A Scalable Model for Tracking Topical Evolution in Large Document Collections

Sheikh Motahar Naim$^1$, Arnold P. Boedihardjo$^2$, M. Shahriar Hossain$^1$

$^1$Department of Computer Science, University of Texas at El Paso, El Paso, TX 79912, USA
$^2$U. S. Army Corps of Engineers, Alexandria, VA 22315, USA
snaim@miners.utep.edu, arnold.p.boedihardjo@usace.army.mil, mhossain@utep.edu

Abstract—In this era of big data, many domains on the web naturally have massive amount of labeled text data that are growing over time, for example, digital publication archives, social media posts, and question-answer forums. Probabilistic graphical models have shown great potential for mining such text corpora in recent years. Some of these algorithms utilize explicit annotations and labels associated with documents to guide the probabilistic model to find hidden themes. A few techniques attempt to utilize the timestamps associated with documents to model the evolution of those latent topics. However, no effort has been devoted to utilize these two different dimensions of information together — timestamps and labels or annotations — to discover evolution of labeled themes. In this paper, we present a new topical model called the Supervised Topical Evolution Model (STEM), which is a monolithic graphical model capable of using annotations, timestamps, and textual contents to discover interpretable and evolving themes from big text datasets. STEM simultaneously learns latent themes and their changes over time using a stochastic process that is driven by labels or annotations. In addition, we provide an asynchronously distributed inference process for STEM that results in significant speedup in learning time, making the model scalable for large datasets. Extensive experiments demonstrate that our proposed model is able to infer highly interpretable topics that reflect temporal patterns, in much less time than other comparable topic modeling methods.

Keywords—Scalability; Graphical Models; Temporal Topic Modeling; Probabilistic Inference

I. INTRODUCTION

Though the web is an ever increasing source of information of every type — text, imagery, audio, video, and graph, most of the explanatory information on the web are still in text format. Therefore the necessity of new models to enhance explanatory capabilities using large text collections is still increasing. This paper particularly focuses on the domains where a text collection grows over time and the documents are annotated through labels or hand-coded tags. Some examples are — online question-answer or discussion forums like Stack Overflow (stackoverflow.com) where questions (and some answers) are associated with relevant tags; social networks like Twitter (twitter.com) where posts are usually marked by multiple hashtags; digital libraries like IEEE Xplore (ieeexplore.ieee.org) and ACM digital library (dl.acm.org) where scientific publications from different conferences and journals are being added every year, and each paper is tagged with author-defined keywords and publisher-defined categories.

Understanding the evolution of a dynamic corpus is of great importance to the users of any domain. Consider the case of a researcher who wishes to explore a new research area. A high level summary of the important topics in that area extracted from related publication archives, along with the evolution of those topics over time would give her a quick and informative snapshot of that area. It would help her detect the emerging topics and narrow down the scope of the area that she is interested in. For example, Figure 1 shows the evolutionary patterns of 10 computer science and engineering related topics extracted by our proposed STEM model from the IEEE publication dataset. This figure reveals, for instance, the fall of “Filtering Theory” followed by the rise of “Artificial Intelligence” in the last decade, which rightly indicates the paradigm shift that took place in the area of prediction and estimation. Another useful application of such models can be in the intelligence analysis domain. Instead of going over hundreds of articles from multiple sources, analysts can use STEM model to conveniently analyze recent topics and their activity levels. Those information would help them achieve situational awareness to aid decision making.

An effective way to summarize large corpus is to find hidden topics that are embedded in the corpus. Probabilistic graphical models like Latent Dirichlet Allocation (LDA) [1] have been shown to be effective in inferring such latent topics. LDA is a generative model that assumes that the corpus
has a fixed number of latent topics and each document is a mixture of all those topics, while each topic in turn is a probability distribution over all the words. This model results in very concise representations of the documents which are shown to be very useful in information retrieval, document summarization, and classification tasks. A number of variations [2], [3], [4] were proposed subsequently to improve different aspects of LDA.

However, all these models operate in an unsupervised manner and the topics are learned mostly based on co-occurrences of words. In case of very large datasets, which usually contain hundreds of topics, it becomes very difficult to interpret the topics and separate them from one another. To make the topics more interpretable, several supervised techniques have been proposed in the literature. Supervised LDA [5] and DiscLDA [6] generate topics by associating one label with each document but these methods are inapplicable in many domains since documents may have a mixture of multiple labels. Labeled LDA [7] and MM-LDA [8] techniques address this issue by considering each document as a bag of words with multiple labels.

Though Labeled LDA and MM-LDA produce interpretable topics very efficiently, both of them consider the corpus as a static collection. However, text collections grow over time in many domains on the web. Different topics may have varying degree of influence in the overall topical structure at different time periods. By ignoring the dynamics of the corpus, static models miss the opportunity for the topics to be guided by the documents’ temporal information.

A few models like Dynamic Topic Models (DTM) [9] and Topics Over Time (TOT) [10] leverage timestamps of the documents to capture the evolution of the topics. However, being unsupervised, these models suffer from lack of flexibility in incorporating domain knowledge and from difficulty in interpretation of the resultant topics.

In this paper we present a novel graphical model that incorporates both annotations and timestamps of each document in the topic inference process. The proposed model, Supervised Topical Evolution Model (STEM), is a generative process where the topics in a document are constrained by its set of labels and each topic is considered as a Beta distribution over time. Unlike some temporal topic modeling methods that divide the documents into fixed time slots, we consider time as a continuous phenomena and model it as such. This enables us to predict future trend of a topic, estimate timestamp of a document from its content, and also to avoid the issues related to choosing a proper discretization window size. We derive a sampling algorithm using Collapsed Gibbs Sampler to simultaneously infer the topic distributions of the documents, term distributions of the topics, and topical changes over time. Table I shows a comparison of capabilities of the proposed model with the state-of-the-art models.

In summary, the contributions of this paper are:

1) We propose a probabilistic graphical model, STEM, that incorporates both temporal information and annotations along with the textual content of documents.
2) We derive an efficient sampling algorithm to infer the latent variables of the model.
3) We show a multiprocessor and shared memory approach to significantly accelerate the inference process for large scale datasets.
4) We demonstrate that STEM is able to effectively learn highly interpretable topics and their evolution.
5) Through extensive experiments on synthetic and real-world datasets we show that STEM can model the labels and topical patterns very efficiently.

### Table I

<table>
<thead>
<tr>
<th>Model</th>
<th>Incorporates labels</th>
<th>Dynamic time</th>
<th>Continuous timestamps</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Labeled LDA</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>MM-LDA</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>DTM</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TOT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>STEM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

## II. RELATED WORK

Finding hidden themes in a document collection has been of great interest to data mining and information retrieval researchers for more than two decades. An early work in the literature is latent semantic indexing (LSI) [11] that maps document and terms in a special “latent semantic” space by applying dimensionality reduction on traditional bag-of-words vector space representations of documents. A probabilistic version of LSI, pLSI [12], introduces a mixture model where each document is represented by a mixing proportion of hidden topics. Latent Dirichlet Allocation (LDA) [1], a somewhat generalized but more sophisticated version of pLSI, is one of the most notable models in the literature. It provides a generative probabilistic approach for document modeling assuming a random process by which the documents are created. LDA spawned a deluge of work exploring different aspects of topic modeling. For example, Online LDA (OLDA) [2] handles streams of documents with dynamic vocabulary, Griffiths et al. [4] exploit the sentence structures of documents and Correlated Topic Model (CTM) [3] captures the correlation between topics.

One important limitation of these models is that they infer the hidden topics in an unsupervised manner, mostly based on word co-occurrences. This makes the topics hard to interpret for the end users. A number of supervised learning models have been proposed to address this interpretability issue. Supervised LDA [5] adds a response variable associated...
with each document, and jointly models the documents and the responses in order to find latent topics that will best predict the response variables for future unlabeled documents. DiscLDA [6], which is a discriminative variation of LDA, associates a class label with each document and learns a topic mixture for each label. Unlike these two models, MM-LDA [8] is not limited to one label per document. It assumes multiple tags for each document and learns the topics by observing both the words and tags simultaneously. Labeled LDA [7] is also a multi-label supervised model which brings even more interpretability to the topics by defining an one-to-one correspondence between the labels and the latent topics.

Maximum entropy discrimination LDA (MedLDA) [13] model integrates the mechanism behind the max-margin prediction models like SVM with the mechanism of LDA under a unified constrained optimization framework. The latent topics generated by MedLDA are more discriminative and suitable for prediction tasks such as document classification or regression. There are also a few semi-supervised topic models that handle the situations where labels for many of the documents in the corpus are not available. SHSLDA [14] is a semi-supervised hierarchical topic model that automatically explores new topics from the corpus while incorporating information from observed hierarchical labels into the modeling process. Partially Labeled LDA (PLDA) [15] introduces models that make use of the unsupervised learning style of topic models to discover the hidden topics within each label, as well as unlabeled, corpus-wide latent topics.

Although these supervised models leverage explicit annotations of the documents to improve the quality of the topics, they all consider the document collections as static. A number of models such as Dynamic Topic Model (DTM) [9], Topics Over Time (TOT) [10] and Evolutionary Theme Pattern [16] were proposed to address the dynamics of the corpus. DTM divides the corpus into a fixed time slots and applies topic modeling on the subset of documents in each time slot while learning the topical transition in consecutive time slots by a Gaussian noise. The evolutionary theme pattern model also divides documents into time slots but tries to capture evolution by computing topic similarity in close time intervals. TOT model alleviates the problems related to discretization of time by learning a Beta distribution over continuous time space for every topic. Our proposed model STEM also treats time as a continuous phenomena, but by adopting a supervised approach it is able to produce more crisp and interpretable topics while capturing the topical evolutions more effectively.

There are existing efforts on improving the runtime of classic LDA to make it feasible for large web-scale datasets. Yan et al. proposed a parallel inference method [17] for LDA using Graphical Processing Units (GPU). Newman et al. [18] introduced another parallel version of LDA that can provide substantial memory and time savings. The method proposed by Newman et al. is fully synchronous requiring a global synchronization at each iteration. Asuncion [19] proposed an asynchronous sampling approach where the data pieces are distributed across P processors. The processors independently perform Gibbs sampling on their local data and communicate in a local asynchronous manner with other processors. PLDA [20] is a distributed, MapReduce based solution that resolves storage and computation bottlenecks of LDA and provides fault recovery for lengthy distributed computations. Smola and Narayanamurthy [21] proposed a similar distributed method but achieved a much better throughput by avoiding separate computation and synchronization phases. Liu et al. [22] provides another improvement over PLDA by suggesting a number of new strategies like data placement, word bundling and priority-based scheduling. Our STEM model is unique in this comparison due to the fact that we infer not only the topics but also their evolution, and use an in-memory, asynchronous approach to parallelizing the inference.

III. METHODOLOGY

The proposed Supervised Topical Evolution Model (STEM) is a probabilistic graphical approach for modeling dynamic document collections with explicit labels. Similar to LDA, STEM views each document as a mixture of underlying topics, and each topic as a distribution over the words. However, STEM learns the topics and their evolution over time by guiding its inference procedure through the incorporation of timestamp and label information of the documents. We first derive the sequential STEM model in the three following subsections. Then in the fourth subsection, we describe how we parallelize the operations.

A. The Generative Process

Let the corpus \( \mathcal{D} = \{d_1, d_2, \ldots, d_D\} \) be a set of \( D \) documents, where each document \( d \in \mathcal{D} \) consists of a list of words \( W_d = [w_{1d}, w_{2d}, \ldots, w_{Nd}] \), a timestamp \( t_d \) and a list of binary variables \( \Lambda_d = [l_1, l_2, \ldots, l_K] \) to represent the labels of the document. Each word \( w_i \) comes from a vocabulary of size \( V \) and each \( l_k \in \{0, 1\} \) is an indicator of the presence or absence of the \( k \)-th label in this document. \( N_d \) is the length of document \( d \), and \( K \) the total number of unique labels in the corpus. Number of topics in STEM is the number of unique labels \( K \), making an one-to-one mapping between topics and labels. The generative process for the model is as follows:

1) For \( k = 1 \) to \( K \):
   a) \( \phi_k \sim \text{Dirichlet}(\beta) \)

2) For each document \( d \in \mathcal{D} \):
   a) For \( k = 1 \) to \( K \):
      i) \( \Lambda_{dk} \sim \text{Bernoulli}(\gamma_k) \)
   b) Compute \( L_d \) from \( \Lambda_d \)
c) \( \alpha_d = L_d \times \alpha \)
d) \( \theta_d \sim \text{Dirichlet}(\alpha_d) \)
e) For each word \( w_{di} \in d: 
   i) \ z_{di} \sim \text{Multinomial}(\theta_d) 
   ii) \ w_{di} \sim \text{Multinomial}(\phi_{z_{di}}) 
   iii) \ t_{di} \sim \text{Beta}(\psi_{z_{di}}) 
\)

List of all the symbols used in the generative process is provided in Table II.

### B. Modeling Labels and Timestamps

In step 1 of the generative process above, a multinomial topic distributions over vocabulary \( \phi_k \) is drawn for each topic indexed by \( k \) from a Dirichlet prior \( \beta \). In the conventional LDA model, a multinomial mixture distribution \( \theta_d \) is then drawn over all \( K \) topics from a Dirichlet prior \( \alpha \) for each document \( d \). However, STEM restricts \( \theta_d \) to be distributed over only the topics that correspond to its labels \( \Lambda_d \). With this restriction we ensure that all the topic assignments \( z_{di} \) to the words in document \( d \) are limited to the document’s labels, since \( z_{di} \) are drawn only from \( \theta_d \).

In step 2, at first a Bernoulli coin toss is used to generate the document’s labels \( \Lambda_d \) for each topic \( k \), with a labeling prior probability \( \gamma_k \). Now, we define \( \lambda_d = \{ k | \Lambda_d k = 1 \} \), which is a vector of the document’s labels and helps us to compute a document-specific label projection matrix \( L_d \) for each document \( d \). Size of this matrix is \( M_d \times K \), where \( M_d = | \lambda_d | \). The projection matrix is defined as:

\[
L^j_d = \begin{cases} 
1 & \text{if } \lambda_{di} = j \\
0 & \text{otherwise.} 
\end{cases}
\]  

which means the \( i \)-th row would have an entry of 1 in column \( j \) if and only if the \( i \)-th document label \( \lambda_{di} \) is equal to the \( j \)-th topic, and zero otherwise. We use the \( L_d \) matrix to project the parameter vector of the Dirichlet topic prior \( \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_K] \) to a lower dimensional vector \( \alpha_d \) as \( \alpha_d = L_d \times \alpha \). Here, the dimensions of the projected vector correspond to the topics represented by the labels of the document, which fulfills our requirement that the document’s topics are restricted to its own labels. In the plate notation in Figure 2, the dependency of \( \theta \) on both \( \alpha \) and \( \Lambda \) is shown by directed edges from \( \Lambda \) and \( \alpha \) to \( \theta \).

In the proposed model, topic inference is influenced not only by word co-occurrences like LDA, or the documents’ labels as discussed above but also the temporal information in the documents. To account for the changes in topical influence over time, we model continuous timestamp values instead of modeling a sequence of state changes with a Markov assumption on the dynamics. Time is naturally continuous, and discretization of time always creates the problem of selecting the slice size. The slice size is certainly too small for some regions and too large for others. Some temporal topic models like DTM [9] adopted this fixed time slot based approach. However, we model each topic as a continuous distribution over time and thus avoid discretization. Here we choose Beta distribution, since it can take diverse shapes unlike most other standard distributions, e.g., Gaussian. For the Beta distribution, range of timestamps of the documents in the corpus is normalized to a range from 0 to 1. Although a timestamp is generated for each word in document \( d \) in step 2(e) of the generative process described in the previous subsection, all these timestamps are observed as the same as that document. Eventually that generative process gives us the following joint distribution:

\[
p(w, t, z, \theta, \phi | \alpha, \beta, \psi) = p(\phi | \beta) p(\theta | \alpha) p(z | \theta) p(w | z, \phi) p(t | z, \psi)
\]  

Here \( \alpha \) represent the distribution after the projection made by \( L_d \). For the sake of simplicity, we will refer to the projected vector \( \alpha_d \) by just \( \alpha \) in the rest of this paper.

### C. Posterior Approximation

Now, from the joint distribution in Equation 2, we are interested in inferring the unobserved variables \( z, \theta \) and \( \phi \). This can be done by reversing the defined generative process and learning the posterior distributions of the latent variables given the observed data, which translates to solving the following equation:

\[
p(z, \theta, \phi | w, t, \alpha, \beta, \psi) = \frac{p(z, \theta, \phi, w, t | \alpha, \beta, \psi)}{p(\alpha, \beta, \psi | w, t)}
\]  

However, this distribution is infeasible to compute; particularly, the normalization factor, \( p(\alpha, \beta, \psi | w, t) \), cannot be computed exactly. Fortunately, a number of approximate inference techniques including variational inference and Gibbs approximation.
Applying the rules of conditional probability:

$$\theta_{dz} = \frac{n_{dz} + \alpha}{\sum_{z=1}^{K} (n_{dz} + \alpha)}$$

$$\phi_{zv} = \frac{n_{zv} + \beta}{\sum_{v=1}^{V} (n_{zv} + \beta)}$$

where \(n_{dz}\) is the number of times words in document \(d\) are assigned to topic \(z\) and \(n_{zv}\) is the number of times the \(v\)-th word in the vocabulary is assigned to topic \(z\) in the whole corpus. This allows us to use a simpler algorithm called Collapsed Gibbs Sampler that integrates out the multinomial parameters and simply samples \(z_{di}\). The collapsed Gibbs sampler for STEM computes the probability of a topic \(z_{di}\) being assigned to a word \(w_{di}\), given the topic assignments to all other words in the corpus. More formally, STEM needs to compute the following posterior up to a constant:

$$p(z_{di}|w, t, z_{-di}, \alpha, \beta, \psi)$$

where \(z_{-di}\) means all topic allocations except for \(z_{di}\).

Applying the rules of conditional probability:

$$p(z_{di}|z_{-di}, w, t, \alpha, \beta, \psi) = \frac{p(z_{di}, z_{-di}, w, t|\alpha, \beta, \psi)}{p(z_{-di}, w, t|\alpha, \beta, \psi)} = \frac{\int \int p(z, w, t, \theta, \phi|\alpha, \beta, \psi) d\theta d\phi}{\int \int p(z, w, t|\alpha, \beta, \psi) d\theta d\phi}$$

Following the generative process defined by Equation 2, we can expand the above equation to get:

$$p(z, w, t|\alpha, \beta, \psi) = \frac{\int \int p(\phi|\beta)p(\theta|\alpha)p(z|\theta)p(w|z, \phi)p(t|z, \psi) d\theta d\phi}{\int \int p(\phi|\beta)p(\theta|\alpha) d\theta \int p(\phi|\beta)p(w|z, \phi) d\phi}$$ 

In the above equation, both integration terms are multinomials with Dirichlet priors. Since the Dirichlet distribution is conjugate to the multinomial distribution, we can simply multiply the two results in a Dirichlet distribution with an adjusted parameter. Applying this idea and the chain rule, we can derive the conditional probability as follows. Details of this derivation is omitted here due to the page limitation.

$$p(z_{di}|z_{-di}, w, t, \alpha, \beta, \psi) \propto \frac{n_{z_{di}w_{di}} + \beta - 1}{\sum_{v=1}^{V} (n_{z_{di}v} + \beta) - 1}$$

For the sake of speed and simplicity, we update \(\psi\) after each Gibbs sample using the method of moments as follows:

$$\hat{\psi}_{z1} = \bar{t}_{z} \left( \frac{\bar{t}_{z} (1 - \bar{t}_{z})}{s_{z}^{2}} - 1 \right)$$

$$\hat{\psi}_{z2} = (1 - \bar{t}_{z}) \left( \frac{\bar{t}_{z} (1 - \bar{t}_{z})}{s_{z}^{2}} - 1 \right)$$

where \(\bar{t}_{z}\) and \(s_{z}\) indicate the mean and standard deviation of all the timestamps belonging to topic \(z\), respectively.

Algorithm 1 describes the steps to compute the posterior probabilities of the hidden variables.

### D. Parallel Inference

Now we describe how the inference process of the STEM model is distributed over multiple processors to reduce overall convergence time. At first, the distributions \(\theta\), \(\phi\) and \(\psi\) are initialized by assigning uniformly random topic to each word. These distributions are kept global to the whole process, and are shared by all the processors. Then the documents are distributed to the processors evenly, where each processor receives \(\frac{1}{P}\) documents (\(P\) is the number of processors). Each processor goes over its share of documents, and for every word \(w_{di}\), it samples a topic \(z_{di}\) using Equation 9. Each processor asynchronously updates \(\theta\) and \(\phi\) after every sampling and updates \(\psi\) after a full iteration over its documents. The key point here is that all these updates are done asynchronously — we do not need to maintain a global queue or one processor do not have to
Algorithm 1: InferSTEM – algorithm to compute the posterior probabilities.

```
input : Document set $D$
        List of words of each document, $W_d$
        Timestamp of each document, $t_d$
        Set of labels of each document, $A_d$
parameter: Dirichlet prior $\alpha$ and $\beta$
output : Document-topic distribution, $\theta$
         Topic-word distribution, $\phi$
         Topic distribution over time, $\psi$
1 Randomly initialize topic assignment $z$ for all words
2 Compute the count variables $n_{d,z}$ and $n_{z,v}$
3 for $d = 1$ to $D$ do
4    Compute $L_d$ using Equation 1
5    $\alpha_d = L_d \times \alpha$
6 end
7 for $iter = 1$ to $N_{iter}$ do
8    for $d = 1$ to $D$ do
9        for $i = 1$ to $N_d$ do
10           $v = W_{di}$
11           $n_{d,z_i} += 1$
12           $n_{z_i,v} += 1$
13           draw new $z_{di}$ using Equation 9
14           $n_{d,z_i} += 1$
15           $n_{z_i,v} += 1$
16        end
17    end
18    for $z = 1$ to $K$ do
19        update $\psi_z$ using Equation 10
20-end
21 Compute $\theta$ and $\phi$ using Equation 4 and 5
22 return $\theta$, $\phi$ and $\psi$
```

wait for others to finish — giving a almost linear speedup in sampling rate with respect to number of processors. One prospective downside of such asynchronous sampling could be the concurrency issue — multiple processors might be reading and updating the same probability, causing some processors getting a slightly older value. However, such issues are mitigated quickly for big data since we usually do millions of samplings for reasonably large datasets.

IV. EXPERIMENTAL RESULTS

In this section we seek to answer the following questions.

1) Are the topics generated by STEM more interpretable than those of an unsupervised temporal topic model? (See Section IV-A)

2) Do the topics inferred by the STEM method capture the original temporal information? (See Section IV-B)

3) How do the topics evolve over time with respect to each other? (See Section IV-C)

4) Is the proposed model able to automatically annotate unlabeled documents effectively. (See Section IV-D)

5) How much speedup in convergence time is achieved due to the parallelization of the inference process? (See Section IV-E)

We implemented a sequential and a parallel version of the STEM model following the methodology described in Section III. The parallel version uses 12 processors, and is referred to as STEM-P in the rest of this section. We conducted a set of quantitative and qualitative experiments using these models to answer the preceding questions. We have prepared a synthetic dataset and two real world datasets — a publication archive from IEEE Xplore and a collection of posts from StackOverflow.com website – to facilitate the experiments. We compared our model with two state-of-the-art models – Topics Over Time (TOT) [10], which models topical evolutions over continuous time, and Labeled LDA (LLDA) [7], which uses labels to guide the model inference process.

**Synthetic Dataset:** The synthetic dataset has four separate groups of document, namely G1, G2, G3 and G4, each group containing 100 documents. Each of these 400 documents takes terms from seven mutually exclusive sets of terms, each set having 100 unique terms. The way terms from different sets are shared by the documents of various groups are shown in Figure 3. For example, each document in document set G1 is a mixture of terms from term set 1, 5 and 7. Timestamps span 20 years and are distributed in a linear uniform manner, i.e., first 20 documents of G1 belong to year 1, the next 20 belong to year 2, and so on.

**IEEE Publication Dataset:** This academic dataset contains 412,184 computer science and engineering related papers published between 1989 and 2013. Links to all these IEEE publications are found in the DBLP archive.¹ We collected the title and the abstract of each of these publications using the IEEE Xplore search gateway.² We then extracted the entities from these documents by removing the stop words and applying stemming, and represented each document as a bag of entities. We ignored the documents with less than 10 entities since they do not contain enough discriminative information. There are a number of labels associated with each publication given by the publisher. We have selected the most frequent 50 labels, and also ignored the documents with less than three labels.

¹http://dblp.uni-trier.de/
²http://ieeexplore.ieee.org/gateway/
probable terms of each topics should give a clear idea of reproducibility, we kept all our data sets and codes in a much larger compared to the IEEE dataset. For the purpose of comparison, we selected only the questions which have accepted answers, here is a question or an answer. From this collection, we ranging from July 2008 to September 2016. Each post is online forum for computer programming related problems. However, most of the topics of TOT method are carrying multiple themes. Almost all the topics generated by the STEM methods are easily interpretable and very distinctive. Similar topics are found by the LLDA model as well.

Stackoverflow Post Dataset: Stackoverflow\(^3\) is a popular online forum for computer programming related problems. We have collected a dump of all of its 32 million posts ranging from July 2008 to September 2016. Each post here is a question or an answer. From this collection, we selected only the questions which have accepted answers, and concatenated a question and its accepted answer to make a single post. We then randomly selected 20% of the posts from each month, which gives a total of 1.5 million posts. After that we processed those post following the similar steps we performed for the IEEE dataset. The only difference is, we took top 100 labels here since this dataset is much larger compared to the IEEE dataset. For the purpose of reproducibility, we kept all our data sets and codes in a public domain\(^4\).

### A. Quality of Topics

The primary concern of any topic modeling algorithm is to produce high quality topics where, ideally, the most probable terms of each topics should give a clear idea about the theme of the topic. Also each topic should be carrying only one theme and be distinguishable from others. Figure 4(top) shows the top 10 terms, in descending order of their probabilities, of some of the topics inferred by the proposed methods from IEEE publication dataset. Top terms of STEM and STEM-P are shown together, since they are proposed methods from IEEE publication dataset. Top terms of their probabilities, of some of the topics inferred by the STEM and STEM-P, (middle row) LLDA and (bottom row) TOT. Almost all the topics generated by the STEM methods are easily interpretable and very distinctive. Similar topics are found by the LLDA model as well. However, most of the topics of TOT method are carrying multiple themes.

![Figure 4. Ten topics in IEEE publication dataset inferred by four models – (top row) STEM and STEM-P, (middle row) LLDA and (bottom row) TOT. Almost all the topics generated by the STEM methods are easily interpretable and very distinctive. Similar topics are found by the LLDA model as well. However, most of the topics of TOT method are carrying multiple themes.](image)

![Figure 5. Average KL divergence among the topics learned by STEM, STEM-P, LLDA and TOT methods from three datasets. STEM methods show higher KL divergence meaning more distinct topics.](image)

\(^3\)http://stackoverflow.com/

\(^4\)https://www.dropbox.com/sh/ytymir7lu06g089/AAD26Z2mNizaWuAD2KdyBjCa?dl=0

The corresponding topics discovered by LLDA in Figure 4(middle) are also more or less similarly interpretable. However, the topics produced by the TOT method shown in Figure 4(bottom) are not easily interpretable and most of them involve multiple themes. For example, the list of top terms in Topic 43 contains wireless LAN related terms but also has some unrelated terms. Topic 17 contains similar mixtures with network routing.

The topics inferred from the Stackoverflow posts by those three methods also show similar characteristics. For example, STEM finds the terms ‘lock’, ‘queue’, ‘worker’, ‘wait’ and ‘mutex’ as the most probable terms for the topic ‘multithreading’, while the top terms of the topic ‘security’ were found to be ‘attack’, ‘hash’, ‘authentication’, ‘encryption’ and ‘cookie’. All these top terms of each topic carry a single theme which coincides with the topic’s label. We cannot show more topics from the Stackoverflow dataset due to space limitation.

We also computed all-pair average Kullback-Leibler (KL) divergence among the topics to quantitatively measure the diversity in the topics learned by STEM, STEM-P, TOT and LLDA models. In Figure 5 we see that the KL divergence in case of the STEM topics is larger than that of the TOT topics, and also the LLDA topics, in all three datasets. That means STEM models are able to infer topics which are more distinguishable and have less overlap of multiple themes. This is the desired property of STEM model, since we are not seeking hierarchical topics in this paper.

B. Timestamp Estimation

While generating the topics underlying the document collection, STEM makes use of the temporal information of each document. To verify if the topics inferred by the proposed method can capture time, we attempt to estimate the timestamp of each document from its topic distribution. We use Support Vector Machine regression to predict the timestamp considering the topic distribution in documents as the feature vectors. We compared our proposed model with the Topics Over Time (TOT) model and a baseline model that randomly predicts a timestamp between 0 and 1 for each document. We measured Root Mean Squared Error (RMSE) of each model using cross-validation for comparison. Figure 6 shows the performance of different models at various training/test split of the dataset (marked by the number of folds in cross-validation) on the three datasets.

In Figure 6(left) we see that the proposed models incurs much less prediction error than the TOT model at every training/test split. Similar behavior from both the models are also found for the IEEE publication dataset in Figure 6(middle) and the Stackoverflow dataset in Figure 6(right).

C. Evolution of the Topics

One crucial aspect of the proposed model is its ability to learn the change of influence of each topic over time. Such evolutionary behavior is represented by the beta distribution \( \psi_z \) for each topic \( z \). After inferring the two parameters \( \bar{\alpha}_z \) and \( \bar{\beta}_z \) of \( \psi_z \) through Algorithm 1, we can compute the probability of topic \( z \) at each timestamp \( t \) (normalized) by the following equation:

\[
p(z, t) = \frac{1}{B(\bar{\alpha}_z, \bar{\beta}_z)} t^{\bar{\alpha}_z - 1} (1 - t)^{\bar{\beta}_z - 1}
\]

Figure 1 shows the relative probabilities of 10 topics learned by STEM from the IEEE publication dataset. We observe that the speech recognition topic has lost its influence gradually in last 20 years. Similar pattern is seen for the filtering theory. The image segmentation topic has shown more or less steady traction during this time period. On the other hand, interests in topics such as artificial intelligence, pattern clustering, pattern classification, wireless LAN has grown continuously after year 2000. The mobile radio topic started getting more importance after 2000 but seem to be eclipsed by other topics within a decade.

D. Label Prediction

In many cases a corpus contains a large number of documents with no or insufficient labels. Manually labeling large number of documents would always be expensive. In this section, we demonstrate how the proposed model can be
used to annotate new documents. We first divide the whole set of documents in the corpus into a training and a test set — training set containing two-thirds of the documents and the test set the rest — and learn the models only from the training documents. Now, for each test document we compute its topic distribution without using the labels. For each word token \( w \) in test document \( d \), we obtain a vector of probabilities \( \phi_w \) where \( \phi_{zw} \) is the probability of \( w \) being chosen from the topic \( z \). Now, by adding the probability vectors of all the words in document \( d \), we get a distribution of topics in this document. From this distribution, we choose top \( n \) topics with the highest probabilities as the predicted labels of this document. Then we compute the number of common labels in the predicted and original set of labels in this document.

The test documents we selected from the IEEE dataset had an average of 3.36 labels per document. Therefore, a perfect model that can predict all the labels correctly for each test document would have on average 3.36 labels in common between the predicted and the original label sets. Figure 7(left) shows the prediction performance of the three models for different number \( (n) \) of top labels. The proposed STEM and STEM-P models outperform the supervised LLDA model and predicts almost 80% of the labels when 8 to 10 top labels are considered. In Figure 7(right), we show the results of the same experiment on the Stackoverflow dataset. Average labels per document in this dataset is 3.23.

Both the plots in Figure 7 depicts that considering dynamics of the corpus and incorporating topical evolution in the model structure helped STEM models to estimate the labels better than the static model LLDA for both the real world datasets.

We also pose the label prediction problem as a multi-label document classification task. Features for each document are extracted in the same way as the label prediction task above for both LLDA and STEM. For the classification task, we used a one-vs-rest SVM classifier. Performance of the models were computed by Macro-F1 and Micro-F1 scores using 10-fold cross validation. In Figure 8 we see that STEM models outperform LLDA on both metrics on both real world datasets.

E. Scalability of the Model

In this section, we perform experiments on the scalability of our STEM model, i.e., how quickly the model converges as we distribute the inference process over increasing number of CPUs. In topic modeling, convergence is usually measured in terms of perplexity. Perplexity is defined as the inverse of the geometric mean per-word likelihood, and a lower perplexity indicates a better fit to the data.

\[
Perplexity = \exp \left( -\frac{\sum_{d=1}^{D} \sum_{i=1}^{N_d} \log p(w_{di} | \theta, \phi, \psi)}{\sum_{d=1}^{D} N_d} \right) \tag{12}
\]

Figure 9 shows the change of perplexity of the STEM model for different number of CPUs. We see that the model converges significantly faster as we add more CPUs. Moreover, the speedups achieved by parallelization are much higher for the larger dataset. Since the sequential STEM model converges in less than 10 minutes on the IEEE dataset (which is relatively smaller), impact of parallelization is not very significant. However, on the large Stackoverflow dataset, parallelization with 12 CPUs gives almost four times speedup in convergence time (30 minutes vs. 120 minutes).

We also investigate the benefit of the asynchronous approach used in our distributed inference over a synchronous approach. Since the STEM itself is a new model, there is no comparable synchronous model in the literature. For comparison, we implemented a synchronous version of STEM-P following the distributed LDA model of Newman [18]. We measured the perplexity of each model after every five minutes, and a model is considered to have converged if perplexity difference drops to less than 1% from the
Figure 10. Convergence time of synchronous and asynchronous STEM-P. Asynchronous approach gives 20% to 30% speedup over the synchronous version.

previous timestamp. Figure 10 shows the convergence times of both versions of STEM-P on the Stackoverflow dataset for different number of topics. We observe that our asynchronous version achieves significant speedup (20% to 30%) in convergence time over its synchronous counterpart.

V. CONCLUSION

This paper presents Supervised Topical Evolution Model (STEM) by incorporating textual content, temporal information, and document-level annotations simultaneously. Experimental results demonstrate that the proposed model can effectively extract highly interpretable and distinctive topics and capture their evolutionary patterns. In this paper we considered that only document-level annotations are available. In the future we will leverage content-level annotations. Content-level annotations are sometimes useful for analytic purpose. For example, mentions of specific locations, organizations or person of interest may form a topic. Another future direction is to generate hidden topics simultaneously with the topics parallel to labels.

REFERENCES


