Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Ordering-Sensitive and Semantic-Aware Topic Modeling

Min Yang
The University of Hong Kong
myang@cs.hku.hk

Tianyi Cui
Zhejiang University
tianyicui@gmail.com

Wenting Tu
The University of Hong Kong
wttu@cs.hku.hk

Abstract

Topic modeling of textual corpora is an important and challenging problem. In most previous work, the "bag-of-words" assumption is usually made which ignores the ordering of words. This assumption simplifies the computation, but it unrealistically loses the ordering information and the semantic of words in the context. In this paper, we present a Gaussian Mixture Neural Topic Model (GMNTM) which incorporates both the ordering of words and the semantic meaning of sentences into topic modeling. Specifically, we represent each topic as a cluster of multi-dimensional vectors and embed the corpus into a collection of vectors generated by the Gaussian mixture model. Each word is affected not only by its topic, but also by the embedding vector of its surrounding words and the context. The Gaussian mixture components and the topic of documents, sentences and words can be learnt jointly. Extensive experiments show that our model can learn better topics and more accurate word distributions for each topic. Quantitatively, comparing to state-of-the-art topic modeling approaches, GMNTM obtains significantly better performance in terms of perplexity, retrieval accuracy and classification accuracy.

Introduction

With the growing of large collection of electronic texts, much attention has been given to topic modeling of textual corpora, designed to identify representations of the data and learn thematic structure from large document collections without human supervision. Topic models have been applied to a variety of applications, including information retrieval (Wei and Croft 2006), collaborative filtering (Marlin 2003), authorship identification (Rosen-Zvi et al. 2004) and opinion extraction (Lin et al. 2012), etc. Existing topic models (Griffiths and Tenenbaum 2004; Mcauliffe and Blei 2008; Blei 2012) are built based on the assumption that each document is represented by a mixture of topics, where each topic defines a probability distribution over words. These models, including the probabilistic latent semantic analysis (PLSA) (Hofmann 1999) model and latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003) model, can be viewed as graphical models with latent variables. Some nonparametric extensions to these models have also been quite successful (Teh et al. 2006; Steyvers and Griffiths 2007). Nevertheless, exact inference for these model is computationally hard, so one has to resort to slow or inaccurate approximations to compute the posterior distribution over topics. New undirected graphical model approaches, including the Replicated softmax model (Hinton and Salakhutdinov 2009), are also successfully applied to exploring the topics of the text, and in particular cases they outperform LDA (Srivastava, Salakhutdinov, and Hinton 2013).

A major limitation of these topic models and many of their extensions is the bag-of-word assumption, which assumes that document can be fully characterized by bag-of-word features. This assumption is favorable in the computational point of view, but loses the ordering of the words and cannot properly capture the semantics of the context. For example, the phrases "the department chair couches offers" and "the chair department offers couches" have the same unigram statistics, but are about quite different topics. When deciding which topic generated the word "chair" in the first sentence, knowing that it was immediately preceded by the word "department" makes it much more likely to have been generated by a topic that assigns high probability to words related to university administration (Wallach 2006).

There has been little work on developing topic models where the order of words is taken into consideration. To remove the assumption that the order of words is negligible, Gruber, Weiss, and Rosen-Zvi (2007) propose modeling the topics of words in the document via a Markov chain. Wallach (2006) explores a hierarchical generative probabilistic model that incorporates both n-gram statistics and latent topic variables. Even though they consider the order of words to some extent, their model is still not capable of characterizing the semantics of words. For example, the integer representation of the words "teacher" and "teach" are completely unrelated, even if we know they have strong semantic connections and are very likely belonging to the same topic. To seek a distributed way of representing words that capture semantic similarities, several Neural Probabilistic Language Models (NPLMs) have been proposed (Mnih and Hinton 2009; Mnih and Teh 2012; Mnih and Kavukcuoglu 2013; Mikolov et al. 2013; Le and Mikolov 2014). Nevertheless, the dense word embeddings learned by previous NPLMs cannot be directly interpreted as topics. This is because that word embeddings are usually considered opaque, in the sense that it is difficult to assign meanings to the the vector representation.

In this paper, we proposed a novel topic model called the Gaussian Mixture Neural Topic Model (GMNTM). The work is inspired by the recent neural probabilistic language models (Mnih and Hinton 2009; Mnih and Teh 2012; Mnih and Kavukcuoglu 2013; Mikolov et al. 2013; Le and Mikolov 2014). We represent the topic model as a Gaussian mixture model of vectors which encode words, sentences and documents. Each mixture component is associated with a specific topic. We present a method that jointly learns the topic model and the vector representation. As in NPLM methods, the word embeddings are learnt to optimize the predictability of a word using its surrounding words, with an important constraint that the vector representations are sampled from the Gaussian mixture which represents topics. Because the semantic meaning of sentences and documents are incorporated to infer the topic of a specific word, in our model, words with similar semantics are more likely to be clustered into the same topic, and topics of sentences and documents are more accurately learned. It potentially overcomes the weaknesses of the bag-of-word method and the bag-of-n-grams method, both of which don't use the order of words or the semantic of the context. We conduct experiments to verify the effectiveness of the proposed model on two widely used publicly available datasets. The experiment results show that our model substantially outperforms the state-of-the-art models in terms of perplexity, document retrieval quality and document classification accuracy.

Related works

In the past decade, a great variety of topic models have been proposed, which can extract interesting topics in the form of multinomial distributions automatically from texts (Blei, Ng, and Jordan 2003; Griffiths and Tenenbaum 2004; Blei 2012; Gruber, Weiss, and Rosen-Zvi 2007; Hinton and Salakhutdinov 2009). Among these approaches, LDA (Blei, Ng, and Jordan 2003) and its variants are the most popular models for topic modeling. The mixture of topics per document in the LDA model is generated from a Dirichlet prior mutual to all documents in the corpus. Different extensions of the LDA model have been proposed. For example, Teh et al. (2006) assumes that the number of mixture components is unknown a prior and is to be inferred from the data. Mcauliffe and Blei (2008) develops a supervised latent Dirichlet allocation model (sLDA) for document-response pairs. Recent work incorporates context information into the topic modeling, such as time (Wang and McCallum 2006), geographic location (Mei et al. 2006), authorship (Steyvers et al. 2004), and sentiment (Yang et al. 2014b; 2014a), to make topic models fit expectations better.

Recently, there are several undirected graphical models being proposed, which typically outperform LDA. Mcauliffe and Blei (2008) present a two-layer undirected graphical model, called "Replicated Softmax", that can be used to model and automatically extract low-dimensional latent semantic representations from a large unstructured collection of document. Hinton and Salakhutdinov (2009) extend "Replicated Softmax" by adding another layer of hidden units on top of the first with bipartite undirected connections. Neural network based approaches, such as Neural Autoregressive Density Estimators (DocNADE) (Larochelle and Lauly 2012) and Hybrid Neural Network-Latent Topic Model (Wan, Zhu, and Fergus 2012), are also shown outperforming the LDA model.

However, all of these these topic models employ the bag-of-words assumption, which is rarely true in practice. The bag-of-word assumption loses the ordering of the words and ignore the semantics of the context. There are several previous literature taking the order of words into account. Wallach (2006) explores a hierarchical generative probabilistic model that incorporates both n-gram statistics and latent topic variables. They extend a unigram topic model so that it can reflect properties of a hierarchical Dirichlet bigram model. Gruber, Weiss, and Rosen-Zvi (2007) propose modeling the topic of words a Markov chain. Florez and Nachman (2014) exploits the semantics regularities captured by a Recurrent Neural Network (RNN) in text documents to build a recommender system. Although these methods captures the ordering of words, none of them them consider the semantics, thus they cannot capture the semantic similarities between words such as "teach" and "teacher". In contrast, our model is inspired by the recent work in learning vector representations of words which are proved to capture the semantics of texts (Mnih and Hinton 2009; Mnih and Teh 2012; Mnih and Kavukcuoglu 2013; Mikolov et al. 2013; Le and Mikolov 2014). Our topic model captures both the ordering of words and the semantics of the context. As a consequence, semantically similar words are more likely having similar topic distribution (e.g., "Jesus" and "Christ").

The GMNTM Model

In this section, we first describe the GMNTM model as a probabilistic generative model. Then we illustrate the inference algorithm for estimating the model parameters.

Generative model

We assume there are W different words in the vocabulary and there are D documents in corpus. For each word $w \in \{1, \ldots, W\}$ in vocabulary, there is an associated V-dimensional vector representation $\mathrm{vec}(w) \in \mathcal{R}^V$ for the word. Each document in corpus with index $d \in \{1, \ldots, D\}$ also has a vector representation $\mathrm{vec}(d) \in \mathcal{R}^V$. If all the documents contain S sentences, then these sentences are indexed by $s \in \{1, \ldots, S\}$. The sentence with index s is associated with a vector representation $\mathrm{vec}(s) \in \mathcal{R}^V$.

There are T topics in the GMNTM model, where T is designated by the user. Each topic corresponds to a Gaussian mixture component. The k-th topic is represented by a V-dimensional Gaussian distribution $\mathcal{N}(\mu_k, \Sigma_k)$ with mixture weight $\pi_k \in \mathcal{R}$, where $\mu_k \in \mathcal{R}^V$, $\Sigma_k \in \mathcal{R}^{V \times V}$, and $\sum_{k=1}^T \pi_k = 1$. The parameters of the Gaussian mixture model are collectively represented by

$$\lambda = \{\pi_k, \mu_k, \Sigma_k\} \quad k = 1, \ldots, T \tag{1}$$

Given the collection of parameters, we use

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{T} \pi_i \mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i) \tag{2}$$

to represent the probability distribution for sampling a vector $\mathbf{x}$ from the Gaussian mixture model.

We describe the procedure that the corpus is generated. Given the Gaussian mixture model λ, the generative process is described as follow: for each word w in the vocabulary, we sample its topic $z(w)$ from the multinomial distribution $\pi := (\pi_1, \pi_2, \ldots, \pi_T)$ and sample its vector representation $\text{vec}(w)$ from distribution $\mathcal{N}(\mu_{z(w)}, \Sigma_{z(w)})$. Equivalently, the vector $\text{vec}(w)$ is sampled from the Gaussian mixture model parameterized by λ. For each document d and each sentence s in the document, we sample their topics $z(d)$, $z(s)$ from distribution π and sample their vector representations, namely $\text{vec}(d)$ and $\text{vec}(s)$, also from the Gaussian mixture model. Let $\varPsi$ be the collection of latent vectors associated with all the words, sentences and documents in the corpus,

$$\varPsi := \{\text{vec}(w)\} \cup \{\text{vec}(d)\} \cup \{\text{vec}(s)\} \tag{3}$$

For each word slot in the sentence, its word realization is generated according to the document's vector $\text{vec}(d)$, the current sentence's vector $\text{vec}(s)$ as well as at most m previous words in the same sentence. Formally, for the i-th location in the sentence, we represent its word realization by w_i. The probability distribution of w_i is defined by:

$$\begin{aligned} & p\left(w_i = w | d, s, w_{i-m}, \ldots, w_{i-1}\right) \\ & \propto \exp(a_{\text{doc}}^w + a_{\text{sen}}^w + \sum_{t=1}^{m} a_t^w + b) \end{aligned} \tag{4}$$

where a_{doc}, a_{sen} and a_t are influences from the document, the sentence and the previous word, respectively. They are defined by

$$a_{\text{doc}}^w = \langle u_{\text{doc}}^w, \text{vec}(d) \rangle \tag{5}$$

$$a_{\text{sen}}^w = \langle u_{\text{sen}}^w, \text{vec}(s) \rangle \tag{6}$$

$$a_t^w = \langle u_t^w, \text{vec}(w_{i-t}) \rangle \tag{7}$$

Here, $u_{\text{doc}}^w, u_{\text{sen}}^w, u_t^w \in \mathcal{R}^V$ are parameters of the model, and they are shared across all slots in the corpus. We use U to represent this collection of vectors,

$$U := \{u_{\text{doc}}, u_{\text{sen}}\} \cup \{u_t | t \in 1, 2, \ldots, m\}\} \tag{8}$$

Combining the equations above, the probability distribution of w_i is defined by a multi-class logistic model, where the features come from the vectors associated with the document, the sentence and the m previous words. By estimating the model parameters, we learn the word representations that make one word predictable from its previous words and the context. Jointly, we learn the distribution of topics that words, sentences and documents belong to.

Given the model parameters and the vectors for documents, sentences and words, we can infer the posterior probability distribution of topics. In particular, for a document d with vector representation $\text{vec}(d)$, the posterior distribution of its topic, namely $q(z(d))$, is easy to calculate. For any $z \in 1, 2, \ldots, T$, we have

$$q(z(d) = z) = \frac{\pi_z \mathcal{N}\left(\text{vec}(d) | \mu_z, \Sigma_z\right)}{\sum_{k=1}^{T} \pi_k \mathcal{N}\left(\text{vec}(d) | \mu_k, \Sigma_k\right)}. \tag{9}$$

Similarly, for each sentence s in the document d, the posterior distribution of its topic is

$$q(z(s) = z) = \frac{\pi_z \mathcal{N}\left(\text{vec}(s) | \mu_z, \Sigma_z\right)}{\sum_{k=1}^{T} \pi_k \mathcal{N}\left(\text{vec}(s) | \mu_k, \Sigma_k\right)}. \tag{10}$$

For each word w in the vocabulary, the posterior distribution of its topic is similarly calculated as

$$q(z(w) = z) = \frac{\pi_z \mathcal{N}\left(\text{vec}(w) | \mu_z, \Sigma_z\right)}{\sum_{k=1}^{T} \pi_k \mathcal{N}\left(\text{vec}(w) | \mu_k, \Sigma_k\right)} \tag{11}$$

Finally, for each word slot in the document, we also want to explore its topic. Since the topic of a particular location in the document is affected by its word realization and the sentence/document it belongs to, we define the probability of it belonging to topic z proportional to the product of $q(z(w) = z)$, $q(z(s) = z)$ and $q(z(d) = z)$, where w, s, and d are the word, the sentence and the document that this word slot associates with.

Estimating model parameters

We estimate the model parameters λ, U and $\varPsi$ by maximizing the likelihood of the generative model. The parameter estimation consists of two stages. In Stage I, we maximize the likelihood of the model with respect to λ. Since λ characterizes a Gaussian mixture model, this procedure can be implemented by the Expectation Maximization (EM) algorithm. In Stage II, we maximize the model likelihood with respect to U and $\varPsi$, this procedure can be implemented by stochastic gradient descent. We alternatively execute Stage I and Stage II until the parameters converge. The algorithm in this section is summarized in Algorithm 1.

Stage I: Estimating λ In this stage, the latent vector of words, sentences and documents are fixed. We estimate the parameters of the Gaussian mixture model $\lambda = \{\pi_k, \mu_k, \Sigma_k\}$. This is a classical statistical estimation problem which can be solved by running the EM algorithm. The reader can refer to the book (Bishop 2006) for the implementation details.

Stage II: estimating U and $\varPsi$ When λ is known and fixed, we estimate the model parameters U and the latent vectors $\varPsi$ by maximizing the log-likelihood of the generative model. In particular, we iteratively sample a location in the corpus, and consider the log-likelihood of the observed word at this location. Let the word realization at location i be represented by w_i. The log-likelihood of this location is equal to

$$\begin{aligned} J_i(U, \varPsi) = & \log(p(\varPsi|\lambda)) + a_{\text{doc}}^{w_i} + a_{\text{sen}}^{w_i} + \sum_{t=1}^{m} a_t^{w_i} + b \\ & - \log\big(\sum_{w} \exp(a_{\text{doc}}^w + a_{\text{sen}}^w + \sum_{t=1}^{m} a_t^w + b)\big) \end{aligned} \tag{12}$$

Algorithm 1 Inference Algorithm

```
• Inputs: A corpus containing $D$ documents, $S$ sentences,
  and a vocabulary containing $W$ distinct words
• Initialize parameters
    – Randomly initialize the vectors $\Psi$.
    – Initialize parameters $U$ with all-zero vectors.
    – Initialize Gaussian mixture model parameters with the
      standard normal distribution $\mathcal{N}(\mathbf{0}, \mathrm{diag}(1))$.
• Repeat until converge
    – Fixing parameters $U$ and $\Psi$, run the EM algorithm to
      estimate the Gaussian mixture model parameters $\lambda$.
    – Fixing the Gaussian mixture model $\lambda$, run stochastic
      gradient descent to maximize the log-likelihood of the
      model with respect to parameters $U$ and $\Psi$.
```

where $p(\Psi|\lambda)$ is the prior distribution of parameter Ψ in the Gaussian mixture model, defined by equation (2). The quantities a^w_{doc}, a^w_{sen} and a^w_t are defined in equations (5), (6), and (7). The objective function $J_i(U, \Psi)$ involves all parameters in the collections (U, Ψ). Taking the computation efficiency into consideration, we only update the parameters associated with the word w_i. Concretely, we update

$$\mathrm{vec}(w_i) \leftarrow \mathrm{vec}(w_i) + \alpha \frac{\partial J_i(U, \Psi)}{\partial \mathrm{vec}(w_i)} \tag{13}$$

$$u^{w_i}_t \leftarrow u^{w_i}_t + \alpha \frac{\partial J_i(U, \Psi)}{\partial u^{w_i}_t} \tag{14}$$

with α as the learning rate. Similarly, we update $\mathrm{vec}(s)$, $\mathrm{vec}(d)$ and u^w_{doc}, u^w_{sen} using the same gradient step, as they are parameters associated with the current sentence and the current document. Once the gradient update is accomplished, we sample another location to continue the update. The procedure terminates when there are sufficient number of updates performed, so that both U and Ψ converge to fixed values.

Experiments

In this section, we evaluate our model on the 20 Newsgroups and the Reuters Corpus Volume 1 (RCV1-v2) data sets. Followed the evaluation in (Srivastava, Salakhutdinov, and Hinton 2013), we compare our GMNTM model with the state-of-the-art topic models in perplexity, retrieval quality and classification accuracy.

Datasets description

We adopt two widely used datasets, the 20 Newsgroups data and the RCV1-v2 data, in our evaluations. Data preprocessing is performed on both datasets. We first remove non-alphabet characters, numbers, pronoun, punctuation and stop words from the text. Then, stemming is applied so as to reduce the vocabulary size and settle the issue of data spareness. The detailed properties of the datasets are described as follow.

20 Newsgroups dataset: This dataset is a collection of 18,845 newsgroup documents[1]. The corpus is partitioned into 20 different newsgroups, each corresponding to a separate topic. Following the preprocessing in (Hinton and Salakhutdinov 2009) and (Larochelle and Lauly 2012), the dataset is partitioned chronologically into 11,314 training documents and 7,531 testing documents.

Reuters Corpus Volume 1 (RCV1-v2): This dataset is an archive of 804,414 newswire stories produced by Reuters journalists between August 20, 1996, and August 19, 1997 (Lewis et al. 2004)[2]. RCV1-v2 has been manually categorized into 103 topics, and the topic classes form a tree which is typically of depth 3. As in (Hinton and Salakhutdinov 2009) and (Larochelle and Lauly 2012), the data was randomly split into 794,414 training documents and 10,000 testing documents.

Baseline methods

In the experiments, the proposed topic modeling approach is compared with several baseline methods, which we describe below:

Latent Dirichlet Allocation (LDA): In the LDA model (Blei, Ng, and Jordan 2003), we used the online variational inference implementation of the gensim toolkit [3]. We used the recommended parameter setting $\alpha = 1/T$.

Hidden Topic Markov Models (HMM): This model is proposed by (Gruber, Weiss, and Rosen-Zvi 2007), which models the topics of words in the document as a Markov chain. The HMM model is run using the publicly available code[4]. We use default settings for all hyper parameters.

Over-Replicated Softmax (ORS): This model is proposed by (Srivastava, Salakhutdinov, and Hinton 2013). It is a two hidden layer DBM model, which has been shown to obtain a state-of-the-art performance in terms of classification and retrieval tasks compared with Replicated Softmax model (Hinton and Salakhutdinov 2009) and Cannonade model (Larochelle and Lauly 2012).

Implementation details

In our GMNTM model, the learning rate α is set to 0.025 and gradually reduced to 0.0001. For each word, at most $m = 6$ previous words in the same sentence is used as the context. For easy comparison with other models, the word vector size is set to the same as the number of topics $V = T = 128$. Increasing the word vector size further could improve the quality of the topics that are generated by the GMNTM model.

Documents are split into sentences and words using the NLTK toolkit (Bird 2006)[5]. The Gaussian mixture model is inferred using the variational inference algorithm in scikit-learn toolkit (Pedregosa et al. 2011)[6]. To perform comparable experiments with restricted vocabulary, words outside

[1] Available at http://qwone.com/~jason/20Newsgroups

[2] Available at http://trec.nist.gov/data/reuters/reuters.html

[3] http://radimrehurek.com/gensim/models/remodel.html

[4] http://code.google.com/p/Oppenheimer/downloads/list

[5] http://www.nltk.org/

[6] http://scikit-learn.org/

Data Set	LDA	HTMM	ORS	GMNTM
20 Newsgroups	1068	1013	949	**933**
RCV1-v2	1246	1039	982	**826**

Table 1: Comparison of test perplexity per word with 128 topics

of the vocabulary is replaced as a special token and doesn't count into the word perplexity calculation.

Generative model evaluation

We first evaluate our model's performance as a generative model for documents. We perform our evaluation on the 20 Newsgroups dataset and the RCV1-v2 dataset. For each of the datasets, we extract the words from raw data and preserve the ordering of words. We follow the same evaluation as in (Srivastava, Salakhutdinov, and Hinton 2013), comparing our model with the other models in terms of perplexity.

We estimate the log-probability for 1000 held-out documents that are randomly sampled from the test sets. After running the algorithm to infer the vector representations of words, sentences, and documents in held-out test documents, the average test perplexity per word is then estimated as $\exp\left(-\frac{1}{N}\sum_w \log p(w)\right)$, where N are the total number of words in the held-out test documents, and $p(w)$ is calculated according to equation (4).

Table 1 shows the perplexity for each dataset. The perplexity for Over-Replicated Softmax is taken from (Srivastava, Salakhutdinov, and Hinton 2013). As shown by Table 1, our model performs significantly better than the other models on both datasets in terms of perplexity. More specifically, for 20 Newsgroups data set, the perplexity decreases from 949 to 933, and for RCV1-v2 data set, it decreases from 982 to 826. This verifies the effectiveness of the proposed topic modeling approach in fitting the dataset. The GMNTM model works particularly well on large-scale datasets such as RCV1-v2.

Document retrieval evaluation

To evaluate the quality of the documents representations that are learnt by our model, we perform an information retrieval task. Following the setting in (Srivastava, Salakhutdinov, and Hinton 2013), documents in the training set are used as a database, while the test set is used as queries. For each query, documents in the database are ranked using cosine distance as the similarity metric. The retrieval task is performed separately for each label and the results are averaged. Figure 1 compares the precision-recall curves with 128 topics. The curves for LDA and Over-Replicated are taken from (Srivastava, Salakhutdinov, and Hinton 2013). We see that for the 20 Newsgroups dataset, our model performs on par or slightly better than the other models. While for the RCV1-v2 dataset, our model achieves a significant improvement. Since RCV1-v2 contains a greater amount of texts, the GMNTM model considering the ordering of words is more powerful in mining the semantics of the text.

Data Set	LDA	HTMM	ORS	GMNTM
20 Newsgroups	65.7%	66.5%	66.8%	**73.1%**
RCV1-v2	0.304	0.395	0.401	**0.445**

Table 2: Comparison of classification accuracy on 20 Newsgroups and Mean Precision on Reuters RCV1-v2 with 128 topics

Document classification evaluation

Following the evaluation of (Srivastava, Salakhutdinov, and Hinton 2013), we also perform document classification with the learnt topic representation from our model. The same as in (Srivastava, Salakhutdinov, and Hinton 2013), multinomial logistic regression with a cross entropy loss function is used for the 20 Newsgroups data set, and the evaluation metric is the classification accuracy. For the RCV1-v2 data set, we use independent logistic regression for each label. The evaluation metric is Mean Average Precision.

We summarize the experiment results with 128 topics in Table 3. The results of document classification for LDA and Over-Replicated Softmax are taken from (Srivastava, Salakhutdinov, and Hinton 2013). According to Table 3, the proposed model substantially outperforms other models on both datasets for document classification. For the 20 Newsgroups dataset, the overall accuracy of the Over-Replicated Softmax model is 66.8%, which is slightly higher than LDA and HTMM. Our model further improves the classification result to 73.1%. On RCV1-v2 dataset, we observe the similar results. The mean average precision increases from 0.401 (Over-Replicated Softmax) to 0.445 (our model).

Qualitative inspection of topic specialization

Since topic models are often used for the exploratory analysis of unlabeled text, we also evaluate whether meaningful semantics are captured by our model. Due to the space limit, we only illustrate four topics extracted by our model and LDA which are topics about religion, space, sports and security. These topics are also captured as (sub)categories in the 20 Newsgroups dataset. Table 3 shows the 4 topics learnt by the GMNTM model and the corresponding topics learnt by LDA. In each topic, we visualize it using 10 words with the largest weights. The 4 topics shown in Table 3 for both models are easy for interpretation according to the top words. However, we see that the topics found by the two models are different in nature. GMNTM finds topics that consist of the words that are consecutive in the document or the words having similar semantics. For example, in the GMNTM model, “Christ” and “christian” share the same topics, mainly because they have strong semantic connections, even though they don't co-occur that often, which makes LDA unable to put them in the same topic. On the other hand, LDA often find some general words such as “would” and “accept” for the religion topic, which are unhelpful for interpreting the topics.

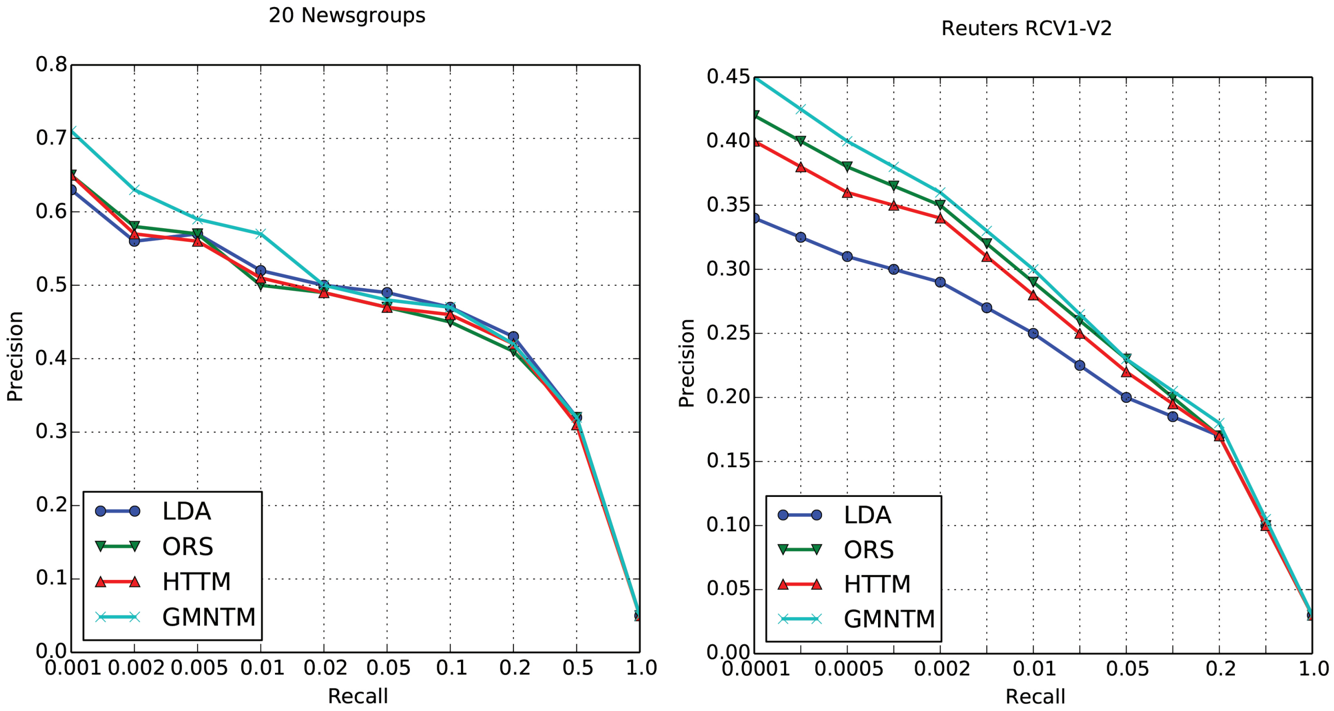

Figure 1: Document Retrieval Evaluation

GMNTM Topic words				LDA Topic Words			
god	space	game	key	god	space	year	key
jesus	orbit	play	public	believe	nasa	hockey	encryption
christ	earth	season	encryption	jesus	research	team	use
believe	solar	team	security	sin	center	division	des
christian	spacecraft	win	escrow	one	shuttle	league	system
bible	surface	hockey	secure	mary	launch	nhl	rsa
lord	planet	hand	data	lord	station	last	public
truth	mission	series	privacy	would	orbit	think	security
sin	satellite	chance	government	christian	april	maria	nsa
faith	shuttle	nhl	nsa	accept	satellite	see	secure

Table 3: Topic words

Conclusion and Future Work

Rather than ignoring the semantics of the words and assuming that the topic distribution within a document is conditionally independent, in this paper, we introduce an ordering-sensitive and semantic-aware topic modeling approach. The GMNTM model jointly learns the topic of documents and the vectorized representation of words, sentences and documents. The model learns better topics and disambiguates words that belong to different topics. Comparing to state-of-the-art topic modeling approaches, the GMNTM outperforms in terms of perplexity, retrieval accuracy and classification accuracy.

In future works, we will explore using non-parametric models to cluster word vectors. For example, we look forward to incoporating infinite Dirichelet process to automatically detect the number of topics. We can also use hierarchical model to further capture the subtle semantics of the text. As another promising direction, we consider building topic models on popular neural probabilistic methods, such as the Recurrent Neural Network Language Model (RNNLM). The GMNTM model has appplications to several tasks in natural language processing, including entity recognition, information extraction and sentiment analysis. These applications also deserve further study,

References

Bird, S. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, 69–72. Association for Computational Linguistics.

Bishop, C. M. 2006. *Pattern recognition and machine learning*, volume 1. springer New York.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3:993–1022.

Blei, D. M. 2012. Probabilistic topic models. *Communications of the ACM* 55(4):77–84.

Florez, O. U., and Nachman, L. 2014. Deep learning of semantic word representations to implement a content-based recommender for the recsys challenge¡¯14.

Griffiths, D., and Tenenbaum, M. 2004. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems* 16:17.

Gruber, A.; Weiss, Y.; and Rosen-Zvi, M. 2007. Hidden topic markov models. In *International Conference on Artificial Intelligence and Statistics*, 163–170.

Hinton, G. E., and Salakhutdinov, R. 2009. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, 1607–1614.

Hofmann, T. 1999. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 289–296. Morgan Kaufmann Publishers Inc.

Larochelle, H., and Lauly, S. 2012. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, 2708–2716.

Le, Q. V., and Mikolov, T. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Lewis, D. D.; Yang, Y.; Rose, T. G.; and Li, F. 2004. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research* 5:361–397.

Lin, C.; He, Y.; Everson, R.; and Ruger, S. 2012. Weakly supervised joint sentiment-topic detection from text. *CIKM* 24(6):1134–1145.

Marlin, B. M. 2003. Modeling user rating profiles for collaborative filtering. In *Advances in neural information processing systems*.

Mcauliffe, J. D., and Blei, D. M. 2008. Supervised topic models. In *Advances in neural information processing systems*, 121–128.

Mei, Q.; Liu, C.; Su, H.; and Zhai, C. 2006. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th international conference on World Wide Web*, 533–542. ACM.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mnih, A., and Hinton, G. E. 2009. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, 1081–1088.

Mnih, A., and Kavukcuoglu, K. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, 2265–2273.

Mnih, A., and Teh, Y. W. 2012. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; and Dubourg, V. 2011. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research* 12:2825–2830.

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.

Srivastava, N.; Salakhutdinov, R. R.; and Hinton, G. E. 2013. Modeling documents with deep boltzmann machines. *arXiv preprint arXiv:1309.6865*.

Steyvers, M., and Griffiths, T. 2007. Probabilistic topic models. *Handbook of latent semantic analysis* 427(7):424–440.

Steyvers, M.; Smyth, P.; Rosen-Zvi, M.; and Griffiths, T. 2004. Probabilistic author-topic models for information discovery. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 306–315. ACM.

Teh, Y. W.; Jordan, M. I.; Beal, M. J.; and Blei, D. M. 2006. Hierarchical dirichlet processes. *Journal of the american statistical association* 101(476).

Wallach, H. M. 2006. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, 977–984. ACM.

Wan, L.; Zhu, L.; and Fergus, R. 2012. A hybrid neural network-latent topic model. In *International Conference on Artificial Intelligence and Statistics*, 1287–1294.

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

Wei, X., and Croft, W. B. 2006. 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*, 178–185. ACM.

Yang, M.; Peng, B.; Chen, Z.; Zhu, D.; and Chow, K.-P. 2014a. A topic model for building fine-grained domain-specific emotion lexicon. *ACL 2014* 421–426.

Yang, M.; Zhu, D.; Mustafa, R.; and Chow, K.-P. 2014b. Learning domain-specific sentiment lexicon with supervised sentiment-aware lda. *ECAI 2014* 927–932.