Tag-Weighted Topic Model For Large-scale Semi-Structured Documents

Shuangyin Li, Jiefei Li, Guan Huang, Ruiyang Tan, and Rong Pan

Abstract—To date, there have been massive Semi-Structured Documents (SSDs) during the evolution of the Internet. These SSDs contain both unstructured features (e.g., plain text) and metadata (e.g., tags). Most previous works focused on modeling the unstructured text, and recently, some other methods have been proposed to model the unstructured text with specific tags. To build a general model for SSDs remains an important problem in terms of both model fitness and efficiency. We propose a novel method to model the SSDs by a so-called Tag-Weighted Topic Model (TWTM). TWTM is a framework that leverages both the tags and words information, not only to learn the document-topic and topic-word distributions, but also to infer the tag-topic distributions for text mining tasks. We present an efficient variational inference method with an EM algorithm for estimating the model parameters. Meanwhile, we propose three large-scale solutions for our model under the MapReduce distributed computing platform for modeling large-scale SSDs. The experimental results show the effectiveness, efficiency and the robustness by comparing our model with the state-of-the-art methods in document modeling, tags prediction and text classification. We also show the performance of the three distributed solutions in terms of time and accuracy on document modeling.

Index Terms—semi-structured documents, topic model, tag-weighted, variational inference, large-scale, parallelized solutions

1 Introduction

In the evolution of the Internet, there have been a huge amount of documents in many web applications. Such kinds of documents with both plain text data and document metadata (tags, which can be viewed as features of the corresponding document) are called the Semi-Structured Documents (SSDs). How to characterize the semi-structured document data becomes an important issue addressed in many areas, such as information retrieval, artificial intelligence and data mining etc. The tags can be more important than the text data in document mining. For example, in IMDB [1], the world's most popular and authoritative source for movie, TV and celebrity content, each movie has lots of tags, like director, writers, stars, country, language and so on, and a storyline as text data. Given a movie with a tag "Dick Martin", we may have an idea that it has a higher chance to be a comedy, without read the full text of its storyline or watch it. Another example is that in a collection of scientific articles, each document has a list tags(authors and keywords). Before read the main text of paper, we would know what it talks about after we see the authors or the keywords that the paper provides.

Many solutions have been proposed to deal with the semi-structured documents (e.g., SVD, LSI), and shown to be useful in document mining [11], [25], [35], [33], e.g., text classification and structural information exploiting. For document modeling, topic models have been used to be a powerful method of analyzing and modeling of document corpora, using Bayesian statistics and machine learning to discover the thematic contents of untagged documents. Topic models can discover the latent structures in documents and establish links between them, such as latent Dirichlet allocation (LDA) [9]. However, as an unsupervised method, only the words in the documents are modeled in LDA. Thus, LDA could only treat the tags as word features rather than a new kind of information for document modeling.

To model semi-structured documents needs to consider the characteristics of different kinds of objects, including word, topic, document, and tag, and the relationship among them. In this problem, topic is a kind of hidden objects, and the other three are the observations. Relative to tag, word and document are objective; tag can be either objective (e.g., author and venue information of publications) and subjective (e.g., tags in social bookmark marked by people). Similar to the topic models, we should consider binary relationships between the pairs of the objects, including topic-word and document-topic. In addition, we may consider the binary relationships, like tag-word, tag-topic, tag-document, and tag-tag. The tag-document relationship implies that we should consider the weights of the tags in each document. The tag-topic and tag-tag relationships can be more complicated, thus are difficult to model. Some earlier works consider certain tags. For example, the author-topic model in [31] considers the authorship information of the documents to be modeled. In this work, we

• *Shuangyin Li, Jiefei Li, Guan Huang, Ruiyang Tan, and Rong Pan's E-mails: shuangyinli@cse.ust.hk, {lijiefei@mail2., huangg6@mail2., tanry@mail2., panr@}sysu.edu.cn. Submitted and reviewed by IEEE Transactions on Knowledge and Data Engineering (TKED).*

1. http://www.imdb.com

don't limit the types and number of the tags in each document. In an extreme case, where there is no tag in any document, the new model may degenerates into LDA. On the other hand, since the tags can be created by some people, they should be relevant to topics of the documents; however, some of them may be correlated, redundant, and even noisy. Therefore, the tag-topic relationships should be general enough and we should also model the weights of the tags in each document.

In the past few years, researchers have proposed approaches to model documents with tags or labels [26], [29], [30]. For example, Labeled LDA [29] assumes there is no latent topics and each topic is restricted to be associated with the given labels. PLDA assumes that each topic is associated with only one label [30]. However, both Labeled LDA and PLDA have implicit assumptions that the given labels should be strongly associated with the topics to be modeled or the labels are independent to each other.

Another problem is that we would get into trouble when we need to deal with large-scale semi-structured documents. A variety of algorithms have been used to estimate the parameters of these proposed topic models for mining documents, such as Monte Carlo Markov chain (MCMC) sampling techniques [1], [19], variational methods [3] and others methods [2], [32]. For sampling methods, actually, we may have to appeal to a tailored solution of MCMC [7] for a particular model, which would impede the requirement of convergence properties and speed, especially when the corpus comprise millions of words. Variational methods as approximation solutions to some extent improve the learning speed. However, it would also be ineffective on learning speed and model accuracy when it comes to a large-scale corpus.

In this paper, we propose a framework of Tag-Weighted Topic Model (TWTM) to represent the text data and the various tags with weights to evaluate the importance of the tags. Besides learning the topic distributions of documents and generating the topic distributions over words, the framework also infers the topic distributions of tags. The weights of observed tags in each document, which we infer from the dataset, give us an opportunity to provide a method to rank the tags.

In many web applications, not all the documents in the corpora have tags. There are lots of documents only consist of words without any tags which maybe removed after data preprocessing for denoising. Only consider the weights among tags would not hold this case. To address this problem, we also propose a more flexible model called Tag-Weighted Dirichlet Allocation (TWDA) as an extended model. It is based on TWTM, and learns the weights among a Dirichlet prior and the given tags, not just among the tags. Therefore, TWDA handles not only the semi-structured documents, but also the unstructured documents. For the unstructured documents, TWDA degenerates into latent Dirichlet allocation (LDA). For hybrid corpora which consist of both the semi-structured documents and unstructured documents, TWDA can handle this complex type of corpora more effectively and easily.

For the challenge of modeling large-scale corpora, we propose three distributed schemes for the framework of TWTM model in MapReduce programming framework [16]. The proposed model has four principal contributions.

1) It is a novel topic modeling method to model the semi-structured documents, not only generating the topic distributions over words, but also inferring the topic distributions of tags.
2) The TWTM leverages the weights among the observed tags in a document to evaluate the importance of the tags using a function of tag-weighted topic assignment process. The weights are associated with the observed tags in a document providing a way to rank the tags. In addition, this could be used to predict latent tags in the document.
3) The framework of tag-weighted process is easy to extend for many different real world applications. For example, with the extended model TWDA, we can handle both the multi-tag documents and non-tag documents simultaneously, which is very useful to process some complicated web applications.
4) Three distributed solutions for TWTM have been proposed that focus on challenges of working at a large-scale semi-structured documents in MapReduce programming framework.

The rest of the paper is organized as follows. In Section 2, we first analyze and discuss related works. In Section 3, after introducing the notations, we present the novel topic modeling framework of TWTM, and give the methods of learning and inference. In Section 4, we show the extended model TWDA, and give the process of learning and inference. In Section 5, we will give the theoretical analysis to discuss the differences between TWTM and TWDA, comparing the other topic models. In Section 6, we propose three distributed solutions of TWTM for a large-scale semi-structured documents. In Section 7, we present the experimental results on three domains to show the performance of the proposed method in document modeling, text classification and the effectiveness and efficiency of the three large-scale solutions on a large scale semi-structured documents modeling. We end the paper in Section 8.

2 Related Works

Topic models provide an amalgam of ideas drawn from mathematics, computer science, and cognitive science to help users understand unstructured data.

There are many topic models proposed and shown to be powerful on document analyzing, such as in [28], [20], [9], [8], [10], [14], which have been applied to many areas, including document clustering and classification [12], and information retrieval [34]. They are extended to many other topic models for different situation of applications in analyzing text data [21], [23], [36]. However, most of these models only consider the textual information and can only treat the tag information as plain text as well.

TMBP [17] and cFTM [15] propose the methods to make use of the contextual information of documents for topic modeling. TMBP is a topic model with biased propagation to leveraging contextual information, the authors and venue. TMBP needs to predefine the weights of the author and venue information on word assignment, which limits the usefulness in real applications. The method of cFTM has a very strong assumption that each word is associated with only one tag, either author or venue. In many applications, this assumption may not hold.

Several models have been proposed to take advantage of tags or labels, such as Labeled LDA [29], DMR [26] and PLDA [30], or modeling relationships among several variables, such as Author-Topic Model [31]. Labeled LDA [29] get the topic distribution for a document through picking out the several hyper-parameter components that correspond to its labels, and draw the topic components by the new hyper-parameter without inferring the topic distribution of labels. Labeled LDA does not assume the existence of any latent topics [30]. PLDA [30] provides another way of modeling the tagged text data, which assumes the generation topics assignment is limited by only one of the given tags for one word, and in the training process, PLDA assumes that each topic takes part in exactly one label, and may optionally share global label present on every document. In Author Topic Model, it obtains the topic distributions of authors, without giving the importance weights among the given authors in each document. DMR [26] is a Dirichlet-multinomial regression topic model that includes a log-linear prior on the document-topic distributions, which is an exponential function of the given features of the document. However, DMR doesn't output the tag weights either [31], which is useful for tag ranking.

So in this work, we propose a tag-weighted topic modeling framework which leverages the tag information given in a document by a list of weight values to model the topic distribution of the document. Meanwhile, for a mixture collection of semi-structured documents and unstructured documents, we present an extended model called tag-weighted Dirichlet Allocation which considers both a Dirichlet prior and the tags by the weight values among them. Based on the framework of Tag-Weighted Topic Model, we also show three large-scale solutions under the MapReduce distributed computing platform for large-scale semi-structured documents.

3 TWTM Model and Algorithms

In this section, we will mathematically define the tag-weighted topic model (TWTM), and discuss the learning and inference methods.

3.1 Notation

Similar to LDA [9], we formally define the following terms. Consider a semi-structured corpus, a collection of M documents. We define the corpus $D = \{(\mathbf{w}^1, \mathbf{t}^1), \ldots, (\mathbf{w}^M, \mathbf{t}^M)\}$, where each 2-tuple $(\mathbf{w}^d, \mathbf{t}^d)$ denotes a document, the bag-of-word representation $\mathbf{w}^d = (w_1^d, \ldots, w_N^d)$, $\mathbf{t}^d = (t_1^d, \ldots, t_L^d)$ is the document tag vector, each element of which being a binary tag indicator, and L is the size of the tag set in the corpus D. For the convenience of the inference in this paper, $\mathbf{t}^d$ is expanded to a $l^d \times L$ matrix T^d, where l^d is the number of tags in the document d. For each row number $i \in \{1, \ldots, l^d\}$ in T^d, $T_{i\cdot}^d$ is a binary vector, where $T_{ij}^d = 1$ if and only if the i-th tag of the document d[2] is the j-th tag of the tag set in the corpus D. In this paper, we wish to find a probabilistic model for the corpus D that assigns high likelihood to the documents in the corpus and other documents alike utilizing the given tag information.

3.2 Tag-Weighted Topic Model

TWTM is a probabilistic graphical model that describes a process for generating a semi-structured document collection. In the previous topic models, a document d is typically characterized by a multinomial distribution over topics, θ^d, and each topic k is represented by ψ_k, over words in a vocabulary. Take LDA [9] as an example, the generative process of topic distribution of document d is assumed as follows.

$$\text{Choose } \theta^d \sim \text{Dirichlet}(\alpha),$$
$$\text{and choose } z_{ni} \sim \text{Multinomial}(\theta^d),$$

where α is the hyperparameter of θ^d. In LDA, the topic distribution θ^d is drawn from a hyperparameter α, without considering the given tags. However, the tag information should be more useful for the generation of θ^d than a Dirichlet prior.

In this paper, we use ϑ^d, instead of θ^d, to denote the topic distribution of document d as shown in Figure 1. Let θ represent a $L \times K$ topic distribution matrix over the tag set, where K is the number of topics. Let ψ represent a $K \times V$ distribution matrix over words in the dictionary, where V is the number of words in the dictionary of D. Similar to LDA, TWTM models the document d as a mixture of underlying

2. Note that we can sort the tags of the document d by the index of the tag set of the corpus D.

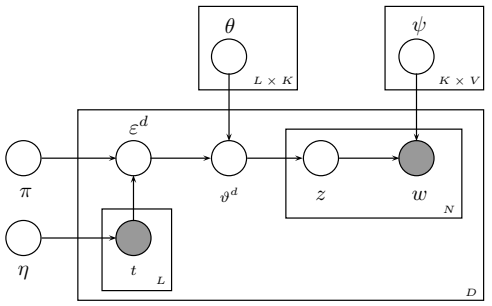


Fig. 1. Graphical model representation for TWTM, where θ is distribution matrix of the whole tags, ψ is distribution matrix of words, ϵ^d represents the weight vector of the tags, and ϑ^d indicates the topic components for each document. π is a Dirichlet prior and η is a Bernoulli prior.

topics and generates each word from one topic. The topic proportions ϑ^d of the document d is a mixture of tag-topic distributions, not only controlled by a hyperparameter described as in LDA.

The generative process for TWTM is given in the following procedure:

1) For each topic $k \in \{1, \ldots, K\}$, draw $\psi_k \sim \text{Dir}(\beta)$, where β is a V dimensional prior vector of ψ.
2) For each tag $t \in \{1, \ldots, L\}$, draw $\theta_t \sim \text{Dir}(\alpha)$, where α is a K dimensional prior vector of θ.
3) For each document d:
   a) For each $l \in \{1, \ldots, L\}$, draw $t_l^d \sim \text{Bernoulli}(\eta_l)$.
   b) Generate T^d by $\mathbf{t}^d$.
   c) Draw $\varepsilon^d \sim Dir(T^d \times \pi)$.
   d) Generate $\vartheta^d = (\varepsilon^d)^{\mathrm{T}} \times (T^d \times \theta)$.
   e) For each word w_{di}:
      i) Draw $z_{di} \sim$Mult(ϑ^d).
      ii) Draw $w_{di} \sim$Mult($\psi_{z_{di}}$).

In this process, Dir($\cdot$) designates a Dirichlet distribution, Mult($\cdot$) is a multinomial distribution, and π is a $L \times 1$ column vector, a Dirichlet prior. Note that ε^d indicates the weight vector of the observed tags in constituting the topic proportions of the document d, and $(\varepsilon^d)^{\mathrm{T}}$ is the transpose of ε^d. Furthermore, ε^d is drawn from a Dirichlet prior which obtained by the matrix multiplication of $T^d \times \pi$. Clearly, the result of $T^d \times \pi$ will be a ($l^d \times 1$) vector whose dimension is depended on the number of the observed tags in the document d.

In Step 3, for one document d, we first generate the document's tags t_l^d using a Bernoulli coin toss with a prior probability η_l, as shown in step (a). After draw out the ε^d, we generate the ϑ^d through ε^d, T^d and θ. The remaining part of the generative process is just familiar with LDA [9]. As shown above, in TWTM, we introduce a novel way to model the topic proportions of semi-structured document by document-special tags and text data. The key discussed in this paper is the tag's weight topic assignment by which ϑ^d is generated through ε^d, T^d, and θ, which provides an effective and direct method to infer the weights of the tags.

3.3 Tag-Weighted Topic Assignment

As we assume that all the observed tags in the document d make contributions to infer the topic distribution ϑ^d of the document, it is expected that different tags works corresponding to their own weights. For example, in some blog application, a blog has tags of an author, a blog's date, a blog category and a blog's url. Clearly, compared to other tags, the tag of the author plays the most important role in constituting the topic components of the blog.

The function of how to leverage the tag information or contextual to infer topic distribution of a document is defined as follows:

$$\vartheta \longleftarrow f(t_1, \cdots, t_l),$$

where $f(\cdot)$ is the way of making use of the tag information. Topic models using tag information or contextual take advantage of the different $f(\cdot)$ in the past. In TWTM, we assume that ϑ^d is made up by all the observed tags with their own weights. Figure 1 shows that how TWTM works in a probabilistic graphical model. As shown in Figure 1, ϑ^d is controlled by two sides, the topic distributions over tags θ, and the weights of the given tags of the document d. It is important to distinguish TWTM from the Author-Topic Model [31]. In the author-topic model, the words w is chose only by one of the given tags' distribution, while in TWTM, for word w, all the observed tags in the document would make the contributions.

The $f(\cdot)$ in the proposed model is assumed as this, for the document d,

$$f(\vartheta^d) = (\varepsilon^d)^{\mathrm{T}} \times T^d \times \theta,$$

where the linear multiplication of $(\varepsilon^d)^{\mathrm{T}}$, T^d and θ maintains the condition of $\sum_{k=1}^{K} \vartheta_k^d = 1$ without normalization of ϑ^d, since ε^d and θ satisfy

$$\sum_{i=1}^{l^d} \varepsilon_i^d = 1, \sum_{k=1}^{K} \theta_{lk} = 1.$$

Firstly, we pick out the topic distributions of the given tags in the document d from θ by $T^d \times \theta$, where T^d is a $l^d \times L$ matrix and θ is a $L \times K$ matrix. Here we define

$$\Theta^d = T^d \times \theta,$$

where the Θ^d is a $l^d \times K$ topic distribution matrix of the given tags in d as sub-components of θ. Secondly, ε^d is the weight vector of the observed tags in d, and each dimension of ε^d represents the weight or importance associated to the corresponding tag. Thus, ϑ^d is mixed by Θ^d with corresponding weight values.

$$\vartheta^d = (\sum_{i=1}^{l^d} \varepsilon_i^d \Theta_{i1}^d, \ldots, \sum_{i=1}^{l^d} \varepsilon_i^d \Theta_{ij}^d, \ldots, \sum_{i=1}^{l^d} \varepsilon_i^d \Theta_{iK}^d).$$

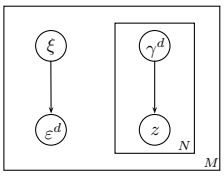


Fig. 2. Graphical model representation of the variational distribution used to approximate the posterior in TWTM.

With ϑ^d, TWTM generates all the words in the document d with the assumption of bag-of-words.

Based on the above framework, we can define a special topic assignment function $f(\cdot)$ in an extended model for a real world application.

3.4 Inference for TWTM

In the topic models, the key inferential problem that we need to solve is to compute the posterior distribution of the hidden variables given a document d. Given the document d, we can easily get the posterior distribution of the latent variables in the proposed model, as:

$$p(\varepsilon^d, \mathbf{z}|\mathbf{w}^d, T^d, \theta, \eta, \psi, \pi) = \frac{p(\varepsilon^d, \mathbf{z}, \mathbf{w}^d, T^d|\theta, \eta, \psi, \pi)}{p(\mathbf{w}^d, T^d|\theta, \eta, \psi, \pi)}. \quad (1)$$

In Eq. (1), integrating over ε and summing out z, we easily obtain the marginal distribution of d:

$$p(\mathbf{w}^d, T^d|\eta, \theta, \psi, \pi) = p(\mathbf{t}^d|\eta) \int p\left(\varepsilon^d|(T^d \times \pi)\right) \cdot \prod_{i=1}^{N} \sum_{z_i^d=1}^{K} p(z_i^d|(\varepsilon^d)^{\mathrm{T}} \times T^d \times \theta) p(w_i^d|z_i^d, \psi_{1:K})\, d\varepsilon^d.$$

In this work, we make use of mean-field variational EM algorithm [4] to efficiently obtain an approximation of this posterior distribution of the latent variables. In the mean-field variational inference, we minimize the KL divergence between the variational posterior probability and the true posterior probability through by maximizing the evidence lower bound (ELBO) $\mathcal{L}(\cdot)$ [8]. For a single document d, we obtain the $\mathcal{L}(\cdot)$ using Jensen's inequality:

$$\begin{aligned}\mathcal{L}(\xi_{1:l^d}, \gamma_{1:K}; \eta_{1:L}, \pi_{1:L}, \theta_{1:L}, \psi_{1:K}) &= E[\log p(T_{1:l^d}|\eta_{1:L})] + E[\log p(\varepsilon^d|T^d \times \pi)] \\ &+ \sum_{i=1}^{N} E[\log p(z_i|(\varepsilon^d)^{\mathrm{T}} \times T^d \times \theta)] \\ &+ \sum_{i=1}^{N} E[\log p(w_i|z_i, \psi_{1:K})] + H(q),\end{aligned}$$

where ξ is a l^d-dimensional Dirichlet parameter vector and γ is $1 \times K$ vector, both of which are variational parameters of variational distribution shown in Figure 2, and $H(q)$ indicates the entropy of the variational distribution:

$$H(q) = -E[\log q(\varepsilon^d)] - E[\log q(z)].$$

Here the exception is taken with respect to a variational distribution $q(\varepsilon^d, z_{1:N})$, and we choose the following fully factorized distribution:

$$q(\varepsilon^d, z_{1:N}|\xi_{1:L}, \gamma_{1:K}) = q(\varepsilon^d|\xi) \prod_{i=1}^{N} q(z_i|\gamma_i).$$

The dimension of parameter ξ is changed with different documents. It could be difficult to compute the expected log probability of a topic assignment by the way of tag-weighted topic assignment used in TWTM.

Then, we maximize the lower bound $\mathcal{L}(\cdot)$ with respect to the variational parameters ξ and γ, using a variational expectation-maximization(EM) procedure as follows.

3.4.1 Variational E-step

We first maximize $\mathcal{L}(\cdot)$ with respect to ξ_i for the document d. Maximize the terms which contain ξ:

$$\begin{aligned}\mathcal{L}_{[\xi]} = &\sum_{i=1}^{l^d} (\sum_{l'=1}^{L} \pi_{l'} T_{il'}^d - 1)(\Psi(\xi_i) - \Psi(\sum_{j'=1}^{l^d} \xi_{j'})) \\ &+ \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \cdot \sum_{j=1}^{l^d} \log \theta_k^{(j)} \xi_j / \sum_{j'=1}^{l^d} \xi_{j'} \\ &- \log \Gamma(\sum_{i=1}^{l^d} \xi_i) + \sum_{i=1}^{l^d} \log \Gamma(\xi_i) \\ &- \sum_{i=1}^{l^d} (\xi_i - 1)(\Psi(\xi_i) - \Psi(\sum_{j'=1}^{l^d} \xi_{j'})),\end{aligned} \quad (2)$$

where $\Psi(\cdot)$ denotes the digamma function, the first derivative of the log of the Gamma function. Here we use gradient descent method to find the ξ to make the maximization of $\mathcal{L}_{[\xi]}$.

Next, we maximize $\mathcal{L}(\cdot)$ with respect to γ_{ik}. Adding the Lagrange multipliers to the terms which contain γ_{ik}, taking the derivative with respect to γ_{ik}, and setting the derivative to zero yields, we obtain the update equation of γ_{ik}:

$$\gamma_{ik} \propto \psi_{k,v^{w_i}} \exp\{\sum_{j=1}^{l^d} \log \theta_k^{(j)} \frac{\xi_j}{\sum_{j'=1}^{l^d} \xi_{j'}}\}, \quad (3)$$

where v^{w_i} denotes the index of w_i in the dictionary.

In E-step, we update the ξ and γ for each document with the initialized model parameters. For the reason of different document with different number of tags, we have to keep all the ξ updated by each document for the M-step estimation.

3.4.2 M-step estimation

The M-step needs to update four parameters: η, the tagging prior probability, π, the Dirichlet prior of the tags' weights, θ, the topic distribution over all tags in the corpus, and ψ, the probability of a word under a topic. Because each document's tag-set is observed, the Bernoulli prior η is unused included for model completeness. For a given corpus, the η_i is estimated by adding up the number of i-th tag which appears in the corpus.

For the document d, the terms that involve the Dirichlet prior π:

$$\mathcal{L}_{[\pi]} = \log\Gamma\left(\sum_{i=1}^{l^d}(T^d\times\pi)_i\right) - \sum_{i=1}^{l^d}\log\Gamma\left((T^d\times\pi)_i\right) + \sum_{i=1}^{l^d}\left((T^d\times\pi)_i - 1\right)\left(\Psi(\xi_i) - \Psi(\sum_{j=1}^{l^d}\xi_j)\right), \tag{4}$$

where $(T^d\times\pi)_i = \sum_{i=1}^{l^d}\sum_{l=1}^{L}\pi_l T^d_{il}$. We use gradient descent method by taking derivative of Eq. (4) with respect to π_l on the corpus to find the estimation of π.

To maximize with respect to θ and ψ, we obtain the following update equations:

$$\theta_{lk} \propto \sum_{d=1}^{D}\sum_{i=1}^{N}\gamma^d_{ik}\frac{\xi^d_l t^d_l}{\sum_{l=1}^{L}(\xi^d_l t^d_l)}, \tag{5}$$

and

$$\psi_{kj} \propto \sum_{d=1}^{D}\sum_{i=1}^{N}\gamma^d_{ik}(w^d)^j_i. \tag{6}$$

We provide a detailed derivation of the variational EM algorithm for TWTM in Appendix A. And we show the variational expectation maximization (EM) procedure of TWTM in Algorithm 1.

Algorithm 1 The variational expectation maximization (EM) algorithm of TWTM

Input: a semi-structured corpora including totally V unique words, L unique tags, and the expected number K of topics.
Output: Topic-word distributions ψ, Tag-topic distributions θ, π, topic distribution ϑ^d and weight vector ε^d of each training document.
initialize π, and initialize θ and ψ with the constraint of $\sum_{k=1}^{K}\theta_{lk}$ equals 1 and $\sum_{i=1}^{V}\psi_{ki}$ equals 1.
repeat
  for each document d **do**
    update ξ^d with Eq. (2) using gradient descent method.
    update γ_{ik} with Eq. (3).
  end for
  update π with Eq. (4) using gradient descent method.
  update θ by Eq. (5).
  update ψ by Eq. (6).
until convergence

4 Tag-Weighted Dirichlet Allocation

In a real world application, a corpus is very likely to contain both semi-structured documents and unstructured documents. Many documents in the corpus have no tags, just with unstructured text data. In this case, TWTM does not work, which generates the topic distribution of a document by leveraging the weights among the observed tags. Our proposed solution to the problem is to add a Dirichlet prior to the topic distribution ϑ^d, which means that we learn the weights among the Dirichlet prior and the given tags, not just among the tags. We call this solution Tag-Weighted Dirichlet Allocation (TWDA). When handling the unstructured documents in a hybrid corpus, TWDA degenerates into LDA [9] which just draws the topic proportions for a document from a Dirichlet distribution.

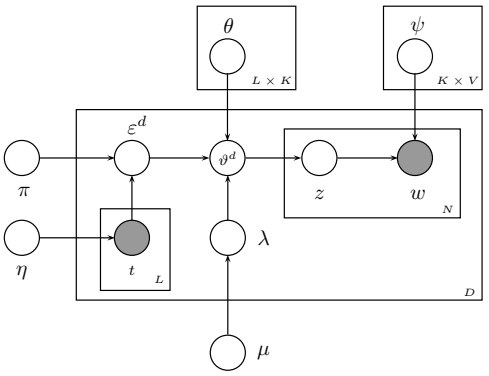


Fig. 3. Graphical model representation for TWDA, where μ is a Dirichlet prior of λ.

As an extended model of TWTM, TWDA uses the same parameter notations. Unlike TWTM, for the convenience of the inference in TWDA, $\mathbf{t}^d$ is expanded to a $l^d \times (L+1)$ matrix T^d, where l^d is one more than the number of the given tags in the document d (For example, if the document d has five tags, l^d is six). For each row number $i \in \{1,\ldots,l^d\}$ in T^d, $T^d_{i\cdot}$ is a binary vector, where $T^d_{ij} = 1$ if and only if the i-th tag of the document d is the j-th tag of the tag set in the corpus D. Note that, we set the last dimension of the last row in T^d to 1, and the other dimensions of the last row equal to 0 for all documents. The detail of the above setting will be shown later.

TWDA defines a Dirichlet prior μ over a latent topic distribution of a document, and mixes the latent topic proportion with these topic distributions of the given tags by importance or weight (tag-weighted) to form the final topic distribution of the document. Figure 3 shows the graphical model representation of TWDA, and the generative process for TWDA is given in the following procedure:

1) For each topic $k \in \{1,\ldots,K\}$, draw $\psi_k \sim \text{Dir}(\beta)$, where β is a V dimensional prior vector of ψ.
2) For each tag $t \in \{1,\ldots,L\}$, draw $\theta_t \sim \text{Dir}(\alpha)$, where α is a K dimensional prior vector of θ.
3) For each document d:
   a) Draw $\lambda \sim Dir(\mu)$.
   b) Generate T^d by $\mathbf{t}^d$.
   c) Draw $\varepsilon^d \sim Dir(T^d \times \pi)$.
   d) Generate $\vartheta^d = (\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda})$.
   e) For each word w_{di}:

i) Draw $z_{di} \sim$Mult(ϑ^d) .

ii) Draw $w_{di} \sim$Mult($\psi_{z_{di}}$) .

Note that, L is the number of tags appeared in the corpora and K is the number of topics. Different from TWTM, here π is a $(L+1)\times 1$ column vector and μ is a $K \times 1$ column vector. Both of them are Dirichlet prior. λ is a $1 \times K$ row vector which is drawn from μ. $(\varepsilon^d)^{\mathrm{T}}$ is the transpose of ε^d, and ε^d is drawn from a Dirichlet prior which obtained by the matrix multiplication of $T^d \times \pi$. Clearly, the result of $T^d \times \pi$ will be a $(l^d \times 1)$ vector whose dimension is depended on the number of the observed tags in the document d. Note that, l^d is one more than the number of tags given in d as we described above.

In other words, we treat the λ as a topic distribution of one latent tag, the Dirichlet prior μ. Each document is controlled by a latent tag, that is the same idea both TWDA and Latent Dirichlet Allocation (LDA). The form of $(\frac{\theta}{\lambda})$ is the augmented matrix of θ and λ, which represents that we add the vector λ to the matrix θ as the last row, so $(\frac{\theta}{\lambda})$ becomes a $(L+1)\times K$ matrix. As we show above, T^d is the matrix form of the given tags in the document d, and the last row of T^d is a binary vector, of which only the last dimension equals to 1 and the others equal 0. Here we define

$$\Theta^d = T^d \times (\frac{\theta}{\lambda}).$$

Clearly, Θ^d is a $l^d \times K$ matrix, whose last row is λ. Actually, the purpose of Θ^d is to pick out the rows corresponded to the tags appeared in d from tag-topic distribution matrix θ.

The key idea of tag-weighted Dirichlet allocation is to model the topic proportions of semi-structured documents by document-special tags and text data. Different from LDA, the topic proportion of one document assumed in this model is controlled not only by a Dirichlet prior μ, but also by all the observed tags. The way to generate the normalized topic distribution of the document d is that we mix both Dirichlet allocation and tags information through a weight vector ε^d. Thus, we use the function $f(\cdot)$ of topic assignment to obtain the topic distribution of d by

$$f(\vartheta^d) = (\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda}).$$

It is worth to note that the ε^d is draw by a Dirichlet prior π, each row of θ is draw by a Dirichlet prior α, and λ is draw by a Dirichlet prior μ, so ε^d and θ satisfy

$$\sum_{i=1}^{l^d} \varepsilon_i^d = 1, \sum_{k=1}^{K} \theta_{lk} = 1, \text{and } \sum_{k=1}^{K} \lambda_k = 1.$$

Therefore, the linear multiplication of $(\varepsilon^d)^{\mathrm{T}}$, T^d, θ and λ maintains the condition of $\sum_{k=1}^{K} \vartheta_k^d = 1$ without normalization of ϑ^d. With ϑ^d, the topic proportions of the document d, the remaining part of the generative process is just familiar with LDA.

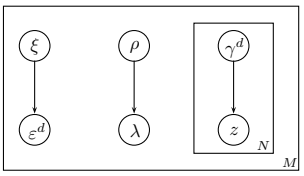


Fig. 4. Graphical model representation of the variational distribution used to approximate the posterior in TWDA.

4.1 Inference for TWDA

In TWDA, we treat π, μ, η, θ and ψ as unknown constants to be estimated. Similar to TWTM, the marginal distribution of d is not efficiently computable as follows:

$$\begin{aligned} p(\mathbf{w}^d, T^d|\eta, \theta, \psi, \pi, \mu) = p(\mathbf{t}^d|\eta) \int p\left(\varepsilon^d|(T^d \times \pi)\right) \\ \cdot p(\lambda|\mu) \prod_{i=1}^{N} \sum_{z_i^d=1}^{K} p(z_i^d|(\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda})) \\ \cdot p(w_i^d|z_i^d, \psi_{1:K})\, d\varepsilon^d. \end{aligned}$$

In this case, We also use a variational expectation-maximization (EM) procedure to carry out approximate maximum likelihood estimation of TWDA.

4.1.1 Variational inference

In TWDA, we use the following fully factorized distribution as shown in Figure 4:

$$q(\varepsilon^d, \lambda^d, z_{1:N}|\xi_{1:L}, \rho_{1:K}, \gamma_{1:K}) = q(\varepsilon^d|\xi) q(\lambda^d|\rho) \prod_{i=1}^{N} q(z_i|\gamma_i),$$

and the entropy of the variational distribution will be

$$H(q) = -E[\log q(\varepsilon^d)] - E[\log q(\lambda)] - E[\log q(z)].$$

For the variational parameter ξ, we take the terms which contain ξ out of the evidence lower bound (ELBO) $\mathcal{L}(\cdot)$ of TWDA to form $\mathcal{L}_{[\xi]}$, and we use gradient descent method to find the ξ to make the maximization of $\mathcal{L}_{[\xi]}$:

$$\begin{aligned} \mathcal{L}_{[\xi]} = \sum_{i=1}^{l^d} (\sum_{l'=1}^{L+1} \pi_{l'} T_{il'}^d - 1)(\Psi(\xi_i) - \Psi(\sum_{j'=1}^{l^d} \xi_{j'})) \\ + \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \cdot \sum_{j=1}^{l^d} C_k^{(j)} \frac{\xi_j}{\sum_{j'=1}^{l^d} \xi_{j'}} \\ - \log \Gamma(\sum_{i=1}^{l^d} \xi_i) + \sum_{i=1}^{l^d} \log \Gamma(\xi_i) \\ - \sum_{i=1}^{l^d} (\xi_i - 1)(\Psi(\xi_i) - \Psi(\sum_{j'=1}^{l^d} \xi_{j'})), \end{aligned} \tag{7}$$

where

$$C_k^{(j)} = \begin{cases} \log \theta_k^{(j)} & j \in \{1, \cdots, l^d - 1\} \\ \Psi(\rho_k) - \Psi(\sum_{j'=1}^{K} \rho_{j'}) & j = l^d \end{cases},$$

and $\Psi(\cdot)$ denotes the digamma function, the first derivative of the log of the Gamma function.

In particular, by computing the derivatives of the $\mathcal{L}(\cdot)$ and setting them equal to zero, we obtain the following pair of update equations for the variational parameters ρ^d and γ_{ik}:

$$\rho_i \propto \mu_i + \sum_{n=1}^{N} \gamma_{ni} \cdot \frac{\xi_{l^d}}{\sum_{j=1}^{l^d} \xi_j}, \tag{8}$$

$$\gamma_{ik} \propto \psi_{k,v^{w_i}} \exp\{\sum_{j=1}^{l^d} C_k^{(j)} \frac{\xi_j}{\sum_{j'=1}^{l^d} \xi_{j'}}\}, \tag{9}$$

where v^{w_i} denotes the index of w_i in the dictionary.

In the E-step, we update the variational parameters ξ, ρ and γ for each document with the initialized model parameters. We show the detailed derivation of the variational parameters for TWDA in Appendix B.

4.1.2 Model Parameter Estimation

There are four model parameters that need to estimate in M-step, π, the Dirichlet prior of the tags' weights, θ, the topic distribution over all tags in the corpus, ψ, the probability of a word under a topic, and μ, a Dirichlet prior of model. In TWDA, we can estimate π, θ and ψ as same as in TWTM.

Different from TWTM, TWDA has an extra Dirichlet prior μ. The involved terms of μ are:

$$\begin{aligned}\mathcal{L}_{[\mu]} = &\sum_{d=1}^{D}(\log\Gamma(\sum_{j=1}^{K}\mu_j) - \sum_{i=1}^{K}\log\Gamma(\mu_i) \\ &+ \sum_{i=1}^{K}(\mu_i - 1)(\Psi(\rho_i^d) - \Psi(\sum_{j=1}^{K}\rho_j^d))).\end{aligned} \tag{10}$$

We can invoke the linear-time Newton-Raphson algorithm to estimate μ as same as the Dirichlet parameter described in LDA [9].

In the variational expectation maximization (EM) procedure of TWDA, we update the variational parameters ξ^d, ρ and γ_{ik} with Eqs. (7), (8) and (9) respectively in the E-step. In the M-step, besides the update of π, θ and ψ, we also update μ with Eq. (10) by Newton-Raphson algorithm. The detailed derivation of the model parameter estimation in TWDA is shown in Appendix B.

5 Analysis of TWDA

In TWDA, we introduce a better way to directly model the semi-structured documents and unstructured documents by adding a latent tag to each documents, which the topic distribution of a document is controlled by the observed tags and one latent tag. In LDA, the topic distribution of a document is drawn from a hyperparameter, without considering the given tags, and while in TWTM, the topic distribution is controlled by a list of given tags with corresponding weight values. The main difference among the models which handle the unstructured text (e.g., LDA and CTM [7]) or the semi-structured documents (e.g., ATM [31], Label-LDA [29], DMR [26] and PLDA [30]) is the function that how to generate the topic distribution of a document, or, in other words, the assumption that what distribution the topic of a document follows.

In TWDA, the topic proportions ϑ^d for a document d is obtained by the following function:

$$\vartheta^d = (\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda})$$

When we ignore the tags in a document, the T^d in Eq. (5) becomes a binary row vector and the last dimension equals to 1 and the others are 0. In this case, $(\frac{\theta}{\lambda})$ is simplified to λ:

$$\begin{aligned}\vartheta^d &= (\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda}) \\ &= \lambda.\end{aligned}$$

The topic distribution of d is simplified to λ, and as we shown above, λ is draw by a Dirichlet prior μ. It means that the topic proportions for the document d as a draw from a Dirichlet distribution which is the basic assumption of LDA [9]. In others words, when handling the unstructured documents, TWDA degenerates into LDA.

In other words, the topic distribution of a document in TWTM is the weighted average of the topic distributions of the given tags, and to some extent, it is a linear relation between the topic distribution of a document and the tags. While, in TWDA, with the addition of the Dirichlet prior μ, which is equal to generate a latent tag for each document with a special topic distribution, it is a non-linear topic generation procedure in each document.

6 Large Scale Solutions

Currently, many web applications appear with large scale tagged documents, and highlight the issues of large scale semi-structured documents in many areas. In this paper, we propose and compare three different distributed methods based on the framework of TWTM, which focus on the challenge of working at a large scale, in the MapReduce programming framework.

Solution I

The first solution is a tailored parallel algorithm for TWTM. The learning and inference of the proposed model are based on variational method with an EM algorithm. Thus, we design a parallel algorithm for TWTM using MapReduce programming framework.

As shown above, we need to update the global parameters π, θ, and ψ for a corpus. Every document has associated with the corresponding variational parameters ξ and γ. The mapper computes these variational parameters for each document and uses them to generate the sufficient statistics to update π, θ, andψ.

And the reducer updates the global parameters π, θ, and ψ.

1) *Mapper*: For each document d, we compute γ^d using the update equation Eq. (3) and obtain ξ^d by Eq. (2). The sufficient statistics are kept for each document.
2) *Reducer*: The Reduce function adds the value to the global parameters θ and ψ using the sufficient statistics as in Eqs. (5), and (6).
3) *Driver*: The driver program marshals the entire inference process. At the beginning, the driver initializes all the model parameters K, L, θ, ψ, and π. The topic number K is user specified; the number of tags L is determined by the data; the initial value of π is given by the user, θ and ψ is randomly initialized. After each MapReduce iteration, the driver normalizes the global θ and ψ.

Note that, because π is a global parameter over the corpus, we have to update π at the end of each iteration in driver. However, this will lead to a large scale data migration to compute the π by Eq. (4), since π is associated with each document and different documents have different tags which affect the different dimensions in π. The whole corpus data would migrate to the single driver node. This could generate a bottleneck in the driver.

Solution II

On account of the bottleneck in Solution I, we optimize the calculation of π through an approximate method as the Solution II. The MapReduce procedure of Solution II is as follows.

1) *Mapper*: For each document d, we compute γ^d and ξ^d by Eqs. (2) and (3) and the sufficient statistics for updating θ and ψ. Different with Solution I, we obtain a π^s for each map data split s by Eq. (4).
2) *Reducer*: The Reduce function adds the value to the global parameters θ and ψ using the sufficient statistics as in Eqs. (5), and (6).
3) *Driver*: In the driver function, we only need to compute an average of $\pi^s, s \in (1, \cdots, S)$ where S is the total number of mapper in the cluster. The driver also normalizes the global θ and ψ for next iteration.

Solution II is an approximate solution of TWTM, which computes the π_s for each mapper and takes their average as the solution of π to avoid the large scale data migration.

Solution III

As shown in Eq. (4), $\pi_l, l \in (1, \cdots, L)$ is only associated with the documents who contain the l^{th} tag. Thus, before running TWTM, we can cluster the documents into several clusters with the condition that the documents which contain one or a plurality of the same tag should be in the same cluster. It means that the documents are divided into the mutually independent space by the tags. We show the detailed process of the clustering in Appendix C. The MapReduce procedure of Solution III is the following procedure.

1) *Mapper*: The input of mapper is clusters. For each cluster, we obtain a π^c for the cluster c, $c \in (1, \cdots, C)$, where C is the number of document clusters, which is the sufficient statistics for updating θ and ψ.
2) *Reducer*: The Reduce function adds the value to the global parameters θ and ψ using the sufficient statistics as in Eqs. (5), and (6).
3) *Driver*: In the driver, we update θ and ψ. Note that there is no need to recompute π, and we combine all the π^c to obtain the final π for current iteration.

Solution III is an exact solution for TWTM, and it is equivalent to Solution I when the documents are all belong to one cluster. However, Solution III provides a more efficient method than Solution I, and this depends on the result of document clustering, which would be anther bottleneck in some real applications. Although Solution II is an approximate method for modeling the semi-structured documents, it effectively avoids the bottleneck brought by Solution I and Solution III. The experiment results in Section 7 show that Solution II works better than Solution I and Solution III.

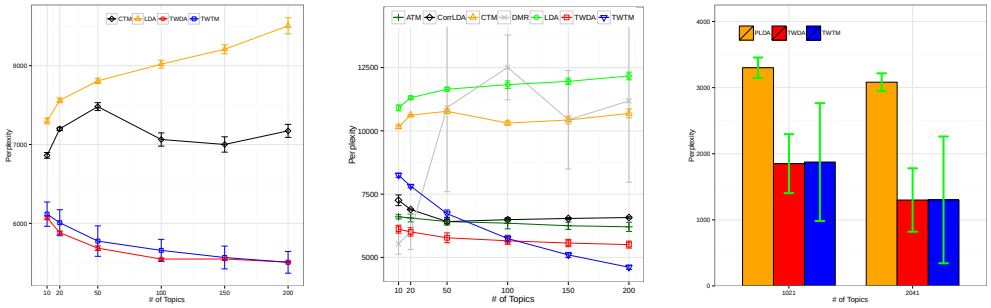
It is worth note that all the solutions need to iterate the MapReduce procedure in driver function until convergence or maximum number of iterations is reached. In Section 7, we show the experimental results about the comparisons of the three solutions on document modeling and efficiency.

7 Experimental Analysis

7.1 Experiment Settings

In the experiments of this work, we used three semi-structured corpora. The first document collection is the data from Internet Movie Database (IMDB). The data set includes 12,091 movie storylines, 52,274 words after removing stop words, and 3,654 tags. These movies belong to 29 genres. And the tags we used contain directors, stars, time, and movie keywords. The second one consists of technical papers of the Digital Bibliography and Library Project (DBLP) data set[3], which is a collection of bibliographic information on major computer science journals and proceedings. In this paper, we use a subset of DBLP that contains abstracts of D=27,435 papers, with W=70,062 words in the vocabulary and L=6,256 unique tags. The tags we used in DBLP include authors and

3. http://www.informatik.uni-trier.de/~ley/db/



(a) TWTM, TWDA, LDA and CTM (b) TWTM, TWDA, DMR, ATM, CorrLDA, LDA and CTM (c) TWTM, TWDA and PLDA

Fig. 5. Perplexity results of different models on IMDB corpora. LDA and CTM only use the words when training in (a), and add the tags as the word features during the training process in (b).

keywords. The last corpus we used contains about 967,012 Wordpress blog posts[4] from Kaggle[5]. In the corpus, there are 163,504 tags and 2,592,562 words. We used this corpus to test the effectiveness and performance of TWTM over a large scale dataset. We implemented the three distributed methods of TWTM using Hadoop 1.1.1 and ran all experiments on a cluster containing 7 physical nodes; each node has 4 cores and 8 threads, and could be configured to run a maximum of 7 mappers and 7 reducers of tasks. With the configuration, we build different scales distributed environments by setting the maximum of mappers used in each node.

We have released the codes on GitHub[6] including TWTM, TWDA and the three distributed solutions using the Hadoop platform.

7.2 Results on Documents Modeling

In order to evaluate the generalization capability of the model, we use the perplexity score that described in [9]. For a test set of D documents, the perplexity is:

$$perplexity = \exp\left\{-\frac{\sum_d^D \log p(\mathbf{w}_d)}{\sum_d^D N_d}\right\},$$

where a lower perplexity score represents better document modeling performance.

There are two parts of the experiments. First, We trained four latent variable models including LDA [9], CTM [7], TWTM and TWDA, on the corpora of a set of movie documents in IMDB, to compare the generalization performance of the four models. In this part, LDA and CTM trains text data without taking advantage of tag information. We removed the stop words and conducted experiments using 5-fold cross-validation. Figure 5(a) demonstrates the perplexity results on the IMDB data set. Clearly, TWTM and TWDA excel both CTM and LDA significantly and consistently.

Second, in order to compare the performance of TWTM and TWDA with other topic models which take advantage of the tag information, we trained TWTM, TWDA, DMR[7], PLDA[8], Author Topic Model (ATM) [31], CorrLDA[6], CTM, and LDA on the set of movie documents in IMDB and computed the perplexity on test data set. Since CTM and LDA could not handle corpus with tags easily, in this experiment, we treated the given tags as word features for them. In CorrLDA, we used the tags in each document to represent the image segments, so that the CorrLDA can handle the SSDs. Figure 5(b) demonstrates the perplexity results of the six models on the IMDB data. The experiment results shows that TWTM and TWDA are better than the other models, and when T increases, CorrLDA, DMR, CTM and LDA are running into over-fitting, while the trend of TWTM and TWDA keeps going down and the perplexity is significantly lower than those of the baselines.

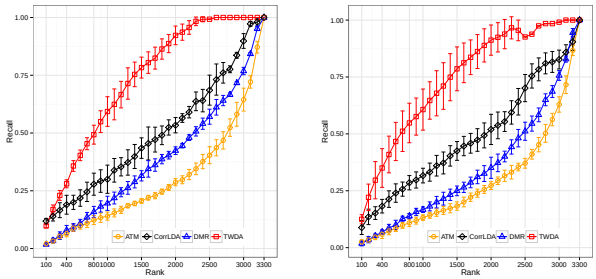
As PLDA [30] assumes that one of tags may optionally denote as a tag "latent" present on every document d, thus, we trained PLDA, TWTM and TWDA over 1021 and 2041 topics on IMDB data set with 1020 tags, since in PLDA, each latent topic takes part in exactly one tag in a collection. As shown in [30], PLDA builds on Labeled LDA [29], and when it set one latent topic and one topic for each tag, it is approximately equivalent to Labeled LDA. For this case, we trained PLDA over 1021 topics. Figure 5(c) shows the perplexity results of TWTM, TWDA and PLDA. Note that TWDA has less mean squared error (MSE) than TWTM. As the results of Figure 5 shown, TWTM and TWDA both work well compared with the other topic models which make use of tag information.

7.3 Results on Tags prediction

In this section we use TWDA to demonstrated the performance of our works on the tags prediction by

4. http://wordpress.com
5. http://www.kaggle.com/c/predict-wordpress-likes/data
6. https://github.com/Shuangyinli

7. We used the Mallet code (http://mallet.cs.umass.edu/).
8. We used the code in Stanford Topic Modeling Toolbox (http://www-nlp.stanford.edu/software/tmt/tmt-0.4/).



(a) K=100 (b) K=200

Fig. 6. Prediction results of TWDA, DMR and ATM for authors on DBLP corpora. We set the number of topic in the corpora to be 100 in (a) and 200 in (b).

process the paper collection in DBLP. In addition to predicting the tags given a document, we evaluate the ability of the proposed model, compared with ATM, DMR and CorrLDA, to predict the tags of the document conditioned on words in the document. In this part, we treat the authors of each paper as the tags, and the abstract as the word features, and we predict the authors of one paper by modeling the paper abstract document data using ATM, DMR, CorrLDA, and TWDA. For each model, we evaluate the likelihood of the authors given the word features in a document, and rank each possible author by the likelihood function of the author. First, for each model, we can get the topic distribution over a test document d_{test} given one author a. Then, we evaluate the $p(d_{test}|a)$ for d_{test} over each author a in the tags(authors) set by

$$p(d_{test}|a) = \prod_{i}^{N}(\sum_{z} p(z|a)p(w_i|z)).$$

For CorrLDA, we let authors represent image regions, and used $p(d_{test}|region)$ shown in [6] to evaluate the likelihood of a author given a document. For DMR and ATM, the method which define $p(d_{test}|a)$ is shown as [26]. Note that the likelihoods for a given author over a document are not necessarily comparable among the topic models, however, what we are interested in is the ranking as same as [26].

We trained the three models on DBLP data set using 5-fold cross-validation and shows the recall when the topic in the corpora is set to be 100 and 200. Results are shown in Figure 6(a) and Figure 6(b). TWDA ranks authors consistently higher than the other models.

7.4 Results on Feature Construction for Classification

The next experiment is to test the classification performance utilizing feature sets generated by TWDA and other baselines. For the base classifier, we use LIBSVM [13] with Gaussian kernel and the default parameters. For the purpose of comparison, we trained four SVMs on tf-idf word features, features induced by a 30-topic LDA model and tf-idf word features, features generated by a TWDA model with the same number of topics, and features induced by a 30-topic TWDA model and tf-idf word features respectively.

TABLE 1
Classification results of different features on F1-score

F1-score	@1	@3	@5
TFIDF	0.5	0.41	0.39
LDA+TFIDF	0.5	0.42	0.39
TWDA	0.57	0.5	0.47
TWDA+TFIDF	0.58	0.5	0.47

In these experiments, we conducted multi-class classification experiments using the IMDB data set, which contains 29 genres. We calculated the evaluation metrics @1, @3 and @5 with the provided class tags of movies' genres, using 5-fold cross-validation. We report the movie classification performance of the different methods in Figure 7, where we see that there is significant improvement in classification performance when using LDA and TWDA comparing with only using tf-idf features, and TWDA outperforms both LDA and tf-idf in terms of @1, @3 and @5.

In order to show the classification performance better, we also calculated the evaluation metrics F-Measure (F1-score). The results of F-Measure is reported in Table 1. TWDA provides substantially better performance on F-Measure.

7.5 Results on Model Robustness

We demonstrated the performance of our work on model robustness in this part of experimental analysis. In this part, we measured and compared the perplexity when we added noise tags information to the test documents using DBLP data set. Respectively, we randomly added 20%, 40%, 50%, 80% and 100% noise tags into a test document and then calculated the perplexity. For example, if a paper document in DBLP has five authors, adding 20% noise is that we randomly selected one author from the author set of

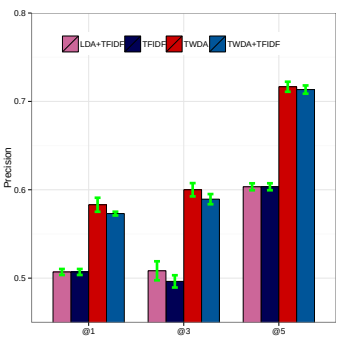


Fig. 7. Classification results of different features on @1, @3 and @5 with 5-fold cross-validation.

the DBLP corpora and added into the paper as a noise author.

In some real-world applications, the noise tags appeared in a document may have some relevance to the real tags. So in this experiment, we selected the noise tags from the author-tag set to meet the real applications to some extent. In this experiment, the DBLP corpora contains more than 6,000 tags, the noise tags we added into a test document would be very sparse for the whole tag set in the corpora. So, we added the different percentages noise tags into the test document to show the trend of perplexity as the noise content increases. Figure 8 shows that both TWDA and ATM have a more steady trend as the noise level increases, compared with DMR. Table 2 shows some examples about the weights between the original tags and noise tags. The red tags are the noise added into the test data, and the values behind are the weights among the tags we inference from the TWDA model. Note that, we showed the weight values after normalized. As the results shown, TWDA has a good performance on model robustness, for the weight values of the noise tags are much smaller than the other original tags. In some applications, we can use the proposed model to rank the tags given in a document, which would be a good approach to tag recommendation and annotation.

7.6 Results on Large-scale Datasets

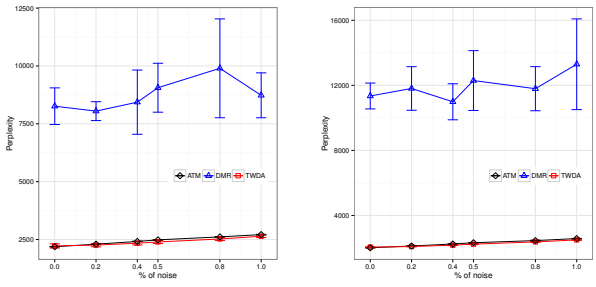
We demonstrated the performance of the three proposed parallelized solutions of TWTM for a large-scale dataset from training time and accuracy on document modeling, which are suitable for TWDA as well.

Firstly, we measured and compared the training time of Solutions I, II and III using the Wordpress blog data set with the same system setting and model parameters. We used a doc-indexed sparse storage mode for the matrix of ξ-document, for the matrix would be very huge over a large scale data set. Figure 9(a) shows the performance on the average training time per iteration of the three solutions compared with the standard TWTM as the baseline, when we set the number of topic K = 10, 20 and 50 respectively.

Secondly, We sampled the training dataset from the Wordpress corpus with different sample ratios, 0.1, 0.3, 0.6, 0.8 and 1.0, to show the performance of running time by different size of training dataset. In addition, we limited the maximum number of Mappers in the configuration when we trained the

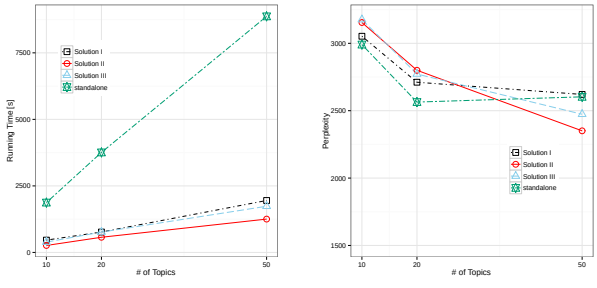
TABLE 2
Some examples of the normalized weights among the original tags and noise tags. The noise tags are in red, and the numbers are the corresponding weight values.

"Bug isolation via remote program sampling [24]"
Ben Liblit: 0.185 Alex Aiken: 0.2257
aAlice X. Zheng: 0.228 Michael I. Jordan: 0.349
K. G. Shin: 0.01
"Web question answering: is more always better? [18]"
Susan Dumais: 0.986 Michele Banko: 0.0032
Eric Brill: 0.0038 Jimmy Lin: 0.0038
Andrew Ng: 0.0024
R. Katz: 0.00018
"Contextual search and name disambiguation in email using graphs [27]"
Einat Minkov: 0.425 William W. Cohen: 0.342
Andrew Y. Ng: 0.128
J. Ma: 0.033 D. Ferguson: 0.07
"A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes [22]"
Michael Kearns: 0.296 Yishay Mansour:0.166
Andrew Y. Ng: 0.31
J. Blythe: 0.089 B. Adida: 0.027
P. J. Modi: 0.1
"The nested Chinese restaurant process and bayesian nonparametric inference of topic [5]"
David M. Blei:0.46 Thomas L. Griffiths:0.186
Michael I. Jordan:0.225
B. Clifford:0.031 R. Szeliski:0.048
X. Wang:0.05



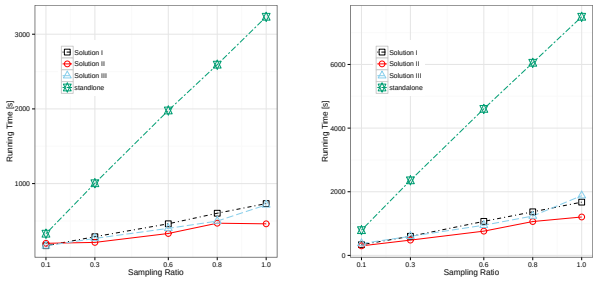
(a) K=100 (b) K=200

Fig. 8. The Results of adding noise to different models(ATM, DMR and TWDA). (a) set K=100, and (b) set K=200. Steady trending means a good performance on model robustness.



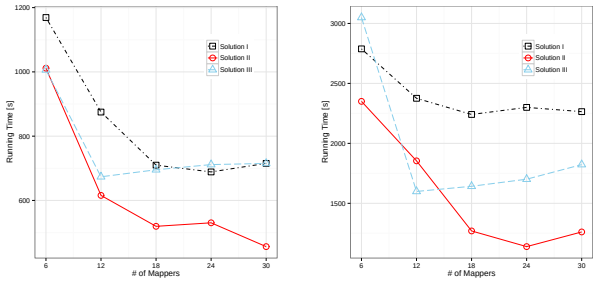
(a) (b)

Fig. 9. (a) The average training time per iteration for Solution I, II, III with different number of topics compared with the standard TWTM. (b) The perplexity results for Solution I, II, III, and the standard TWTM.



(a) K = 20 (b) K = 50

Fig. 10. The average training time per iteration on the Wordpress corpus with different number of sampling radios for Solution I, II, III.



(a) K = 20 (b) K = 50

Fig. 11. The average training time per iteration on the Wordpress corpus with different number of Mappers for Solution I, II, III. Note that the horizontal axis repesents the maximum number of Mappers used in a training task.

model as described in Section 7.1, to demonstrate the comparison performance of the three solutions under the restricted resources. Figure 10 and Figure 11 show the results about the average training time per iteration of the three solutions using different sample ratios and Mappers of dataset when training, by setting the number of topic K = 10, 20 and 50 respectively. From this part of experiments, we find that Solution II has a better performance of efficiency than Solution I and III.

Meanwhile, in order to compare with other model, such as PLDA, we used the Wordpress dataset with 1,000 tags to train a PLDA model with $K_l = 1$ (we used the code from Stanford Topic Modeling Toolbox). We trained TWTM by Solution II with $K = 5$. Table 3 shows the comparison of PLDA and TWTM by Solution II.

TABLE 3
The average training time (second) per iteration for Solution II and PLDA

Sampling radio	0.1	0.3	0.6	0.8	1.0
PLDA	66.6	114.8	193.4	250.6	276.4
Solution II	77.6	88.6	104.8	116.2	120.8

As described in Section 6, in Solution I, it would spend a great deal of time on data migration to update π in Driver process, and in Solution III, a lot of resources are taken on the clustering process in each Mapper, especially when the corpus is non-homogeneous which leads to uneven loading of each Mapper. While, Solution II avoids these problems by a approximation method.

Lastly, we measured the generalization capability of the three solutions using the perplexity and conducted experiments. We held out 20% of the data for test and trained the three solutions on the remaining 80%. We observe that there is relatively little difference among the solutions compared with the standard TWTM in terms of perplexity as shown in Figure 9(b) when the number of topic increases. That is, all the three solutions are good approximations in terms of model fitness. It is worthy to note that Solution II has almost the same performance as Solution I and Solution III.

8 Conclusion

With the tag-weighted topic model proposed in the paper, we provide and analyze a probabilistic approach for mining semi-structured documents. Meanwhile, three distributed solutions for TWTM are presented to handle the large scale problems. Besides, TWTM is able to obtain the topics distribution of tags in the corpus, which is very useful for text classification, clustering and other data mining applications. At the same time, we propose a novel framework of processing the tagged text with a high extensibility, and uses a novel function of tag-weighted topic assignment of documents. As an extended model, TWDA shows the capability on handling the mixture corpora of semi-structured documents and unstructured documents. The second benefit of the tag-weighted topic model is that it allows one to incorporate different types of tags in modeling documents, and provides a general framework for multi-tag modeling at not only the level of tags but also the level of documents. It helps provide a different approach in classification, clustering, recommendation, and so on. For large scale semi-structured documents, the proposed solutions are shown to be effective and efficient for some complex web applications. In the future, we plan to apply TWTM to different practical areas (e.g., image classification and annotation, video retrieval).

References

[1] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.
[2] Arthur U. Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *UAI*, pages 27–34, 2009.
[3] Hagai Attias. A variational baysian framework for graphical models. In *NIPS*, pages 209–215, 1999.
[4] Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern Recognition and Machine Learning*. *J. Electronic Imaging*, 16(4):049901, 2007.

[5] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57(2):7:1–7:30, February 2010.

[6] David M. Blei, Michael I, David M. Blei, and Michael I. Modeling annotated data. In *In Proc. of the 26th Intl. ACM SIGIR Conference*, 2003.

[7] David M. Blei and John D. Lafferty. Correlated topic models. In *NIPS*, 2005.

[8] David M. Blei and Jon D. McAuliffe. Supervised topic models. In *NIPS*, 2007.

[9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[10] Jordan L. Boyd-Graber and David M. Blei. Syntactic topic models. *CoRR*, abs/1002.4665, 2010.

[11] Andrej Bratko and Bogdan Filipic. Exploiting structural information for semi-structured document categorization. *Information Processing and Management*, 42(3):679 – 694, 2006.

[12] Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. Modeling hidden topics on document manifold. In *CIKM*, pages 911–920, 2008.

[13] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM TIST*, 2(3):27, 2011.

[14] Jonathan Chang and David M. Blei. Relational topic models for document networks. *Journal of Machine Learning Research - Proceedings Track*, 5:81–88, 2009.

[15] Xu Chen, Mingyuan Zhou, and Lawrence Carin. The contextual focused topic model. In *KDD*, pages 96–104, 2012.

[16] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[17] Hongbo Deng, Jiawei Han, Bo Zhao, Yintao Yu, and Cindy Xide Lin. Probabilistic topic models with biased propagation on heterogeneous information networks. In *KDD*, pages 1271–1279, 2011.

[18] Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. Web question answering: is more always better? In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 291–298, New York, NY, USA, 2002. ACM.

[19] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. In *PNAS*, pages 449–455, 2004.

[20] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[21] Tomoharu Iwata, Takeshi Yamada, and Naonori Ueda. Modeling social annotation data with content relevance using a topic model. In *NIPS*, pages 835–843, 2009.

[22] Michael Kearns, Yishay Mansour, and Andrew Y. Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Mach. Learn.*, 49(2-3):193–208, November 2002.

[23] Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In *NIPS*, pages 897–904, 2008.

[24] Ben Liblit, Alex Aiken, Alice X. Zheng, and Michael I. Jordan. Bug isolation via remote program sampling. *SIGPLAN Not.*, 38(5):141–154, May 2003.

[25] Pierre-Francois Marteau, Gildas Ménier, and Eugen Popovici. Weighted naive bayes model for semi-structured document categorization. *CoRR*, abs/0901.0358, 2009.

[26] David M. Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*, pages 411–418, 2008.

[27] Einat Minkov, William W. Cohen, and Andrew Y. Ng. Contextual search and name disambiguation in email using graphs. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 27–34, New York, NY, USA, 2006. ACM.

[28] James Petterson, Alexander J. Smola, Tibério S. Caetano, Wray L. Buntine, and Shravan Narayanamurthy. Word features for latent dirichlet allocation. In *NIPS*, pages 1921–1929, 2010.

[29] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pages 248–256, 2009.

[30] Daniel Ramage, Christopher D. Manning, and Susan Dumais. Partially labeled topic models for interpretable text mining. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 457–465, New York, NY, USA, 2011. ACM.

[31] Michal Rosen-Zvi, Thomas L. Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *UAI*, pages 487–494, 2004.

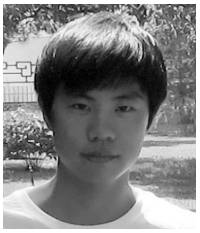
[32] Issei Sato and Hiroshi Nakagawa. Rethinking collapsed variational bayes inference for lda. In *ICML*, 2012.

[33] Markus Tresch, Neal Palmer, and Allen Luniewski. Type classification of semi-structured documents. In *VLDB*, pages 263–274, 1995.

[34] Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM.

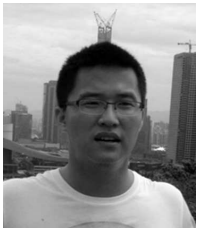
[35] Jeonghee Yi and Neel Sundaresan. A classifier for semi-structured documents. In *KDD*, pages 340–344, 2000.

[36] Jun Zhu, Amr Ahmed, and Eric P. Xing. Medlda: maximum margin supervised topic models for regression and classification. In *ICML*, page 158, 2009.



Shuangyin Li received the Master degree in School of Information Science and Technology, Sun Yat-sen University, China, in 2011. During the Master's program, he focused on the research of large scale image retrieval system on Hadoop platform. Currently, he is active within the field of Text Mining and Artificial Intelligence, and continues his research in a PhD track at Sun Yat-sen University. His PhD research focuses on Topic Model and Deep Neural Networks, and he has published several research mainly focused on the semi-structured documents modeling.

Jiefei Li received the Bachelor's degree in Department of Computer Science, Sun Yat-sen University, in 2011. Currently, he is studying for a master's degree in Sun Yat-sen University. His research focuses on Topic Model.



Guan Huang received the Bachelor's degree in Department of Computer Science, Sun Yat-sen University, in 2009. Currently, he is studying for a master 's degree in Sun Yat-sen University in the filed of word embedding and topic model, learning to rank.

Ruiyang Tan is studying for a Bachelor's degree in Department of Computer Science, Sun Yat-sen University. He has participated in ACM/ICPC twice times and won two Asia regional champions.

Rong Pan received the BSc and PhD degrees in applied mathematics from Sun Yat-sen University, China, in 1999 and 2004, respectively. He was a postdoctoral fellow at the Hong Kong University of Science and Technology (2005 2007) and HP Labs (2007 2009). Since then, he has been a faculty member of Department of Computer Science in Sun Yat-sen University. His research interest includes text mining, recommender systems, data mining, and machine learning.

APPENDIX A
TAG-WEIGHTED TOPIC MODEL

In the topic models, the key inferential problem that we need to solve is to compute the posterior distribution of the hidden variables given a document d. Given the document d, we can easily get the posterior distribution of the latent variables in the proposed model, as:

$$p(\varepsilon^d, \mathbf{z}|\mathbf{w}^d, T^d, \theta, \eta, \psi, \pi) = \frac{p(\varepsilon^d, \mathbf{z}, \mathbf{w}^d, T^d|\theta, \eta, \psi, \pi)}{p(\mathbf{w}^d, T^d|\theta, \eta, \psi, \pi)}.$$

Integrating over ε and summing out z, we easily obtain the marginal distribution of d:

$$p(\mathbf{w}^d, T^d|\eta, \theta, \psi, \pi) = p(\mathbf{t}^d|\eta) \int p\left(\varepsilon^d|(T^d \times \pi)\right) \cdot \prod_{i=1}^{N} \sum_{z_i^d=1}^{K} p(z_i^d|(\varepsilon^d)^{\mathrm{T}} \times T^d \times \theta) p(w_i^d|z_i^d, \psi_{1:K}) \; d\varepsilon^d.$$

In this work, we make use of mean-field variational EM algorithm to efficiently obtain an approximation of this posterior distribution of the latent variables. In the mean-field variational inference, we minimize the KL divergence between the variational posterior probability and the true posterior probability through by maximizing the evidence lower bound (ELBO) $\mathcal{L}(\cdot)$. For a single document d, we obtain the $\mathcal{L}(\cdot)$ using Jensen's inequality:

$$\begin{aligned}\mathcal{L}(\xi_{1:l^d}, \gamma_{1:K}; \eta_{1:L}, \pi_{1:L}, \theta_{1:L}, \psi_{1:K}) &= E[\log p(T_{1:l^d}|\eta_{1:L})] + E[\log p(\varepsilon^d|T^d \times \pi)] \\ &+ \sum_{i=1}^{N} E[\log p(z_i|(\varepsilon^d)^{\mathrm{T}} \times T^d \times \theta)] + \sum_{i=1}^{N} E[\log p(w_i|z_i, \psi_{1:K})] + H(q),\end{aligned}$$

where ξ is a l^d-dimensional Dirichlet parameter vector and γ is $1 \times K$ vector, both of which are variational parameters of variational distribution. $H(q)$ indicates the entropy of the variational distribution:

$$H(q) = -E[\log q(\varepsilon^d)] - E[\log q(z)].$$

Here the exception is taken with respect to a variational distribution $q(\varepsilon^d, z_{1:N})$, and we choose the following fully factorized distribution:

$$q(\varepsilon^d, z_{1:N}|\xi_{1:L}, \gamma_{1:K}) = q(\varepsilon^d|\xi) \prod_{i=1}^{N} q(z_i|\gamma_i),$$

where the dimension of parameter ξ is changed with different documents.

In the $\mathcal{L}(\cdot)$,

$$E[\log p(z_i|(\varepsilon^d)^{\mathrm{T}} \times T^d \times \theta)] = \sum_{k=1}^{K} \gamma_{ik} E[\log((\varepsilon^d)^{\mathrm{T}} \times T^d \times \theta)_k].$$

To preserve the lower bound on the log probability, we upper bound the log normalizer in $E[\log((\varepsilon^d)^{\mathrm{T}} \times T^d \times \theta)_k]$ using Jensen's inequality again:

$$E[\log((\varepsilon^d)^{\mathrm{T}} \times T^d \times \theta)_k] = E[\log \sum_{i=1}^{l^d} \varepsilon_i^d \theta_k^{(i)}] \geq E[\sum_{i=1}^{l^d} \varepsilon_i^d \log \theta_k^{(i)}] = \sum_{i=1}^{l^d} \log \theta_k^{(i)} E[\varepsilon_i^d],$$

where the expression of $\theta^{(i)}$, $i \in \{1, \cdots, l^d\}$, means the i-th tag's topic assignment vector, corresponding to the i-th row of Θ^d. Note that the expectation of Dirichlet random variable is $E[\varepsilon_i^d] = \frac{\xi_i}{\sum_{j=1}^{l^d} \xi_j}$.

Thus, for the document d,

$$\sum_{i=1}^{N} E[\log p(z_i|(\varepsilon^d)^{\mathrm{T}} \times T^d \times \theta)] = \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \cdot \sum_{j=1}^{l^d} \log \theta_k^{(j)} \frac{\xi_j}{\sum_{j'=1}^{l^d} \xi_{j'}}.$$

Finally, we expand $\mathcal{L}(\cdot)$ in terms of the model parameters $(\eta, \pi, \theta, \psi)$ and the variational parameters (ξ, γ) as follows:

$$\begin{aligned}\mathcal{L}(\xi,\gamma;\eta,\pi,\theta,\psi) = &\sum_{l=1}^{L}(t_l^d \log \eta_l^d + (1-t_l^d)\log(1-\eta_l^d)) \\ &+ \log\Gamma(\sum_{i=1}^{l^d}(T^d\times\pi)_i) - \sum_{i=1}^{l^d}\log\Gamma\left((T^d\times\pi)_i\right) + \sum_{i=1}^{l^d}((T^d\times\pi)_i - 1)\left(\Psi(\xi_i) - \Psi(\sum_{j=1}^{l^d}\xi_j)\right) \\ &+ \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\sum_{j=1}^{l^d}\log\theta_k^{(j)}\frac{\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}} + \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{V}\gamma_{ik}(w^d)_i^j\log\psi_{kj} \\ &- \log\Gamma(\sum_{i=1}^{l^d}\xi_i) + \sum_{i=1}^{l^d}\log\Gamma(\xi_i) - \sum_{i=1}^{l^d}(\xi_i-1)\left(\Psi(\xi_i) - \Psi(\sum_{j'=1}^{l^d}\xi_{j'})\right) - \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}^d\log\gamma_{ik}^d.\end{aligned}$$

Then, we maximize the lower bound $\mathcal{L}(\xi, \gamma; \eta, \pi, \theta, \psi)$ with respect to the variational parameters ξ and γ, using a variational expectation-maximization(EM) procedure as follows.

A.1 Variational E-step

A.1.1 ξ

We first maximize $\mathcal{L}(\cdot)$ with respect to ξ_i for the document d. Maximize the terms which contain ξ:

$$\begin{aligned}\mathcal{L}_{[\xi]} = &\sum_{i=1}^{l^d}(\sum_{l'=1}^{L}\pi_{l'}T_{il'}^d - 1)\left(\Psi(\xi_i) - \Psi(\sum_{j'=1}^{l^d}\xi_{j'})\right) + \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\cdot\sum_{j=1}^{l^d}\log\theta_k^{(j)}\frac{\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}} - \log\Gamma(\sum_{i=1}^{l^d}\xi_i) \\ &+ \sum_{i=1}^{l^d}\log\Gamma(\xi_i) - \sum_{i=1}^{l^d}(\xi_i-1)\left(\Psi(\xi_i) - \Psi(\sum_{j'=1}^{l^d}\xi_{j'})\right),\end{aligned}$$

where $\Psi(\cdot)$ denotes the digamma function, the first derivative of the log of the Gamma function, and $(T^d \times \pi)_i$ = $\sum_{i=1}^{l^d}\sum_{l=1}^{L}\pi_l T_{il}^d$. The derivative of $\mathcal{L}_{[\xi]}$ with respect to ξ_i is

$$\mathcal{L}^{'}(\xi_i) = \Psi^{'}(\xi_i)(\sum_{l=1}^{L}\pi_l T_{il}^d - \xi_i) - \Psi^{'}(\sum_{j=1}^{l^d}\xi_j)\cdot\sum_{i=1}^{l^d}(\sum_{l=1}^{L}\pi_l T_{il}^d - \xi_i) + \sum_{i'=1}^{N}\sum_{k=1}^{K}\gamma_{i'k}^d\cdot\left(\frac{\log\theta_k^{(i)}(\sum_{j=1}^{l^d}\xi_j) - \sum_{j=1}^{l^d}\log\theta_k^{(j)}\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}^{\ 2}}\right).$$

Here we use gradient descent method to find the ξ to make the maximization of $\mathcal{L}_{[\xi]}$.

A.1.2 γ

Next, we maximize $\mathcal{L}(\cdot)$ with respect to γ_{ik}. Adding the Lagrange multipliers to the terms which contain γ_{ik}, we get the following equation:

$$\mathcal{L}_{[\gamma]} = \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\sum_{j=1}^{l^d}\log\theta_k^{(j)}\frac{\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}} + \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{V}\gamma_{ik}(w^d)_i^j\log\psi_{kj} - \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}^d\log\gamma_{ik}^d + \sum_{i=1}^{N}\lambda_i(\sum_{k=1}^{K}\gamma_{ik} - 1).$$

By taking the derivative with respect to γ_{ik}, and setting the derivative to zero yields, we obtain the update equation of γ_{ik}:

$$\gamma_{ik} \propto \psi_{k,v^{w_i}}\exp\{\sum_{j=1}^{l^d}\log\theta_k^{(j)}\frac{\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}}\},$$

where v^{w_i} denotes the index of w_i in the dictionary.

A.2 M-step estimation

The M-step needs to update four parameters: η, the tagging prior probability, π, the Dirichlet prior of the tags' weights, θ, the topic distribution over all tags in the corpus, and ψ, the probability of a word under a topic.

A.2.1 η

For a given corpus, the η_i is estimated by adding up the number of i^{th} label which appears in all documents. It does not depend any parameter in the proposed model, except itself. By maximizing the terms which contain η, we have

$$\eta_l = \frac{\sum_d^D t_l^d}{D},$$

where D is the size of corpus. Because each document's tags-set is observed, the Bernoulli prior η is unused, which is included for model completeness.

A.2.2 π

For the document d, the terms that involve the Dirichlet prior π:

$$\mathcal{L}_{[\pi]} = \log\Gamma(\sum_{i=1}^{l^d}(T^d \times \pi)_i) - \sum_{i=1}^{l^d}\log\Gamma\left((T^d \times \pi)_i\right) + \sum_{i=1}^{l^d}\left((T^d \times \pi)_i - 1\right)\left(\Psi(\xi_i) - \Psi(\sum_{j=1}^{l^d}\xi_j)\right).$$

We use gradient descent method by taking derivative of $\mathcal{L}_{[\pi]}$ with respect to π_l on the whole corpus to find the estimation of π. Taking derivatives with respect to π_l on the corpus, we obtain:

$$\mathcal{L}'_{[\pi_l]} = \sum_{d=1}^{D}\Psi(\sum_{i=1}^{l^d}\sum_{l'=1}^{L}\pi_{l'}\cdot T_{il'}^d)\cdot\sum_{i=1}^{l^d}T_{il}^d - \sum_{d=1}^{D}\sum_{i=1}^{l^d}\Psi(\sum_{l'=1}^{L}\pi_{l'}\cdot T_{il'}^d)\cdot T_{il}^d + \sum_{d=1}^{D}\sum_{i=1}^{l^d}\left(\Psi(\xi_i) - \Psi(\sum_{j=1}^{l^d})\right)\cdot T_{il}^d.$$

A.2.3 θ

The only term that involves θ is:

$$\mathcal{L}_{[\theta]} = \sum_{d=1}^{D}\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\sum_{j=1}^{l^d}\log\theta_k^{(j)}\frac{\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}},$$

where ξ_j, $j \in \{1, \cdots, l^d\}$ in the document d needs to be extended to $t_l^d \cdot \xi_l^d$, $l \in \{1, \cdots, L\}$ for convenient to simplify $\mathcal{L}_{[\theta]}$. With the Lagrangian of the $\mathcal{L}_{[\theta]}$, which incorporate the constraint that the K-components of θ_l sum to one, adding $\sum_{l=1}^{L}\lambda_l(\sum_{k=1}^{K}\theta_{lk} - 1)$ to $\mathcal{L}_{[\theta]}$, taking the derivative with respect to θ_{lk}, and setting the derivative to zero yields, we obtain the estimation of θ over the whole corpus,

$$\theta_{lk} \propto \sum_{d=1}^{D}\sum_{i=1}^{N}\gamma_{ik}^d\frac{\xi_l^d t_l^d}{\sum_{l=1}^{L}(\xi_l^d t_l^d)}.$$

A.2.4 ψ

To maximize with respect to ψ, we isolate corresponding terms and add Lagrange multipliers:

$$\mathcal{L}_{[\psi]} = \sum_{d=1}^{D}\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{V}\gamma_{ik}(w^d)_i^j\log\psi_{kj} + \sum_{k=1}^{K}\lambda_k(\sum_{j=1}^{v}\psi_{kj} - 1).$$

Take the derivative with respect to ψ_{kj}, and set it to zero, we get:

$$\psi_{kj} \propto \sum_{d=1}^{D}\sum_{i=1}^{N}\gamma_{ik}^d(w^d)_i^j.$$

APPENDIX B
TAG-WEIGHTED DIRICHLET ALLOCATION

In TWDA, we treat π, μ, η, θ and ψ as unknown constants to be estimated, and use a variational expectation-maximization (EM) procedure to carry out approximate maximum likelihood estimation as TWTM. Given the document d, we can easily get the posterior distribution of the latent variables in the TWDA model, as:

$$p(\varepsilon^d, \lambda^d, \mathbf{z}|\mathbf{w}^d, T^d, \theta, \eta, \psi, \pi, \mu) = \frac{p(\varepsilon^d, \lambda^d, \mathbf{z}, \mathbf{w}^d, T^d|\theta, \eta, \psi, \pi, \mu)}{p(\mathbf{w}^d, T^d|\theta, \eta, \psi, \pi, \mu)}.$$

As with TWTM, it is not efficiently computable. We maximize the evidence lower bound(ELBO) $\mathcal{L}(\cdot)$using Jensen's inequality, and for a document d we have the form:

$$\begin{aligned}\mathcal{L}(\xi_{1:l^d}, \gamma_{1:K}, \rho_{1:K}; \eta_{1:L}, \pi_{1:L}, \mu_{1:K}, \theta_{1:L}, \psi_{1:K}) &= E[\log p(T_{1:l^d}|\eta_{1:L})] + E[\log p(\varepsilon^d|T^d \times \pi)] + E[\log p(\lambda^d|\mu)] \\ &+ \sum_{i=1}^{N} E[\log p(z_i|(\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda}))] + \sum_{i=1}^{N} E[\log p(w_i|z_i, \psi_{1:K})] \\ &+ H(q),\end{aligned}$$

where ξ is a l^d-dimensional Dirichlet parameter vector, ρ is a $1 \times K$ vector and γ is $1 \times K$ vector, all of which are variational parameters of variational distribution. Unlike the TWTM, l^din TWDA is one more than the number of the observed tags in the document d. $H(q)$ indicates the entropy of the variational distribution:

$$H(q) = -E[\log q(\varepsilon^d)] - E[\log q(\lambda)] - E[\log q(z)].$$

Here the exception is taken with respect to a variational distribution $q(\varepsilon^d, q(\lambda^d), z_{1:N})$, and we choose the following fully factorized distribution:

$$q(\varepsilon^d, \lambda^d, z_{1:N}|\xi_{1:L}, \rho_{1:K}, \gamma_{1:K}) = q(\varepsilon^d|\xi) q(\lambda^d|\rho) \prod_{i=1}^{N} q(z_i|\gamma_i).$$

The term of the expected log probability of the topic assignment:

$$E[\log p(z_i|(\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda}))] = \sum_{k=1}^{K} \gamma_{ik} E[\log((\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda}))_k],$$

which could be difficult to compute, because of tag-weighted topic assignment which is used in TWDA. Thus we use Jensen's inequality:

$$\begin{aligned}E[\log((\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda}))_k] &= E[\log(\sum_{i=1}^{l^d-1} \varepsilon_i^d \theta_k^{(i)} + \varepsilon_{l^d}^d \lambda_k)] \\ &\geq E[\sum_{i=1}^{l^d-1} \varepsilon_i^d \log \theta_k^{(i)} + \varepsilon_{l^d}^d \cdot \log \lambda_k] \\ &= \sum_{i=1}^{l^d-1} \log \theta_k^{(i)} E[\varepsilon_i^d] + E[\varepsilon_{l^d}^d \cdot \log \lambda_k],\end{aligned}$$

where the expression of $\theta^{(i)}$, $i \in \{1, \cdots, l^d - 1\}$, means the i-th tag's topic assignment vector, corresponding to the i-th row of Θ^d.

because the variational distribution is fully factorized, so we can get:

$$E[\log((\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda}))_k] = \sum_{i=1}^{l^d-1} \log \theta_k^{(i)} E[\varepsilon_i^d] + E[\varepsilon_{l^d}^d] \cdot E[\log \lambda_k],$$

where

$$E[\varepsilon_{l^d}^d] = \xi_{l^d} / \sum_{j=1}^{l^d} \xi_j,$$

$$E[\log \lambda_k] = \Psi(\rho_k) - \Psi(\sum_{j'=1}^{K} \rho_{j'}).$$

With $E[\varepsilon_i^d] = \frac{\xi_i}{\sum_{j=1}^{l^d}\xi_j}$, Thus, for the document d,

$$\sum_{i=1}^{N} E[\log p(z_i|(\varepsilon^d)^{\mathrm{T}} \times T^d \times (\frac{\theta}{\lambda}))] = \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\cdot[\sum_{j=1}^{l^d-1}\log\theta_k^{(j)}\frac{\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}} + (\Psi(\rho_k) - \Psi(\sum_{j'=1}^{K}\rho_{j'}))\frac{\xi_{l^d}}{\sum_{j=1}^{l^d}\xi_j}].$$

Then we expand the $\mathcal{L}(\cdot)$ of TWDA as follows:

$$\begin{aligned}
\mathcal{L}(\xi,\gamma,\rho;\eta,\pi,\mu,\theta,\psi) = &\sum_{l=1}^{L}(t_l^d\log\eta_l^d + (1-t_l^d)\log(1-\eta_l^d)) \\
&+ \log\Gamma(\sum_{i=1}^{l^d}(T^d\times\pi)_i) - \sum_{i=1}^{l^d}\log\Gamma((T^d\times\pi)_i) + \sum_{i=1}^{l^d}((T^d\times\pi)_i - 1)\left(\Psi(\xi_i) - \Psi(\sum_{j=1}^{l^d}\xi_j)\right) \\
&+ \log\Gamma(\sum_{j=1}^{K}\mu_j) - \sum_{i=1}^{K}\log\Gamma(\mu_i) + \sum_{i=1}^{K}(\mu_i-1)\left(\Psi(\rho_i^d) - \Psi(\sum_{j=1}^{K}\rho_j^d)\right) \\
&+ \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\cdot\sum_{j=1}^{l^d}C_k^{(j)}\frac{\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}} + \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{V}\gamma_{ik}(w^d)_i^j\log\psi_{kj} \\
&- \log\Gamma(\sum_{i=1}^{l^d}\xi_i) + \sum_{i=1}^{l^d}\log\Gamma(\xi_i) - \sum_{i=1}^{l^d}(\xi_i-1)\left(\Psi(\xi_i) - \Psi(\sum_{j'=1}^{l^d}\xi_{j'})\right) \\
&- \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\cdot\log\gamma_{ik} \\
&- \log\Gamma(\sum_{j=1}^{K}\rho_j) + \sum_{i=1}^{K}\log\Gamma(\rho_i) - \sum_{i=1}^{K}(\rho_i-1)\left(\Psi(\rho_i) - \Psi(\sum_{j=1}^{K}\rho_j)\right).
\end{aligned}$$

where

$$C_k^{(j)} = \begin{cases}\log\theta_k^{(j)} & j\in\{1,\cdots,l^d-1\} \\ \Psi(\rho_k) - \Psi(\sum_{j'=1}^{K}\rho_{j'}) & j = l^d\end{cases},$$

and

$$(T^d\times\pi)_i = \sum_{l=1}^{L+1}\pi_l T_{il}^d.$$

B.1 Variational E-step

For a single document d, the variational parameters include ξ^d, ρ^d and γ_{ik}. First, we maximize $\mathcal{L}(\cdot)$ with respect to the variational parameters to obtain an estimate of the posterior.

B.1.1 Optimization with respect to ξ

We first maximize $\mathcal{L}(\cdot)$ with respect to ξ_i for the document d. Maximize the terms which contain ξ:

$$\begin{aligned}
\mathcal{L}_{[\xi]} = &\sum_{i=1}^{l^d}(\sum_{l'=1}^{L+1}\pi_{l'}T_{il'}^d - 1)\left(\Psi(\xi_i) - \Psi(\sum_{j'=1}^{l^d}\xi_{j'})\right) + \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\cdot\sum_{j=1}^{l^d}C_k^{(j)}\frac{\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}} - \log\Gamma(\sum_{i=1}^{l^d}\xi_i) \\
&+ \sum_{i=1}^{l^d}\log\Gamma(\xi_i) - \sum_{i=1}^{l^d}(\xi_i-1)\left(\Psi(\xi_i) - \Psi(\sum_{j'=1}^{l^d}\xi_{j'})\right),
\end{aligned}$$

The derivative of $\mathcal{L}_{[\xi]}$ with respect to ξ_i is

$$\mathcal{L}'(\xi_i) = \Psi'(\xi_i)(\sum_{l=1}^{L+1}\pi_l T_{il}^d - \xi_i) - \Psi'(\sum_{j=1}^{l^d}\xi_j)\sum_{i=1}^{l^d}(\sum_{l=1}^{L+1}\pi_l T_{il}^d - \xi_i) + \sum_{i'=1}^{N}\sum_{k=1}^{K}\gamma_{i'k}^d\cdot\left(\frac{C_k^{(i)}(\sum_{j=1}^{l^d}\xi_j) - \sum_{j=1}^{l^d}C_k^{(j)}\xi_j}{(\sum_{j'=1}^{l^d}\xi_{j'})^2}\right).$$

Here we use gradient descent method to find the ξ to make the maximization of $\mathcal{L}_{[\xi]}$.

B.1.2 Optimization with respect to ρ

Next, we maximize $\mathcal{L}(\cdot)$ with respect to ρ. The terms that involve the variational Dirichlet ρ are:

$$\mathcal{L}_{[\rho]} = \sum_{i=1}^{K}(\mu_i - 1)\left(\Psi(\rho_i) - \Psi(\sum_{j=1}^{K}\rho_j)\right) - \log\Gamma(\sum_{j=1}^{K}\rho_j) + \sum_{i=1}^{K}\log\Gamma(\rho_i) - \sum_{i=1}^{K}(\rho_i - 1)\left(\Psi(\rho_i) - \Psi(\sum_{j=1}^{K}\rho_j)\right) + \sum_{k=1}^{K}\sum_{i=1}^{N}\gamma_{ik}\cdot\frac{\xi_{l^d}}{\sum_{j=1}^{l^d}\xi_j}\cdot\left(\Psi(\rho_k) - \Psi(\sum_{j=1}^{K}\rho_j)\right).$$

This simplifies to:

$$\mathcal{L}_{[\rho]} = \sum_{i=1}^{K}\left(\Psi(\rho_i) - \Psi(\sum_{j=1}^{K}\rho_j)\right)\cdot\left(\mu_i - \rho_i + \sum_{n=1}^{N}\gamma_{ni}\cdot\frac{\xi_{l^d}}{\sum_{j=1}^{l^d}\xi_j}\right) - \log\Gamma(\sum_{j=1}^{K}\rho_j) + \sum_{i=1}^{K}\log\Gamma(\rho_i).$$

Taking the derivative with respect to ρ_i and setting it to zero, we obtain a maximum at:

$$\rho_i = \mu_i + \sum_{n=1}^{N}\gamma_{ni}\cdot\frac{\xi_{l^d}}{\sum_{j=1}^{l^d}\xi_j}.$$

B.1.3 Optimization with respect to γ

The terms that contain γ are:

$$\mathcal{L}_{[\gamma]} = \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\sum_{i=1}^{l^d}C_k^{(i)}\cdot\frac{\xi_i}{\sum_{j=1}^{l^d}\xi_j} + \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{V}\gamma_{ik}w_{ij}\log\psi_{k,v^{w_i}} - \sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\cdot\log\gamma_{ik}$$

Adding the Lagrange multipliers to the terms which contain γ_{ik}, taking the derivative with respect to γ_{ik}, and setting the derivative to zero yields, we obtain the update equation of γ_{ik}:

$$\gamma_{ik} \propto \psi_{k,v^{w_i}}\exp\{\sum_{j=1}^{l^d}C_k^{(j)}\frac{\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}}\},$$

where v^{w_i} denotes the index of w_i in the dictionary.

In E-step, we update the ξ, ρ and γ for each document with the initialized model parameters.

B.2 M-step estimation

The M-step needs to update five parameters: η, the tagging prior probability, π, the Dirichlet prior of the tags' weights, θ, the topic distribution over all tags in the corpus, ψ, the probability of a word under a topic, and μ, a Dirichlet prior of model. It is worthy to note that we update η with the same method as in TWTM.

B.2.1 Optimization with respect to π

For the document d, the terms that involve the Dirichlet prior π:

$$\mathcal{L}_{[\pi]} = \log\Gamma(\sum_{i=1}^{l^d}(T^d\times\pi)_i) - \sum_{i=1}^{l^d}\log\Gamma((T^d\times\pi)_i) + \sum_{i=1}^{l^d}((T^d\times\pi)_i - 1)\left(\Psi(\xi_i) - \Psi(\sum_{j=1}^{l^d}\xi_j)\right),$$

where $(T^d\times\pi)_i = \sum_{l=1}^{L+1}\pi_l T_{il}^d$. We use gradient descent method by taking derivative of $\mathcal{L}_{[\pi]}$ with respect to π_l on the corpus to find the estimation of π. Taking derivatives with respect to π_l on the whole corpus, we obtain:

$$\mathcal{L}'_{[\pi_l]} = \sum_{d=1}^{D}\Psi(\sum_{i=1}^{l^d}\sum_{l'=1}^{L+1}\pi_{l'}\cdot T_{il'}^d)\cdot\sum_{i=1}^{l^d}T_{il}^d - \sum_{d=1}^{D}\sum_{i=1}^{l^d}\Psi(\sum_{l'=1}^{L+1}\pi_{l'}\cdot T_{il'}^d)\cdot T_{il}^d + \sum_{d=1}^{D}\sum_{i=1}^{l^d}\left(\Psi(\xi_i) - \Psi(\sum_{j=1}^{l^d}\xi_j)\right)\cdot T_{il}^d.$$

B.2.2 Optimization with respect to θ

The only term that involves θ is:

$$\mathcal{L}_{[\theta]} = \sum_{d=1}^{D}\sum_{i=1}^{N}\sum_{k=1}^{K}\gamma_{ik}\sum_{j=1}^{l^d}\log\theta_k^{(j)}\frac{\xi_j}{\sum_{j'=1}^{l^d}\xi_{j'}},$$

where ξ_j, $j \in \{1, \cdots, l^d\}$ in the document d needs to be extended to $t_l^d \cdot \xi_l^d$, $l \in \{1, \cdots, L+1\}$ for convenient to simplify $\mathcal{L}_{[\theta]}$. With the Lagrangian of the $\mathcal{L}_{[\theta]}$, which incorporate the constraint that the K-components of θ_l sum to one, we obtain the estimation of θ over the whole corpus,

$$\theta_{lk} \propto \sum_{d=1}^{D}\sum_{i=1}^{N}\gamma_{ik}^d\frac{\xi_l^d t_l^d}{\sum_{l=1}^{L+1}(\xi_l^d t_l^d)}.$$

B.2.3 Optimization with respect to ψ

To maximize with respect to ψ, we isolate corresponding terms and add Lagrange multipliers:

$$\mathcal{L}_{[\psi]} = \sum_{d=1}^{D}\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{V}\gamma_{ik}(w^d)_i^j\log\psi_{kj} + \sum_{k=1}^{K}\lambda_k(\sum_{j=1}^{v}\psi_{kj} - 1).$$

Take the derivative with respect to ψ_{kj} over the whole corpus, and set it to zero, we get:

$$\psi_{kj} \propto \sum_{d=1}^{D}\sum_{i=1}^{N}\gamma_{ik}^d(w^d)_i^j.$$

B.2.4 Optimization with respect to μ

For the Dirichlet parameters μ, the involved terms are:

$$\mathcal{L}_{[\mu]} = \sum_{d=1}^{D}\left(\log\Gamma(\sum_{j=1}^{K}\mu_j) - \sum_{i=1}^{K}\log\Gamma(\mu_i) + \sum_{i=1}^{K}(\mu_i - 1)(\Psi(\rho_i^d) - \Psi(\sum_{j=1}^{K}\rho_j^d))\right).$$

Taking the derivative with respect to μ_i gives:

$$\mathcal{L}_{[\mu_i]}^{'} = D\left(\Psi(\sum_{j=1}^{K}\mu_i) - \Psi(\mu_i)\right) + \sum_{d=1}^{D}\left(\Psi(\rho_i^d) - \Psi(\sum_{j=1}^{K}\rho_j^d)\right)$$

We can invoke the linear-time Newton-Raphson algorithm to estimate μ as same as in LDA.

APPENDIX C
CLUSTER ALGORITHM IN SOLUTION III

As shown in Eq. 4, $\pi_l, l \in (1, \cdots, L)$ is only associated with the documents who contain the l^{th} tag. Thus, before running TWTM, we can cluster the documents into several clusters with the condition that the documents which contain the same tags should be in the same cluster. It means that the documents are divided into the mutually independent space by the tags. We show a simple example as shown in Figure 12, left panel.

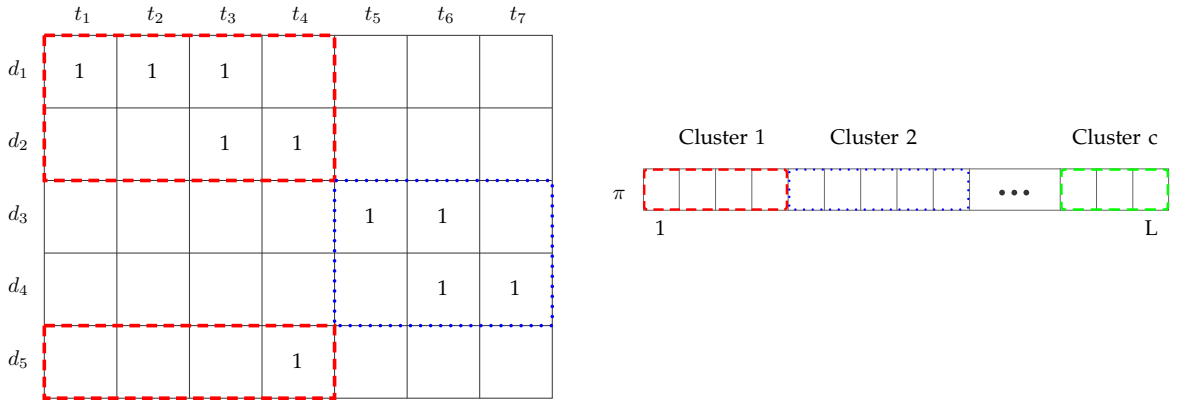


Fig. 12. Left: An example of the clustering result. Each row represents a document d in a corpora D, and Each column represents a tag t. $D_{ij} = 1$ means that t_j is given in d_i. The documents in the red circle belong to one cluster, and the documents in the blue circle belong to another cluster. Right: The illustration to update π by combine the different parts.

After document clustering, the tags contained in one cluster are not appeared to any other clusters. In this case, we could assign each cluster to different computed nodes. When update the π, we just simply combine the π^c where $c \in (1, \cdots, C)$ and C is the number of document clusters, just as shown as in Figure 12, right panel. We show the cluster process of Solution III in Algorithm 2.

Algorithm 2 The cluster process of Solution III

Input: a semi-structured corpora $D = \{(\mathbf{w}^1, \mathbf{t}^1), \ldots, (\mathbf{w}^M, \mathbf{t}^M)\}$ and the tag set T of the corpora.
Output: a cluster set C that contains all the clusters, and each cluster c in C contains a set of documents.
create a cluster set $C = \{\}$.
create a document cluster $c = \{\}$.
create $pre\_added\_docs = \{\}$ to store the documents which are ready to add into cluster c.
create a tag set $scanned\_tags = \{\}$ to store the tags which have been scanned.
add c into C.
for each tag t in T **do**
  if t is not in $scanned\_tags$ **then**
    add t into $scanned\_tags$;
    create a new cluster c, and add c into C;
  else
    continue;
  end if
  add the documents which own t into pre_added_docs;
  repeat
    for each d in the pre_added_docs **do**
      if d is not in c **then**
        add d into c;
      end if
      for each tag t^d in d **do**
        add t^d into $scanned\_tags$;
        add the documents which have t^d and not in c into pre_added_docs;
      end for
      remove d from pre_added_docs;
    end for
  until pre_added_docs is empty.
end for
return C