{t}^{-ji}) = \sum_{t=1}^{m_{j\cdot}} \frac{n_{jt\cdot}^{-ji}}{n_{j\cdot\cdot} - 1 + \alpha} \delta_t + \frac{\alpha}{n_{j\cdot\cdot} - 1 + \alpha} \delta_{t^{new}} \tag{8}$$

We see that the probability of t_{ji} taking a previously used t is proportional to $n_{jt\cdot}^{-ji}$, and t_{ji} will take a new value t^{new} with probability proportional to α. Next we consider the likelihood due to x_{ji} given t_{ji}. If t_{ji} takes a value previously used, the likelihood is $f(x_{ji} \mid \phi_{k_{jt}})$; if a new table allocated, then the likelihood for $t_{ji} = t^{new}$ is calculated by integrating out the possible values of $k_{jt^{new}}$:

$$\begin{aligned} &p(x_{ji} \mid \boldsymbol{x}^{-ji}, \boldsymbol{t}^{-ji}, t^{ji} = t^{new}, \boldsymbol{k}, \boldsymbol{\phi}) \\ =&\sum_{k=1}^{K} \frac{m_{\cdot k}}{m_{\cdot\cdot} + \gamma} f_k^{-x_{ji}}(x_{ji}) + \frac{\gamma}{m_{\cdot\cdot} + \gamma} f_{k^{new}}^{-x_{ji}}(x_{ji}) \end{aligned} \tag{9}$$

where $f_{k^{new}}^{-x_{ji}}(x_{ji}) = \int f(x_{ji} \mid \phi) h(\phi) d\phi$. Thus we have the full conditional of t_{ji} as

$$\begin{aligned} &p(t^{ji} \mid \boldsymbol{x}, \boldsymbol{t}^{-ji}, \boldsymbol{k}, \boldsymbol{\phi}, \boldsymbol{\omega}) \\ \propto &\begin{cases} n_{jt}^{-ji} f_{k_{jt}}^{-x_{ji}}(x_{ji}) \text{ if } t \text{ is previously used,} \\ \alpha p(x_{ji} \mid \boldsymbol{x}^{-ji}, \boldsymbol{t}^{-ji}, t^{ji} = t^{new}, \boldsymbol{k}, \boldsymbol{\phi}) \text{ if } t = t^{new}. \end{cases} \end{aligned} \tag{10}$$

If the sampled value of t_{ji} is t^{new}, we would sample $k_{jt^{new}}$ by:

$$\begin{aligned} &p(k_{jt^{new}} = k \mid \boldsymbol{x}, \boldsymbol{t}, \boldsymbol{k}^{-jt^{new}}, \boldsymbol{\phi}, \boldsymbol{\omega}) \\ \propto &\begin{cases} m_{\cdot k} f_k^{-x_{ji}}(x_{ji}) & \text{if } k \text{ is previously used,} \\ \gamma f_{k^{new}}^{-x_{ji}}(x_{ji}) & \text{if } k = k^{new}. \end{cases} \end{aligned} \tag{11}$$

If k^{new} is sampled, then we need to allocate a new ϕ for this indicator by

$$\begin{aligned} &p(\phi_{k^{new}} \mid \boldsymbol{x}, \boldsymbol{t}, \boldsymbol{k}, \boldsymbol{\phi}^{-k^{new}}, \boldsymbol{\omega}) \\ =&p(\phi_{k^{new}} \mid H^{(t)}) p(x_{ji} \mid \phi_{k^{new}}) \\ =&\sum_{k=1}^{K^{(t-1)}} \omega_k \text{Dirichlet}(\phi_k + c) + \omega_{K^{(t-1)}+1} H(\eta + c) \end{aligned} \tag{12}$$

where c is such a vector that c_i equals 1 if x_{ji} is the ith word in vocabulary, otherwise 0.

Sampling $\boldsymbol{k}$. In fact we have already given how to sample k_{jt} in sampling t when a new table is allocated, the difference is that here all observations in table t associated with component k would be involved, so $f_k^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt})$ not $f_k^{-x_{ji}}(x_{ji})$ is to take the responsibility. The full conditional of k_{jt} is

$$\begin{aligned} &p(k_{jt} \mid \boldsymbol{x}, \boldsymbol{t}, \boldsymbol{k}^{-jt}, \boldsymbol{\phi}, \boldsymbol{\omega}) \\ \propto &\begin{cases} m_{\cdot k}^{-jt} f_k^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt}) & \text{if } k \text{ is previously used,} \\ \gamma f_{k^{new}}^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt}) & \text{if } k = k^{new}. \end{cases} \end{aligned} \tag{13}$$

As before, if the sampled value is k^{new}, then we obtain a sample of $\phi_{k^{new}}$ by sampling from:

$$\begin{aligned} &p(\phi_{k^{new}} \mid \boldsymbol{x}, \boldsymbol{t}, \boldsymbol{k}, \boldsymbol{\phi}^{-k^{new}}, \boldsymbol{\omega}) \\ =&\sum_{k=1}^{K^{(t-1)}} \omega_k \text{Dirichlet}(\phi_k + c') + \omega_{K^{(t-1)}+1} H(\eta + c') \end{aligned} \tag{14}$$

where c' is such a vector that c'_i equals the number of the ith word in vocabulary in $\boldsymbol{x}_{jt}$.

Sampling ϕ. Procedures of sampling ϕ_k have also been given, and the difference is that here all the observations are involved, not just customers at a table or in a restaurant. The full conditional of ϕ_k is

$$p(\phi_k \mid \boldsymbol{x}, \boldsymbol{t}, \boldsymbol{k}, \boldsymbol{\phi}^{-k}, \boldsymbol{\omega}) = \sum_{k=1}^{K^{(t-1)}} \omega_k \text{Dirichlet}(\phi_k + c'') + \omega_{K^{(t-1)}+1} H(\eta + c'') \tag{15}$$

where c'' is such a vector that c''_i equals the number of the ith word in vocabulary in $\boldsymbol{x}$.

Sampling $\boldsymbol{\omega}$. When we sample $\boldsymbol{\phi}$, we record the mixture of the base measure $H^{(t)}$ which ϕ_k comes from as b_k, and with this information available, we can calculate the full conditional of $\boldsymbol{\omega}$ as

$$p(\boldsymbol{\omega} \mid \boldsymbol{x}, \boldsymbol{t}, \boldsymbol{k}, \boldsymbol{\phi}) \sim \text{Dirichlet}(\boldsymbol{\nu}') \tag{16}$$

where $\boldsymbol{\nu}'$ is such a vector that $\nu'_i = \nu_i + \sum_{k=1}^{K} n_{..k}\delta_i(b_k)$

V. Experimental Results

We evaluate our model on a real world dataset Reuters-21578. To illustrate that incorporating time information benefits discovering topics and our model is better at modeling topic evolution, we compare our model with other two models, that is, OLDA and plain HDP without time dependencies considered. Here we notate our proposed model as HDPt (t for time). In OLDA, we will evaluate its performance under different number of topics, specifically, we let the number be $K = 5, 10, 50, 100$. The other parameters for LDA is set following [2], and we let $\alpha = 50/K$, $\beta = 0.1$. The settings for plain HDP and the first model in HDPt are the same, and we let $\alpha = 1.0$, $\gamma = 1.0$, $\beta = 0.1$. Two application domains chosen here are document modeling and document classification, and we'll evaluate these models on perplexity and normalized mutual information. First of all, let's give an introduction to the data and our preprocessing on them.

A. Dataset

Not all the data of Reuters-21578 are used in this experiment. First, only news from March to April in 1987 is considered to conduct our experiments. Second, in these news, only that with exactly one topic is preserved for simplicity (but note that our model is applicable to news with multiple topics without adaptation). Third, only topics appearing in more than four documents are preserved, and the documents with obsolete topics are removed, since we insist that "topics" reflected in just several documents don't mean much. Fourth, on the words, we make some general cleaning, including lowercasing, removing stopwords, cutting low frequency words and stemming. After all these preprocessing, we obtain the final data consisting of 7238 documents, 50 topics, a vocabulary with 6937 terms, and 424986 words in total. Then, we divide the data by week, so 8 slices are obtained. In table I, we give a summary on these slices. One thing to notice is that the topics of the news here are known as a result of our procedures described before, and the number of topics in table 1 is the one that annotated by humans not the one models finding.

TABLE I
A SUMMARY OF THE DATA

Week	The Number of Documents	The Number of Topics
1	1173	40
2	990	43
3	1442	44
4	1139	40
5	1209	45
6	743	41
7	418	32
8	124	28

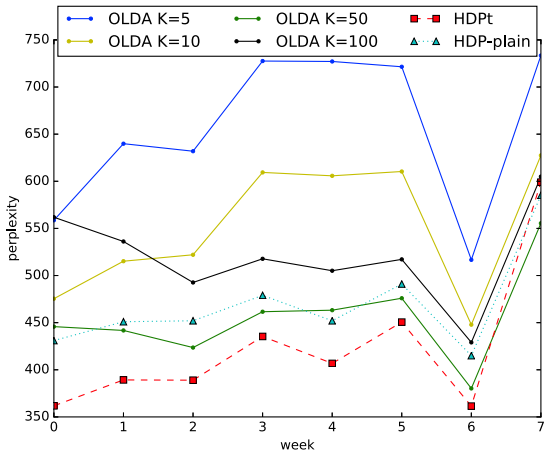

Fig. 2. Perplexities over time

B. Perplexity

Perplexity is a criterion used to measure clustering quality originally used in language modeling, which is often evaluated on held-out data to assessing the model's generalization ability [15]. It's defined as follows

$$Perplexity(\boldsymbol{x}') = \exp\left\{-\frac{\sum_{j=1}^{J'} \log p(\boldsymbol{x}'_j)}{\sum_{j=1}^{J'} N'_j}\right\} \tag{17}$$

where $\boldsymbol{x}'$ are the documents to be tested, $\boldsymbol{x}'_j$ is one of the documents, and N'_j is the length of $\boldsymbol{x}'_j$. Higher perplexity means more misprediction, so lower perplexity is desired.

In this paper, we use perplexity in a different way. Since our purpose is to inspect the influence of previous model on the current one, we don't compute the perplexity on unseen documents as most do, but compute on the data that we used to fit the model. The results are shown in Fig. 2, from the figure we can see that when $K = 50$, OLDA performs better than other number of topics, and plain HDP is comparable with OLDA. Our model HDPt plays best most of the time, except the last time slice, perhaps because there are too few data.

C. Normalized Mutual Information

Perplexity itself is not enough to measure the quality of a model. For example, high perplexity may be achieved

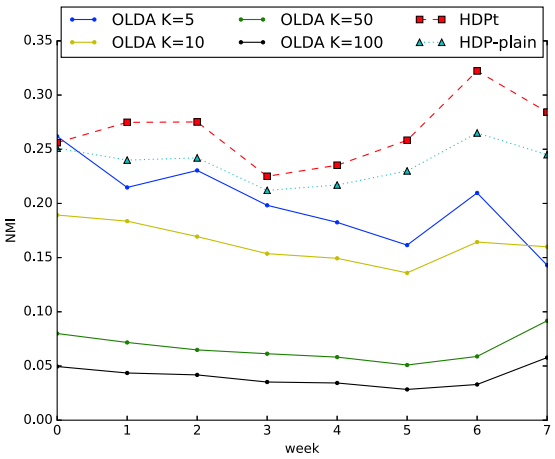

Fig. 3. NMI over time

meanwhile meaningless topics are discovered, which is not what we want. So another criterion used here is to inspect whether it can cluster relevant documents together structurally. Thanks to the great work of Lewis and many others, we are armed with the real topics of the documents, and we can compare the clustering with the real one.

For document d_j, assuming its true distribution over class c is $p(c = l \mid d_j), 1 \leq l \leq L$ and estimated distribution over class z is $p(z = k \mid d_j), 1 \leq k \leq K$. If c and z are independent, which means that we can obtain nothing about one variable from the other, then clearly the clustering result is extremely bad. So our goal is to enlarge the divergence between $p(c, z)$ and the completely random one $p(c)p(z)$, which can be computed through Kullback-Leibler divergence. The quantity is called MI (Mutual Information) expressed as

$$\begin{aligned} &MI(c,z) \\ =&D_{\mathrm{KL}}(p(c,z) \parallel p(c)p(z)) \\ =&\sum_{l=1}^{L}\sum_{k=1}^{K} p(c=l,z=k)\log\frac{p(c=l,z=k)}{p(c=l)p(z=k)} \end{aligned} \tag{18}$$

If we normalize MI, then we obtain NMI

$$NMI(c,z) = \frac{MI(c,z)}{\sqrt{H(c)H(z)}} \tag{19}$$

where $H(x)$ denotes the entropy of variable x. We see that the value of NMI ranges in $[0, 1]$, and higher NMI means better clustering result.

We show our experiment result on NMI in Fig. 3. From the figure, we see that our proposed model HDPt obtains the highest NMI, while all OLDA perform poorly, perhaps that's because the number of topics fixed for OLDA doesn't fit the actual data.

VI. Conclusions

In this paper we introduce the problem of topic evolution, present relevant methods and propose our HDP-based model. Our model works in an online fashion and we develop a solution based on Gibbs sampling to carry out posterior inference. Our approach can automatically find topics at each time slice and we are not required to specify the number of topics.

We conduct an experiment on dataset Reuters-21578, and compare the results with OLDA and plain HDP on two criteria, perplexity and normalized mutual information. The results verify that our model can capture the dependencies between topics better. Here we don't study the evolution of topics on a concrete case, however, with the ability to model topic dependencies, our model can easily accomplish this by evaluating the strength of relationships through some measure like KL-divergence and inspecting the evolutionary pattern qualitatively.

Acknowledgment

This work is supported by National High Technology Research and Development Program of China (No.2013CB329605) (973 Program), and National Natural Science of Foundation of China (No.61370136).

References

[1] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.

[2] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[4] Y. W. Teh, "Dirichlet process," in *Encyclopedia of machine learning*. Springer, 2011, pp. 280–287.

[5] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the american statistical association*, 2012.

[6] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study: Final report," in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, USA, Feb. 1998, pp. 194–218, 007.

[7] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 113–120.

[8] X. Wang and A. McCallum, "Topics over time: a non-markov continuous-time model of topical trends," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 424–433.

[9] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 3–12.

[10] A. Gohr, A. Hinneburg, R. Schult, and M. Spiliopoulou, "Topic evolution in a stream of documents." in *SDM*, vol. 9. SIAM, 2009, pp. 859–872.

[11] L. Ren, D. B. Dunson, and L. Carin, "The dynamic hierarchical dirichlet process," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 824–831.

[12] J. Zhang, Y. Song, C. Zhang, and S. Liu, "Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 1079–1088.

[13] J. Sethuraman, "A constructive definition of dirichlet priors," *Statistica sinica*, pp. 639–650, 1994.

[14] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," *Journal of computational and graphical statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[15] G. Heinrich, "Parameter estimation for text analysis," *University of Leipzig, Tech. Rep*, 2008.