A Novel Collaborative Interactive Topic Model for Academic Recommendation

Abstract

Topic modeling has been extensively used in academic recommendation as it provides latent topics for each document. However, without human involvement, topic modeling method appears to be a "take it or leave it" matter, as users cannot iteratively update document-topic correlations with their prior knowledge. In this paper, we propose a novel interactive topic model based on latent Dirichlet allocation. We include human judgment in the loop by encoding feedbacks from users to tree-structured priors and re-infer certain documents via boosted Gibbs sampling. We also present a crowdsourcing framework for recommending scientific publications based on similar topics, which involves a collaborative version of the interactive topic model. Experiments show that our model presents higher efficiency and more reasonable document-topic distribution.

Introduction

Probabilistic topic models, such as probabilistic latent semantic indexing (Hofmann 1999) and latent Dirichlet allocation(LDA) (Blei, Ng, and Jordan 2003), have been recognized as useful tools for analyzing and summarizing large, unstructured documents.

While topic models provide an attractive solution for understanding contexts, the extracted topics may not always accomodate users' needs for information, which is a three-fold problem. First, the comprehension and interpretation of given documents and topics vary from person to person. People rely more on topics that are semantically coherent and accordant with their personal expertise. Second, topics generated by topic models may present a different level of granularity according to user expectation. For example, users will find the topic machine learning, artificial intelligence unsatisfying when they read an article on topic models, as they want a more specific one. Third, some topics have relatively low human-interpretability because of co-existance of irrelavant words.

Two types of methodologies are used to address this problem. Extrinsic methods make improvements by adding more side information, such as multi-faceted topics (Paul and Girju 2010), authorship information (Rosen-Zvi et al. 2004), correlated topics (Blei and Lafferty 2006). whereas intrinsic methods (Hu, Boyd-Graber, and Satinoff 2011) do not use any source or task from the dataset, but attempt to make models interactive and encode user feedback in the model. As no quantive methods can be adopted to measure how well a topic matches a document, human judgment can provide more personalized modifications for topic models. Therefore, we assume that intrinsic methods are better at analyzing textual information of a given document.

For intrinsic methods mentioned above, former researches, including (Hu and Boyd-Graber 2012), mainly focus on word distribution under each topic. However, when recommending documents to users, we should pay more attention to document-topic distribution rather than topic-word distribution. Based on this, we propose a novel interactive topic model (ITM) to analyze and to recommend scientific articles.

Considering that computational efficiency is rather important as we need to decrease waiting time in ITM, researchers have developed several effective methods for computation, including SparseLDA (Yao, Mimno, and McCallum 2009). In our ITM, we adopt similar techniques and boost the process of Gibbs sampling.

Unlike previous interactive topic modeling systems (Hoque and Carenini 2015; Pleple 2013) which are usually personalized models, we base our interactive topic model on a crowdsourcing platform. Crowdsourcing strategy accomplishes a task with feedbacks from a large number of participants. It has been used in diverse fields, including language studies (Vertanen and Kristensson 2011), tests grading (Bachrach et al. 2012). We distribute the task of labeling documents to users who read the same document. In practice, such mechanism is reasonable because labeling documents and modifying prior tree could achieve higher accuracy if more people, especially more specialists in same fields are involved. Hence, we further modify our interactive topic model into a collaborative version, enabling different users to modify the shared prior tree.

The major contributions of this paper are summarized as follows:

- We provide an interactive topic model with tree-structured priors and encode user feedback into the prior tree, and we significantly increase the computational efficiency by adopting similar mechanism like SparseLDA.

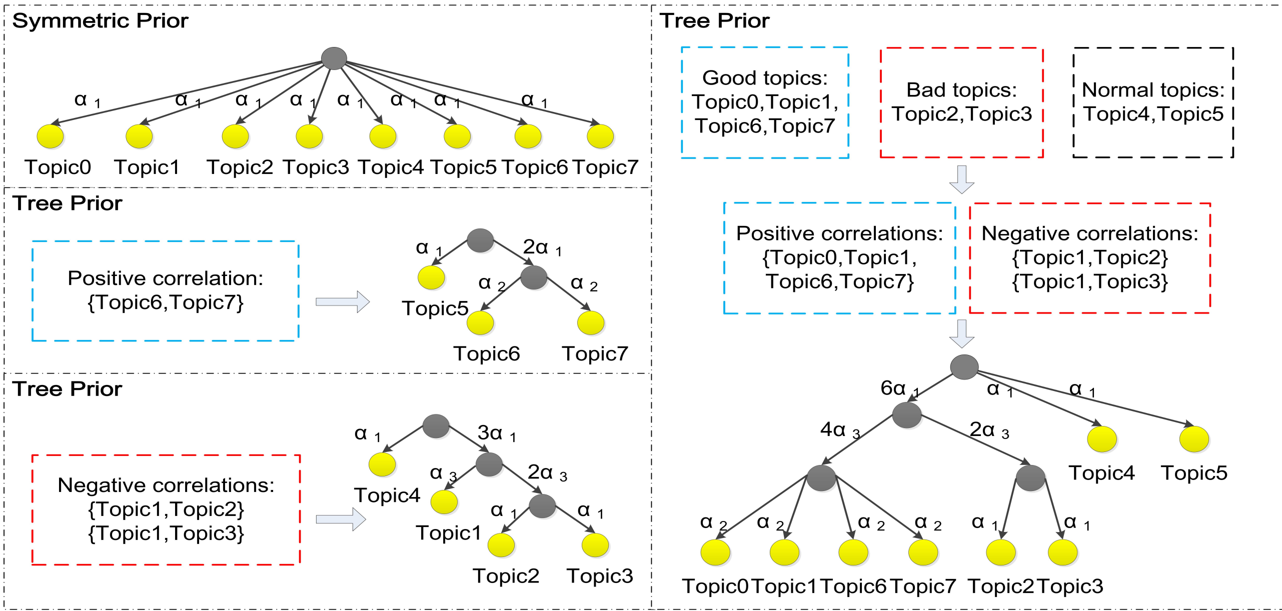

Figure 1: tree-structured priors

- We propose a crowdsourcing framework for recommending publications and further modify our interactive topic model to a collaborative version. In this scenario, users with similar interests can fix a shared prior tree, which promotes article recommendation in related topics.
- The experimental evaluation shows that our topic model successfully modifies inference results of given documents based on feedbacks from different users.

Topic Model with Tree-strutured Priors

In the standard LDA, we adopt symmetric hyper-parameter α over the document-topic distribution. It is assumed that all the topics have the same probability to be generated. In our model, we incorporate users' knowledge on topic distribution after they read documents. The users choose both the most and least relevant topics and it produces correlations among topics. We adopt (Andrzejewski, Zhu, and Craven 2009)'s approach to add user feedback into the prior knowledge in a document's topic tree. In this section, we will give detailed descriptions of this process.

Tree-based priors

In our model, the multinomial distributions for K topics are arranged in a tree. We use root node to represent a document and leaf nodes to represent topics. Each internal node represents an additional multinomial distribution. When Dirichlet parameters are large (bigger than 1), the multinomial distribution is uniform, which means that the correlated topics are likely to occur simultaneously. When Dirichlet parameters are small, the multinomial distribution is sparse. It encourages an exclusively distribution which means that the correlated topics are unlikely to occur at the same time.

Analogous to (Andrzejewski, Zhu, and Craven 2009)'s formalism of correlations among words, we define similar correlations including positive and negative ones. These two correlations are illustrated in a tree-structured model.

Referring to Figure 1, we define three parameters α_1, α_2 and α_3. We use large parameter α_2(like 100) to form positive correlations, small parameter α_3(like 10^{-6}) to form negative correlations. Parameter α_1(like 0.5) is used for uncorrelation.

Encoding user feedback into a tree

After users read a document, we will show them a list of topics generated by our topic model, then users are supposed to label the topics with tag "good" (related the basic idea of the document) or tag "bad" (unrelated to the basic idea of the document).

Recall the positive and negative correlations mentioned in the former section. In this case, positive correlations are created between every two "good" topics and negative correlations are built between a "good" topic and a "bad" one.

Our next task is to generate the prior tree based on the correlations. The standard LDA with the symmetric prior can be considered as a flat tree, with all topics are leaf nodes directly linked to the root node with the same prior, as shown in the upper left part of Figure 1. When considering constraints, we replace all topics involved in the correlation with a single new child, and then link all the correlated topics to this new node. Examples of positive correlation and negative correlation are shown in middle left and lower left part of Figure 1.

As positive correlation is transitive and we can merge all the correlated topics into a cluster, "bad" topics are actually correlated negatively with all of the "good" topics in the same cluster. Such example is presented in the right part of Figure 1.

Boosted Gibbs Sampling-based Inference

In this section, we will give further analysis of the inference process and efficient sampling techniques in order to boost computational efficiency.

Inference process

After new constraints are added, we first generate a modified prior tree and then start the re-inference process of related documents based on Gibbs Sampling. Inference procedures are as follows:

1. For each topic $k \in \{1, ..., K\}$,
   (a) draw a V-dimensional multinomial distribution over all words: $\pi_k \sim \mathbf{Dirichlet}(\beta)$
2. For each document $d \in \{1, ..., M\}$,
   (a) draw a multinomial distribution over topics:
      i. start from the ROOT node, draw a distribution over branches(topics and constraints) $\theta_{d,ROOT} \sim \mathbf{Dirichlet}(\overrightarrow{\alpha_{ROOT}})$
      ii. for each constraint, draw a multinomial distribution over its branches similar to the former step: $\theta_{d,i} \sim \mathbf{Dirichlet}(\overrightarrow{\alpha_i})$.
      iii. continue the former step, until all topics are included.
   (b) For n_{th} token w in this document d:
      i. choose either a topic or a constraint from $Mult(\theta_{d,ROOT})$:
         A if we choose a topic, draw that topic;
         B otherwise if we choose a constraint index l_n, continue choose its branch from $Mult(\theta_{d,l_n})$, until a topic$z_{d,n}$ is chosen.
      ii. draw a token $w_{d,n}|z_{d,n} \sim Mult(\pi_{z_{d,n}})$

Distribution equations

In this model, we can still turn to Gibbs sampling for inference as these models retain conjugacy.

For standard LDA, the common practice is to integrate out the per-document distribution over topics and the topic distribution over tokens. The only latent variable left to sample is the per-token topic assignment

$$p(z_{d,n} = k|Z_{-}, w_{d,n}) \propto (\alpha_k + n_{k|d}) \frac{\beta + n_{w_{d,n}|k}}{\beta V + n_{\cdot|k}} \quad (1)$$

where d is the document index, and n is the token index in that document; $n_{k|d}$ is topic k's count in the document d; α and β are priors for topics and words; Z_{-} are the topic assignments excluding the current token; $n_{w_{d,n}|k}$ is the count of tokens with word $w_{d,n}$ assigned to topic k; V is the vocabulary size, and $n_{\cdot|k}$ is the count of all tokens assigned to topic k.

For our proposed tree-based topic model, we first define four probability functions: transition probability, path probability, topic assignment probability and token probability.

$$\begin{aligned} p_{trans} &= p(\theta_{d,i}|\alpha_i) \\ p_{path} &= p(l_{d,n}|\theta_d) \\ p_{topic} &= p(z_{d,n}|l_{d,n}) \\ p_{token} &= p(w_{d,n}|z_{d,n}) \end{aligned} \quad (2)$$

then the joint probability looks like this,

$$\begin{aligned} &p(L, Z, W, \pi, \theta; \alpha, \beta) \\ &= \prod_{k=1}^{K} p(\pi_k|\beta) \left[\prod_{d=1}^{D} \prod_{i} p_{trans} \left[\prod_{n=1}^{N} p_{path} \cdot p_{topic} \cdot p_{token} \right] \right] \end{aligned} \quad (3)$$

where i is an internal node in the prior tree for α; $p(l_{d,n}|\theta_d)$is the probability of a given path $l_{d,n}$ in the tree; $p(z_{d,n}|l_{d,n})$ is the possibility of a topic $z_{d,n}$ being generated by a path $l_{d,n}$; and $p(w_{d,n}|l_{d,n})$ is the possibility of generating a word token$w_{d,n}$ given topic $z_{d,n}$.

Like standard LDA, since the conjugate prior of the multinomial is the Dirichlet, we can also integrate out the transition distribution θ and topic-word distribution π in the conditional distribution equation

$$\begin{aligned} &p(l_k = \mu|\mathbf{L}_{-}, \mathbf{z} = \mathbf{k}, \mathbf{Z}_{-}, \mathbf{w}) \\ &\propto (\alpha_{root \to i} + n_{root \to i}) \prod_{i \to j \in \mu} \frac{\alpha_{i \to j} + n_{i \to j}}{\sum_{j'} \alpha_{i \to j'} + n_{i \to j'}} \frac{\beta + n_{w|k}}{\beta V + n_{\cdot|k}} \end{aligned} \quad (4)$$

where μ is path in a certain document and $\alpha_{i \to j}$ is the prior for edge $i \to j$ in the prior tree; $n_{i \to j}$ is the count of edges that link the node i and node j. All the other terms are the same as those in standard LDA: $n_{w|k}$ is the count of word w in the topic k, and β is the symmetric Dirichlet hyperparameter for word distribution. Note that node i is directly linked to ROOT node, thus equation for those unconstrained topics is exactly the same as standard LDA, in which case the path μ is zero-length.

$$p(l_k = \mu|L_{-}, z = k, Z_{-}, w) \propto (\alpha_{root \to i} + n_{root \to i}) \cdot \frac{\beta + n_{w|k}}{\beta V + n_{\cdot|k}} \quad (5)$$

Boosted inference for tree-based topic models

The SparseLDA schema for speeding inference begins by rearranging standard LDA's sampling equation(1) as

$$\begin{aligned} &p(z_{d,n} = k|Z_{-}, w_{d,n}) \propto (\alpha_k + n_{k|d}) \frac{\beta + n_{w_{d,n}|k}}{\beta V + n_{\cdot|k}} \\ &= \underbrace{\frac{\alpha_k \beta}{\beta V + n_{\cdot|k}}}_{s_{LDA}} + \underbrace{\frac{n_{k|d} \beta}{\beta V + n_{\cdot|k}}}_{r_{LDA}} + \underbrace{\frac{(\alpha_k + n_{k|d}) n_{w|k}}{\beta V + n_{\cdot|k}}}_{q_{LDA}}. \end{aligned} \quad (6)$$

Since our topic model enjoys same sparsity – each document has only handful topics and each topic contains a limited number of words in a corpus, we can adopt such method in order to boost the process of sampling. This is particularly crucial for interactive topic modeling as slightest delay might annoy an impatient user.

To match the form of equation (6), firstly we can define

$$P_{\mu,k} = \prod_{i\to j\in\mu} \sum_{j'} (\alpha_{i\to j'} + n_{i\to j'}) \tag{7}$$

$$M_\mu = (\alpha_{root\to i} + n_{root\to i}) \prod_{i\to j\in\mu} \alpha_{i\to j} \tag{8}$$

$$N_\mu = (\alpha_{root\to i} + n_{root\to i}) \cdot \left[\prod_{i\to j\in\mu} (\alpha_{i\to j} + n_{i\to j}) - \prod_{i\to j\in\mu} \alpha_{i\to j} \right] \tag{9}$$

We can refactor the sampling equation (4), yielding three parts analogous to SparseLDAf,

$$\begin{aligned} & p(l_k = \mu | \mathbf{L}_-, \mathbf{z} = \mathbf{k}, \mathbf{Z}_-, \mathbf{w}) \\ & \propto (\alpha_{root\to i} + n_{root\to i}) \prod_{i\to j\in\mu} \frac{\alpha_{i\to j} + n_{i\to j}}{\sum_{j'} \alpha_{i\to j'} + n_{i\to j'}} \cdot \frac{\beta + n_{w|k}}{\beta V + n_{\cdot|k}} \\ & \propto \frac{(M_\mu + N_\mu)(\beta + n_{w|k})}{P_{\mu,k} \cdot (\beta V + n_{\cdot|k})} \\ & \propto \frac{\beta M_\mu}{P_{\mu,k}(\beta V + n_{\cdot|k})} + \frac{\beta N_\mu}{P_{\mu,k}(\beta V + n_{\cdot|k})} + \frac{n_{w|k}(N_\mu + M_\mu)}{P_{\mu,k}(\beta V + n_{\cdot|k})} \end{aligned} \tag{10}$$

Likewise, we can then define three 'buckets':

$$\begin{aligned} s &\triangleq \sum_{\mu,k} \frac{\beta M_\mu}{P_{\mu,k}(\beta V + n_{\cdot|k})} \\ q &\triangleq \sum_{\mu,k} \frac{\beta N_\mu}{P_{\mu,k}(\beta V + n_{\cdot|k})} \\ r &\triangleq \sum_{\mu,k} \frac{n_{w|k}(N_\mu + M_\mu)}{P_{\mu,k}(\beta V + n_{\cdot|k})} \end{aligned} \tag{11}$$

Recall that SparseLDA effectively increase the computational efficiency. We can adopt similar methods and make use of the same sparsity. First, the topic-word bucket r is only non-zero when words appear in topic k($n_{w|k}$ is non-zero); and second, the document-topic bucket q is only non-zero when topics appear in document d($n_{i\to j}$ is non-zero for every edge $i \to j$ in the path μ).

Sorting During the sampling process, the efficient sampling assignment within each bucket is important. (Yao, Mimno, and McCallum 2009) places probability mass in decreasing order by sorting the $n_{w|k}$ decreasingly when considering latent variable assignments; (Hu et al. 2014) encodes the paths of word with a dataset named edge-masked count(EMC) and extends the sorting strategy to the path.

In this model, we propose two types of sorting. First, we use the method of (Yao, Mimno, and McCallum 2009) to sort the $n_{w|k}$. When sampling a topic and path from bucket q, we sample based on the decreasing of the count of word in a topic, which roughly correlates with probability mass. Therefore, we will on average choose our sample from the conditional distribution more quickly. The same influence will take place in the bucket q and r, once we sort and keep the EMC of each path of topics in decreasing order.

A crowdsourcing framework involving ITM

In this section, we will discuss our crowdsourcing framework and compare it to other methods used to recommend scientific articles.

Interactive topic modeling

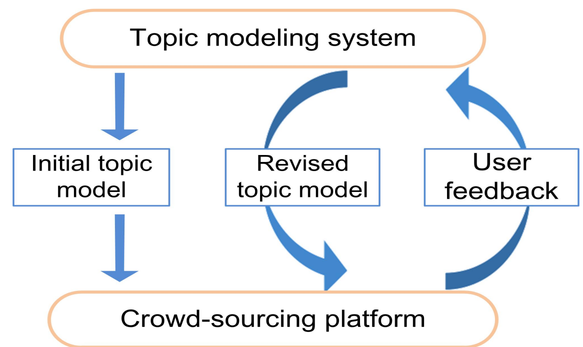

Figure 2: The ITM system based on crowdsourcing platform which gather user feedback to refine iteratively

We propose an ITM system on a crowdsourcing platform which allows users to add correlations to topics interactively. Every time users refine the topics, the system resamples and reinfers the document set to match up with users' expectation. The process is shown in Figure 2.

First we perform unconstrained LDA to draw topic information in the corpus, and then we use the topic-word distribution, estimated from the whole corpus, to re-infer each document respectively and extract each document's topic distribution.

When the user finishes reading a document, he is shown with the topic lists and give feedback to the topics using his own knowledge to the document. We allow users to pick the topics that he thinks are most related to the document ("good" topics) and ones that are least related ("bad" topics). It gives underlying information that the most correlated topics tackle similar problems and have close relationship. The model encodes the correlations into a prior tree and update it in following loops.

The task of giving feedback to each document is a consuming job. We crowdsource it to all the users who have read the document and aggregate their feedback into one correlation. To make the result more convincing, we pick out competitive users and give priority to their opinions. We assume that the more the user reads about one topic, the more skilled he is. It is not difficult to classify users into different levels, according to their previous reading records, e.g. experts, researchers and beginners.

In the recommending process, as each user has a user file recording reading history, we draw out dominant topics (reflection of user's interest) from these documents and find other documents sharing similar topic distribution. In this way, we gather information from a crowdsourcing platform and apply it to personalized recommendation.

Comparison with other recommending models

Like our document-topic ITM, other related models such as collaborative filtering, collaborative topic regression model (Wang and Blei 2011), interactive topic-word tree-structured model (Hu, Boyd-Graber, and Satinoff 2011), also incorporate user knowledge in recommendation. In this section, we will compare our model to the collaborative filtering and interactive topic-word tree-structured model.

Comparison with collaborative filtering Collaborative filtering model gives recommendation based on user community with similar preference for articles. It assumes that articles read by most of users sharing similar interests will appeal to another user. The disadvantage is that it doesnt use the content of the articles. Users may overlap in some interested topics and vary in others. The model can't distinguish which topic the article is about.

Suppose there are two users interested in supervised learning, they both read many articles on support vector machines algorithm. They share one common interest so their preference of articles has a big influence on each other in recommendation. In fact, one user uses the algorithm in image retrieval and the other is interested in its application in biology. If the user in the field of biology reads an article in biological field, for example, gene expression of a DNA microarray, there is a good probability that the system will recommend it to the user on image retrieval. However, the topics of gene expression data and image retrieval dont have a direct link, and it is unreasonable to recommend the article about gene expression data to one who wants articles in image retrieval. This is the deficiency of collaborative filtering model.

However, in our model we focus on the relations between topics. If no user gives feedback that biology and image retrieval appear simultaneously in one article, no correlations exist between these two topics. Since we recommend articles based on specific topics, our model avoids the situation that user receives recommended articles in topics he doesn't like.

Comparison with topic-word ITM The interactive topic-word tree-structured model allows users to make refinements to topics so they have clear meanings. Unlike our model, it uses a tree structure in the topic-word distribution. It includes prior domain knowledge that the co-occurrence of particular words forms a topic. It allows users to pick the words that best describe the topic and those have the least relations. By doing so, users can split topics containing several themes into different topics and merge several sub-topics into one.

Our model is from a different aspect that co-occurrence of good topics better labels a certain document. While the former model modifies topic-word distribution and applies the same prior tree to each single document when sampling, our model treat every document separately and use the modified prior tree to fix the document-topic distribution when re-inferring a given document. By re-inferring a chosen document instead of re-sampling the whole corpus, we provide users with more accurate information concerning this single document without influencing other documents.

Experiments

In this section, we examine the viability of our collaborative ITM model with constraints added to topics and its effectiveness in computation with SparseLDA scheme.

Preprocessing

We use a dataset of 42,673 publications crawled from IEEE library. As these documents are all scientific articles, we remove some frequently used but meaningless words like 'result' and 'proposed' besides normal stopwords.

Computational efficiency

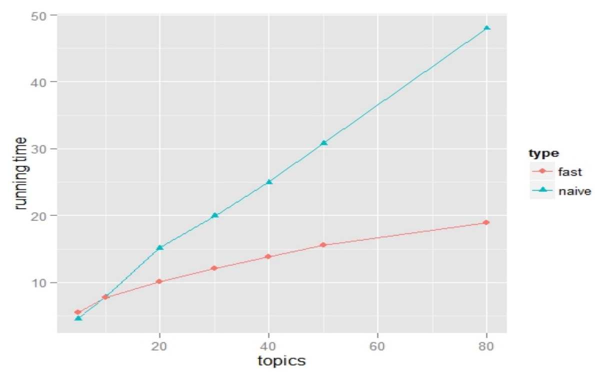

Figure 3: Running time of naive/fast algorithm

To check the computational efficiency in our model, we compare it with naive sampling scheme in 300 iterations of Gibbs Sampling of the training database. We set topic number to be $\{5, 10, 20, 30, 40, 50, 80\}$. The running time per iteration and the corresponding topic numbers are plot in Figure 3.

We can see that except for the first round, our boosted Gibbs sampling process exceeds naive sampling to a large extent. As topic number increases, sparsity of topic distribution becomes more obvious, thus the boosted Gibbs sampling algorithm shows clearer advantage.

User case simulation

To confirm the practicability of our proposed framework, we design a simulation experiment as a prototype of the crowdsourcing system involving ITM.

We start the experiment by applying unconstrained topic model on the training dataset to extract the topic distribution of the library and the word distribution under each topic. We set the topic number to be 20, and use the hyper-parameter α_1=0.5 for uncorrelated topics, α_2=100 for "good" topics, $\alpha_3 = 10^{-6}$ for "bad" topics and β=0.1 for words. After 300 iterations, at which the topic chain is supposed to converge, we apply the estimated distribution to each article and run the inference for another 100 iterations respectively.

Modification of model To examine the viability of the ITM model, we focus on one article *Optimal spectrum sensing framework for cognitive radio networks* to see how cor-

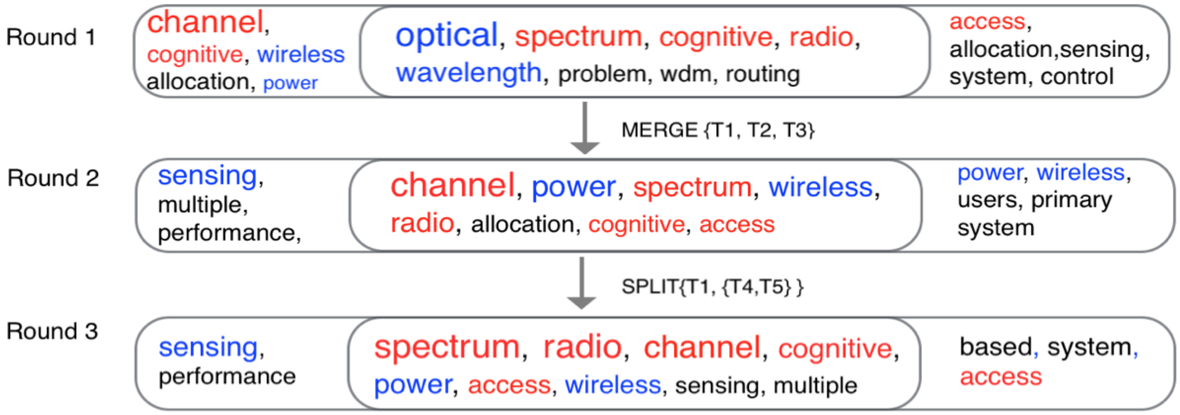

Figure 4: It shows the topic distribution after three Rounds. Words in the middle bucket of the first line represent the most correlated topic (top topic) in each round. Words in red are "good" words related to the document while words in blue are "bad" ones. In round 1, *spectrum,cognitive,radio* are relevant words but *optical, power* is less relevent. After merging with other "good" topic and spliting with "bad" topics, these "bad" words are kicked off and other "good" words like *channel, access* are included in the top topic.

relation reshapes the topic distribution. 50 volunteers are arranged to simulate users' behavior in a way similar to crowdsourcing.

According to the feedbacks from users, these topics are "good" topics, so we update the new topic tree by merging the three topics, the result turns out to be better as the relevant topics own a higher distribution to be generated. In the next round, we repeat the first round except that we split the most relevant topic with another two "bad" topics based on the feedback. We can see from the final result that as more and more feedbacks are collected and constraints are added, we get more specific and accurate topic of this article. After two rounds of modifications, the top topic can be considered a good label of this article. Shown in Figure 4, the middle bucket of top line of the flow-chart shows the word distribution of the most relevant topic associated to the article, while the buckets in the two sides shows the word distribution of the other topics to be merged or splited.

Recommendation Focused on a particular topic (about spectrum), we select five most correlated documents and display the topic distribution of them (Doc1 is the article used for fixing topics) in Figure 5. These articles share a similar topic distribution thus are considered appropriate to be recommended to potential readers. We can infer from their titles and abstracts that these five articles are closely related with common research topics, thus our recommendation system presents a satisfying result.

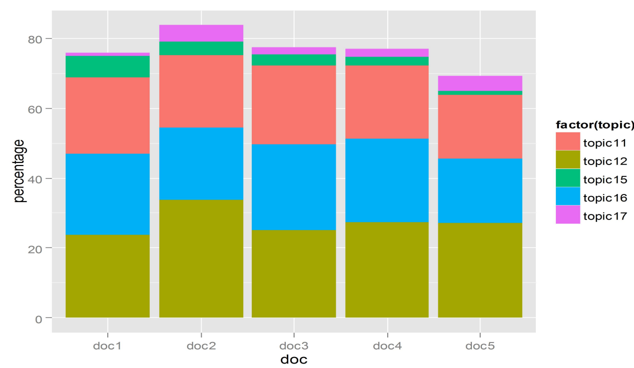

Figure 5: Showing topic distribution of five docs. Topic 11, 12, 15, 16, 17 are Doc1's most relevant topics
Doc1: *Optimal spectrum sensing framework for cognitive radio networks*;
Doc2: *Dynamic spectrum leasing in cognitive radio networks via primary-secondary user power control games*;
Doc3: *Cooperative Communication-Aware Spectrum Leasing in Cognitive Radio Networks*;
Doc4: *Dynamic Spectrum Leasing and Service Selection in Spectrum Secondary Market of Cognitive Radio Networks*;
Doc5: *Pricing-Based Decentralized Spectrum Access Control in Cognitive Radio Networks*;

Conclusion and future works

In this paper, we present an interactive topic model(ITM) for analyzing scientific publications. Our model encodes user feedback into tree-structured priors and updates document-topic distribution iteratively. We also use efficient inference techniques to reduce computational cost. Furthermore, we have shown how this model can be used in a crowdsourcing framework. Our experiment and case study have demonstrated that our model presents increase in computational efficiency and great modification of document-topic distribution compared to the standard LDA.

Further, our framework could be implemented in real-world recommender systems. Also, as more algorithms for sampling are being proposed, we can adopt them to further boost the inference in order to build a faster system.

References

Andrzejewski, D.; Zhu, X.; and Craven, M. 2009. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 25–32. New York, NY, USA: ACM.

Bachrach, Y.; Graepel, T.; Minka, T.; and Guiver, J. 2012. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. *arXiv preprint arXiv:1206.6386*.

Blei, D., and Lafferty, J. 2006. Correlated topic models. *Advances in neural information processing systems* 18:147.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3:993–1022.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, 50–57. New York, NY, USA: ACM.

Hoque, E., and Carenini, G. 2015. Convisit: Interactive topic modeling for exploring asynchronous online conversations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, 169–180. New York, NY, USA: ACM.

Hu, Y., and Boyd-Graber, J. 2012. Efficient tree-based topic modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, 275–279. Stroudsburg, PA, USA: Association for Computational Linguistics.

Hu, Y.; Boyd-Graber, J.; Satinoff, B.; and Smith, A. 2014. Interactive topic modeling. *Machine Learning* 95(3):423–469.

Hu, Y.; Boyd-Graber, J.; and Satinoff, B. 2011. Interactive topic modeling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 248–257. Stroudsburg, PA, USA: Association for Computational Linguistics.

Paul, M., and Girju, R. 2010. A two-dimensional topic-aspect model for discovering multi-faceted topics. *Urbana* 51:61801.

Pleple, Q. 2013. Interactive topic modeling.

Rosen-Zvi, M.; Griffiths, T.; Steyvers, M.; and Smyth, P. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, 487–494. AUAI Press.

Vertanen, K., and Kristensson, P. O. 2011. The imagination of crowds: Conversational aac language modeling using crowdsourcing and large data sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, 700–711. Stroudsburg, PA, USA: Association for Computational Linguistics.

Wang, C., and Blei, D. M. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, 448–456. New York, NY, USA: ACM.

Yao, L.; Mimno, D.; and McCallum, A. 2009. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, 937–946. New York, NY, USA: ACM.