International Journal of Networked and Distributed Computing, Vol. 4, No. 2 (April 2016), 127-136

Hierarchical Latent Semantic Mapping for Automated Topic Generation

Guorui Zhou [1] , Guang Chen [2]

[1] *School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, No. 10 Xi Tu Cheng Road, Beijing, BJ 10, China*

E-mail: zhouguorui@bupt.edu.cn

[2] *School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, No. 10 Xi Tu Cheng Road, Beijing, BJ 10, China*

E-mail: chenguang@bupt.edu.cn

Abstract

Much of information sits in an unprecedented amount of text data. Managing allocation of these large scale text data is an important problem for many areas. Topic modeling performs well in this problem. The traditional generative models (PLSA,LDA) are the state-of-the-art approaches in topic modeling and most recent research on topic generation has been focusing on improving or extending these models. However, results of traditional generative models are sensitive to the number of topics K, which must be specified manually and determines the rank of solution space for topic generation. The problem of generating topics from corpus resembles community detection in networks. Many effective algorithms can automatically detect communities from networks without a manually specified number of the communities. Inspired by these algorithms, in this paper, we propose a novel method named Hierarchical Latent Semantic Mapping (HLSM), which automatically generates topics from corpus. HLSM calculates the association between each pair of words in the latent topic space, then constructs a unipartite network of words with this association and hierarchically generates topics from this network. We apply HLSM to several document collections and the experimental comparisons against several state-of-the-art approaches demonstrate the promising performance.

Keywords: Topic modeling, Network, LDA, Unsupervised learning

1. Introduction

Managing large allocation of documents has become a popular challenge in many fields. Topic modeling, which assigns topics to documents, offers a promising solution for this challenge.

Topic models generate topics from a set of documents and assign topics to these documents. Based on these topics we can solve problems on cross-domain text classification[1,2], understanding text clustering[3,4], text recommendation[5], and other related text data applications[6]. There has been an exceptional amount of research on topic-model algorithms. Although there exists extraordinary research on topic-model, most of them focus on generative models underlying PLSI [7] and LDA [8].

PLSA and LDA are highly modular and can therefore be easily extended. PLSA model assumes

the topics of each document follow a multinomial distribution and treats each topic as a multinomial distribution over the words. LDA model proposed a Dirichlet prior for the topic distributions of the documents and a Dirichlet prior for the words distributions of the topics. The LDA model is essentially the Bayesian version of PLSA model.

Since LDA's introduction, there is much research based on it. The Correlated Topic Model Advances [9] follows this approach, inducing a correlation structure between topics by using the logistic normal distribution instead of the Dirichlet. Another extension is the hierarchical LDA [10], where topics are joined together in a hierarchy by using the nested Chinese restaurant process. [11] explores several classes of structured priors for topic models, and find that an asymmetric Dirichlet prior over the document-topic distributions has substantial advantages over a symmetric prior, while an asymmetric prior over the topic-word distributions provides no real benefit.

The generative models allow sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. In this problem, observations are the form of co-occurrences of words and documents. Generative models estimate the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions ($p(w|t)$ and $p(d|t)$). For both LDA and PLSI, the optimization goal is to find the global maximum of a likelihood function. Much study of disordered systems in physics has been focusing on this optimization problem too [12]. If we make this problem simpler, in which one word belong to one topic, then topic modeling will be similar to the problem of fitting stochastic block models to complex networks [13,14].

A research on the validity of LDA optimization algorithms for inferring topic models proposes that current implementations of LDA have low validity [15]. They conduct a controlled analysis of topic-model algorithms for highly specified sets of synthetic data, and there analysis reveals that standard techniques for likelihood optimization are significantly hindered by the roughness of the likelihood-function landscape. Their paper also proposes a simple network approach to topic modeling named Topic Mapping. TopicMapping constructs a unipartite network of words by connecting words with their co-occurrence, then clusters the words. However the unipartite network in Topic Mapping is simple and not closely related to the problem of topic generation, and TopicMapping treats every cluster of the words as a topic, which is rough.

In this paper we propose a novel approach named Hierarchical Latent Semantic Mapping (HLSM) for topic generation. HLSM calculates the similarity of each pair of words for latent topics based on the Singular Value Decomposition (SVD) of the co-occurrences of words and documents. Then constructs a unipartite network of words with this similarity and hierarchically separates the network into clusters using the Hierarchical Map Equation algorithm [16] . Every cluster can be seem as a topic, then refine these initial topics using a PLSA-like likelihood optimization.

The contribution of this paper can be summarized as follows:

- Propose a novel approach to constructing network of words closely related to the latent topic space.
- Adapt approaches from community detection in networks to initial hierarchical topic generation, and also propose a method to further refine the topics.
- To evaluate the effectiveness of the proposed approach, we conducted experiments on several real-world text data sets. The experimental results demonstrate that our approach provides greatly improvements in terms of documents classification.

2. Problem of standard topic-model algorithms

The core assumption of standard topic-model algorithms is that a corpus consisted of N documents. And each document is generated by the processing selecting one topic from K topics with probability $p(topic|doc)$ then selecting one word from N_w distinct words with probability $p(word|topic)$. Then, our problem is translated to estimate NK probabilities $p(topic|doc)$ and KN_w probabilities p(word—topic). LDA and PLSI both aim to estimate the values of these probabilities with the

highest likelihood of generating the corpus [7,8,17,18]. Thus, the inference problem is transformed to an optimization problem [19]. But there exist many competing models with nearly identical likelihoods. Due to the high degeneracy of the likelihood landscape, standard optimization algorithms will more likely infer different models after different optimization runs than infer the model with the highest likelihood,as has been previously reported [19,11].

Meanwhile, selecting the number of topics K is one of the most problematic modeling choices in finite topic modeling. There is no effective method for choosing K or evaluating the probability of held-out data for various values of K so far. And degree to which LDA is robust to a poor setting of K is not well-understood [11]. Ideally, if LDA has sufficient topics to model the data set well, an increase in K would not have a impact on the assignments of tokens to topics –i.e., the additional topics should be used with low frequency. For example, if twenty topics is adequacy to exactly model the data, then inferred topic assignments would not be significantly affected by increasing the number of topics to fifty. If this is the case, using large K would not have a improvement on the inference. In another words, we still need a robust K. Actually, K could be seem as the rank of the solution space for topic generation. Setting K is same as manually selecting the rank of the solution space, which is obviously not reasonable.

If we think about a easy problem, in which one word can only belongs to one topic. Generating topics from corpus closely approximates to the processing of community detection in networks. A substantial amount of work in the area of community detection in networks has proposed effective algorithms to reveal the struct of the network only using the original information of the network without other prior knowledge. So we create a network of words in the corpus and detecting the communities of the network as the initial guess for topics, then refine these coarse topics.

3. Hierarchical Latent Semantic Mapping

Hierarchical Latent Semantic Mapping (HLSM) is a network approach to topic modeling. Similar to the well-known topic models, each document is represented as a mixture over latent topics. The key feature that distinguishes the HLSM model from the existing topic models is that HLSM directly clusters words and defines each cluster as a topic, then refines these initial topics, thus HLSM estimates the probability distributions $p(word|topic)$ in a novel process.

The HLSM model infers topics as the following steps:

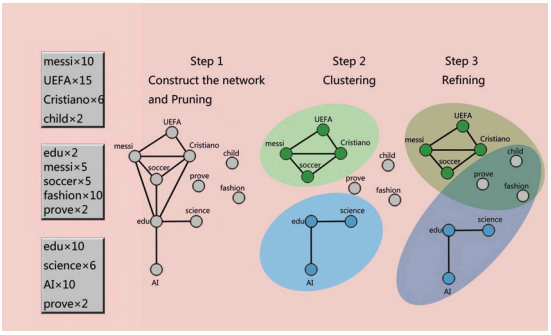

Fig. 1. Illustration of the HLSM algorithm.

step 1. *Construct the unipartite network*.we calculate the association between each pair of words that co-occur in at least one document. Then we construct the unipartite network in which words are connected with the association above the threshold.

step 2. *Clustering of words hierarchically*.The words in the unipartite network are connected by the association in the latent topic space. Naturally we suppose that topics in the corpus will give rise to communities of words in the network. Thus we use the *Hierarchical Map Equation* [16] to detect the communities. And in most of corpus, topics come in the form of multiple levels of abstraction. Abstract topic consists of several concrete topics. Thus we detect some massive communities corresponding to the abstract topics, then we detect minor communities, which correspond to the concrete

topics, from the massive communities. We take the communities as a prior guess for the number of topics and word composition of each of the topics used to generate the documents.*It is worth noting* that we do not set the number of levels and the number of communities for each level. Hierarchical Map Equation can reveal the multilevel organization in the network of words automatically.

step 3. *Refine the prior guess*. After the *last* level of clustering of words, one may get some single communities of words, and in the step 2, one may get some single words not in the network. Thus the prior topics detected in step 2 are rough, we refine the topics using a PLSA-like likelihood optimization.

3.1. Construct the unipartite network

The association between words must be closely related to the topics to ensure the validity of clustering words based on this network. But the topics are latent, and all observations are the words collected into documents. If we assigns topics to documents artificially with prior human knowledge, one can observe that documents share the same topics also are more likely to share some words. Naturally we can believe that the words co-occur in many documents share the same topic, in another word these words are more similar in the latent topic space. To calculate the association between words in the latent topic space. Like the core idea of Latent Semantic Analysis (LSI), we map words to a vector space of reduced dimensionality based on a *Singular Value Decomposition* (SVD) of the co-occurrence matrix M, which *each row i corresponds to a word, each column j to a document* in which the word appeared, and each matrix entry M_{ij} corresponds to the number of occurrences of word i in document j.

Starting with the standard SVD given by

$$M = U\Sigma V^t, \tag{1}$$

the diagonal matrix Σ contains the singular values of M. The approximation of M is computed by setting all but the largest K singular values in Σ to zero ($= \widetilde{\Sigma}$), which is rank K optimal in the sense of the L_2-matrix norm.

One obtains the approximation

$$\widetilde{M} = U\widetilde{\Sigma}V^t \approx U\Sigma V^t = M, \tag{2}$$

The corresponding low-dimensional latent vectors will typically not be sparse, while the original high-dimensional Matrix M is sparse. This implies that one can calculate meaningful association values between pairs of words in the latent topic space. In HLSM, we calculate the cosine similarity between the rows of $U\widetilde{\Sigma}$ as the association of each pair of words in the latent topic space, and connects word i and j with this association $S(i,j)$:

$$W = U\widetilde{\Sigma}, S(i,j) = \frac{\langle W_i \cdot W_j \rangle}{\|W_i\| \cdot \|W_j\|}.$$

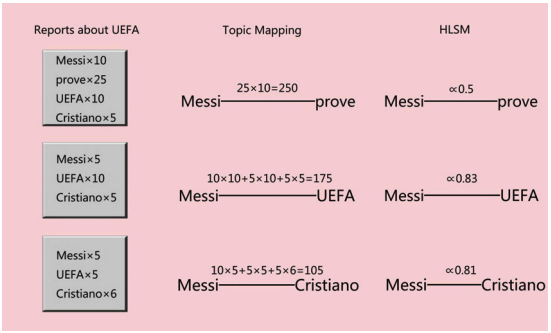

Fig. 2. The connections in HLSM and TopicMapping.

As shown in in Fig. 2, for the reports about the UEFA, the value of connection between "Messi" and "prove" in Topic Mapping is higher than the connection between "Messi" and "UEFA" or the connection between "Messi" and "Cristiano". Actually, for topics generation, we want the value of connection between "Messi" and "UEFA" for documents about the UEFA higher than other connections. In HLSM the value of connections are more valid than Topic Mapping for topics generation.

After calculating all the values of connections. Suppose that the association values between some pair of words are so low that we presume these connections are noise. One can set a threshold of q to purne the connections lower than q.

3.2. *Clustering words hierarchically*

In most of corpus, the structure of topics is not simple and always can be multiple levels. Some concrete topics sit under a same abstract topic. For example, words in a corpus focusing on "soccer" might be drawn from the topics "stars", "matches", "history of soccer", etc.

We construct the network of words based on the association between words in the latent topic space. If the original structure of topics is multiple levels, the network should also have a multilevel structure. To reveal communities at multiple levels, we choose the *Hierarchical Map Equation* [16]. It is worth noting that we do not set the number of levels and the number of communities for each level. Instead Hierarchical Map Equation can reveal the multilevel organization in the network of words automatically.

The Map Equation proposed the duality between finding community structure in networks and minimizing the specification length of a random walker's movements on a network. For a given network partition, the map equation definiens the limit $L(M)$ of how laconic one can describe the trajectory of this random walk in theory.

The core idea of map equation is that if the random walker tends to stay in some blocks of the network for a long time, the code used for specification can be compressed. Therefore, when the proxy for real flow random walk in the network, estimating the minimum map equation over all possible network partitions could reveals the structure of the network with respect to the dynamics on the network.

In our problem, for a hierarchical network M of n nodes, each node corresponds to one word, segmentated into m modules. There is a a submap M^i with m^i submodules in one modules. Correspondingly, there is a submap M^{ij} with m^{ij} submodules in each each submodule ij, and so on.

The corresponding hierarchical map equation is

$$L(M) = q_{switch}H(Q) + \sum_{i=1}^{m} L(M^i) \tag{3}$$

with the specification length of submap M^i at intermediary levels given by

$$L(M^i) = q^i{}_{switch}H(Q^i) + \sum_{j=1}^{m^i} L(M^{ij}) \tag{4}$$

and at the final modular level by

$$L(M^{ij\ldots k}) = p^{ij\ldots k}{}_{in}H(P^{ij\ldots k}) \tag{5}$$

Weight of codebook depends on the rate of use of it, and $L(M)$ is the sum of average length of codewords for each codebook. $H(Q)$ is the average length of codewords in the index codebook according to the rate of use of it, while the entropy terms depends on the rate at which the codebooks are used. On any given step the random walker switches the *first* level modules at probability of q_{switch}, while q_{switch} is the rate of index codebook is used.

At each submodule level, $H(Q^i)$ is the average length of the codewords according to the using rate in the subindex codebook and q^i_{switch} is the rate of codeword use for entering the m_i submodules or exiting to a higher level. At the last level, $H(P^{ij\ldots k})$ is the average length of the codewords according to the using rate in the submodule codebook and $p^{ij\ldots k}_{in}$ is the rate of codeword use for visiting nodes in submodules $ij\ldots k$ or exiting to other submodules. The problem of seeking the hierarchical structure that best represents the structure is translated to finding the hierarchical partition of the network with the minimum map equation. Fig.3 illustrates an example for map equation.

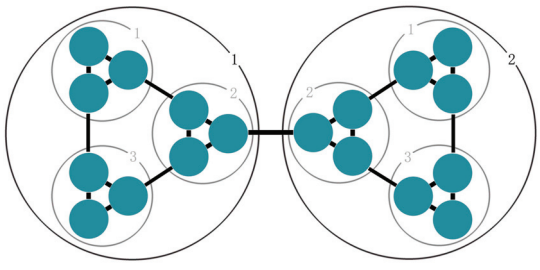

Fig. 3. Example for Minimizing the map equation over all network partitions gives an optimal clustering of the network with respect to the dynamics on the network.

In this example we can assume that all weights for connections in the network are equal, thus all

rates can be calculated by counting links and normalizing. The specification length for an unpartitioned network is $-log_2(1/18) = 4.17bits$. After the network is partitioned, the codewords of the first level modules are used at a total rate $q_{switch} = \frac{2}{50}$ (There are 25 lines in the network and 50 possible moves when considering direction, while only 2 moves can switch between the first level module.), while relative rates $Q = \frac{1}{2}, \frac{1}{2}$. And $Q^1 = \frac{2}{8}, \frac{2}{8}, \frac{3}{8}, \frac{1}{8}$, noticing that there is a rate at $\frac{1}{8}$ random walker existing to Module 2, while q^1_{switch} is $\frac{8}{50}$. Thus $L(M)$ is:

$$L(M) = q_{switch}H(Q) + \begin{cases} q^1_{switch}H(Q^1) + \begin{cases} p^{11}_{in}H(P^{11}) \\ p^{12}_{in}H(P^{12}) \\ p^{13}_{in}H(P^{13}) \end{cases} \\ q^2_{switch}H(Q^2) + \begin{cases} p^{21}_{in}H(P^{21}) \\ p^{22}_{in}H(P^{22}) \\ p^{23}_{in}H(P^{23}) \end{cases} \end{cases}$$

L(M) = 0.04 bits + 0.61bits + 2.54 bits = 3.19 bits .

3.3. *Refine the prior guess*

Once the network is built, we detect clusters (same as the modules detected by *Hierarchical Map Equation*) of highly associated words using the *Hierarchical Map Equation*. After the *last* level of clustering, we get a hard partition of words, meaning that words can only belong to a single cluster. Actually a word may have multiple senses and multiple types of usage in different context. Consequently if we simply define every cluster as a topic, these rough topics can not provide a reasonable probabilistic interpretation of the corpus in terms of the latent topic space. Therefore we propose a method to further refine these rough topics.

We now discuss how we can compute the distributions $p(topic|doc)$ and $p(word|topic)$, given a partition of words. In the prior partition of words, we define every cluster as a topic. In fact, each word in the network can sit in only one module after the *Hierarchical Map Equation* processing. Therefore, $p(t|w) = \delta_{t,w}$. $\delta_{t,w} = 1$ only if the word w sits in the module, which corresponds to the topic t . For other topics $\bar{t}$, $\delta_{\bar{t},w} = 0$. Noticing that in this step word w can only belongs to one topic t, so $p(w,t) = p(w)$, thus:

$$p(w|t) = \frac{p(w)}{\sum_w p(w) \times \delta_{t,w}} \text{ and } p(t|d) = \frac{1}{L_d}\sum_w w^d_w \delta_{t,w}. \tag{6}$$

L_d is the number of words in document d, w^d_w is the number of times word w occurs in the document d. It is also useful to introduce $n(w,t) = L_C \times p(w,t)$, which is the number of times topic t was chosen and word w was drawn.L_C is the number of the words in the corpus. So far, the PLSA-like likelihood of our model is:

$$\begin{aligned} L = log(\prod_{w,d} p(w,d)) &= log(\prod_{w,d}\sum_t p(w|t)p(t|d)) \\ &= \sum_d \sum_w w^d_w \times log(\sum_t p(w|t)p(t|d)) \ . \end{aligned} \tag{7}$$

We can improve this likelihood by simply making documents more specific to fewer topics. For that our optimization algorithm simply finds, for each document, words assigned with some infrequent topics and reassigns the most significant topic in that document to these words.

1. For each document d, we find the most *significant* topic t_s with the smallest p-value, considering a null model where each word is independently sampled from topic t with probability $p(t) = \sum_w p(w)p(t|w)$. Calling x the number of words which actually come from topic t, ($x = L_d \times p(t|d)$, see Eq . (6)) , the p-value of topic t is then computed using a binomial distribution, $B(x; Ld, p(t))$. Obviously p-value represents the significance of the word better than x, which only depends on the $p(t|d)$.

2. For each document d, recall that after the step 2 we may get some single words not in the network. We simply assign these words to the most significant topic t_s and we can calculate a baseline of the PLSA-like likelihood L(see Eq .(7)).

3. For each document d, we define the *infrequent* topics t_{in} simply as those which occur with probability smaller than a parameter: $p(t_{in}|d) < \eta$. We assign the most significant topic t_s to the words which belong to the all infrequent topics t_in. The

$p(t_s|d)$ will be incremented by the sum of all $p(t_{in}|d)$, while all $p(t_{in}|d)$ are set to zero. Similarly, $n(w,t_{in})$(see above) will be decreased by w_w^d for each word w which belongs to an infrequent topic, and $n(w,t_s)$ is increased accordingly.

4. After previous step for all document, we compute:

$$p(w|t) = \frac{n(w,t)}{\sum_w n(w,t)} \tag{8}$$

and the likelihood of model, L_η, where we made explicit its dependency on η. We pick the model with maximum L_η by looping over all possible values of η (from 0% to 50% with steps of 1%).

HLSM estimates the probabilities $p(w|t)$ and $p(t) = \sum_w p(w)p(t|w)$ from training data set, and calculates $p(t|w) = \frac{p(t)p(w|t)}{p(w)}$, for a new document from held out data set, $p(w|t)$ won't be changed, $p(t|d)$ can be calculated by :

$$p(t|d) = \frac{\sum_w p(t|w)}{L_d} \tag{9}$$

HLSM fixed the probabilities $p(w|t)$ and $p(t)$ after the training process, and hence is plagued by overfitting. It will be a shortcoming of the HLSM model, when the scale of the training data set is small.

4. Experimental Evaluations

HLSM is a topic model towards collections of text corpora. It can be applied to lots of applications such as classifying, clustering, filtering, information retrieval and related areas. Follow Blei's idea [8], in this section, we investigate two important applications: document modeling and document classification.

4.1. Document Modeling

The goal of document modeling is to generalize the trained model from the training dataset to a new dataset. The documents in the corpora are unlabeled, our goal is density estimation, thus we wish to obtain high likelihood on a held-out test set. In particular, we computed the *perplexity* of a held-out test set to evaluate the models. Models which yield a lower perplexity are considered to achieve a better generalization performance because the model is less surprised by a portion of the datasets which the model have never seen before. Formally, for a test set of M documents, the perplexity is defined as:

$$perplexity(D_{test}) = exp\left\{\frac{-\sum_{i=1}^{M} logp(d_i)}{\sum_{i=1}^{M} L_i}\right\} \tag{10}$$

Table 1. Data Sets Generated from 20Newsgroups.

Data set	Domain
Comp and Sci	comp.graphics comp.sys.mac.hardware sci.crypt sci.med
Comp and Talk	comp.os.ms-windows.misc comp.sys.ibm.pc.hardware talk.politics.mideast talk.politics.misc
Comp and Rec	comp.graphics comp.sys.ibm.pc.hardware rec.motorcycles rec.sport.baseball
Sci and Rec	sci.crypt sci.med rec.autos rec.sport.baseball
Talk and Rec	talk.politics.mideast talk.politics.misc rec.autos rec.sport.baseball
Talk and Sci	talk.politics.misc talk.religion.misc sci.crypt sci.med

We conduct this experiment on a subset of the 20Newsgroups data set, which has been widely used for evaluating the performance of cross-domain text classification algorithms. It contains nearly 20,000 newsgroup documents which have been evenly partitioned into 20 different newsgroups. We chose 3878 documents (we filtered some little documents) from domain comp.graphics, com.sys.mac.hardware, sci.crypt, and sci.med as our dataset used in the evaluation. We held out 20%

of the corpus for test purpose and trained the models on the remaining 80%. In data preprocessing, we removed 163 stop words in standard list and the words occurrences less than 3 times from each corpus. We compare HLSM against PLSA, asymmetric LDA and TopicMapping. The initial α for asymmetric LDA was set to 0.01 for all topics.

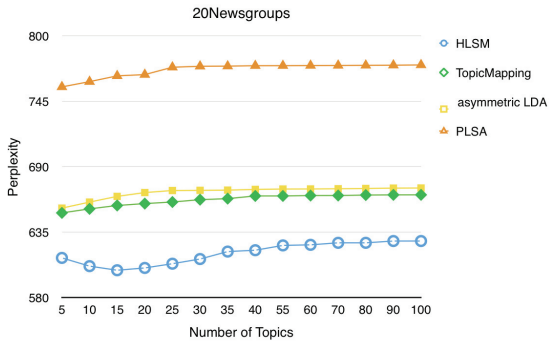

Fig. 4. Perplexity comparisons on the 20Newsgroups dataset.

Fig. 4 shows the perplexity results where the number of the topics varies from 5 to 100. As can be seen, the HLSM model achieves slight improvement in terms of perplexity, while TopicMapping is close to asymmetric LDA. Experiment shows that the prior guess of HLSM makes great difference on the topic generation.

Table 3 presents the examples of top 12 extracted topics on data set *Comp and Sci*, some topics with lower probability were not exhibited. We sorted the words with the learned topic-word probability. By examining the topical words, we can observe that the words in the same topic are always semantically relevant. For example, Topic 1 is about Mac hardware, and one domain in the data set Comp and Sci is comp.sys.mac.hardware, respectively. It is noteworthy that, some topics look similar in abstract level, but there are still some distinctions between them. For instance, words in Topic 2 and Topic 4 are semantically relevant but Topic 2 is more related to medical treatment, while Topic 4 probably describes some reports about disease. The result shows that our method can effectively identify the correlations between domain-specific features from different domains. Furthermore, our method can extracted narrow topics under the level of domain. And we conduct the next experiment on the whole 20Newsgroups data set.

4.2. *Document classification*

In the text classification problem, topic models are wished to classify a document into two or more mutually exclusive classes. The choice of features is a challenging aspect of the document classification problem. By representing the documents in terms of latent topic space, the topic models can generate the probabilities $p(t|d)$. If one use the vector of $p(t|d)$ as the feature of documents to fix the text classification problem, the probabilities vector generated by the most effective model can perform better than the probabilities vector generated by other models.

To test the effectiveness of HLSM, we compared it with the following representative topic models.

1. PLSA
2. symmetric LDA
3. asymmetric LDA
4. TopicMapping

We generated six cross-domain text data sets from 20Newsgroups by utilizing its labeled structure. There are 4 fields in each data set, Table 1 summarizes the data sets generated from 20Newsgroups. To make the classification problem more effective and convincing, the task was defined as a multi-label classification.

In these experiments, we estimated the probabilities $p(t|d)$ using the above topic models on all the documents of each data sets, and used the vector of probabilities $p(t|d)$ as the only features to train a support vector machine (SVM) for multi-label classification. For each data set, 20% of the documents were held out as the test data and we trained a SVM for multi-label classification with the remaining 80% labeled documents. We used these classifiers to predict the class labels of unlabeled documents in the test data. Notice that there were 4 field in each data set, the classification process was considered as correct only if the document was classified into the original field.

Table 2. The Test Classification Accuracy on The Data Sets Generated from 20Newsgroups.

Data set	PLSA	LDA	asymmetric LDA	TopicMapping	HLSM
Comp and Sci	0.761	0.771	0.792	0.831	**0.855**
Comp and Talk	0.785	0.790	0.813	0.846	**0.871**
Comp and Rec	0.770	0.776	0.781	0.834	**0.853**
Sci and Rec	0.724	0.723	0.767	0.803	**0.822**
Talk and Rec	0.811	0.802	0.832	0.821	**0.876**
Talk and Sci	0.804	0.811	0.839	0.847	**0.867**
Average	0.766	0.779	0.804	0.834	**0.857**

Table 3. Examples of Top 12 Topics Extracted by HLSM on Data Set Comp and Sci.

topic: 1 $p(t)$: 0.0801275	topic: 2 $p(t)$: 0.067205	topic: 3 $p(t)$: 0.0661541	topic: 4 $p(t)$: 0.0619122	topic: 5 $p(t)$: 0.0606786	topic: 6 $p(t)$: 0.0600079
mac	doctor	clipper	medic	food	imag
doe	patient	phone	health	msg	jpeg
system	vitamin	chip	1993	diet	file
speed	medic	encrypt	diseas	eat	format
price	candida	govern	hiv	weight	gif
hardware	treatment	onli	report	effect	program
topic: 7 $p(t)$: 0.0561931	topic: 8 $p(t)$: 0.0557023	topic: 9 $p(t)$: 0.0507203	topic: 10 $p(t)$: 0.0462584	topic: 11 $p(t)$: 0.0454941	topic: 12 $p(t)$: 0.0440686
imag	drive	key	anonym	nsa	3d
data	disk	encrypt	email	writes	graphic
system	system	messag	internet	govern	file
packag	work	secur	post	articl	object
sourc	scsi	pgp	comput	david	ray
code	machin	attack	inform	trust	model

We did the same data preprocessing as above, and the number of topics in each data set for LDA, PLSA, and asymmetric LDA was set to 4. Table 2 summarizes the classification performance on each data set, the first three row shows the best accuracy while the number of topics for LDA, PLSA, and asymmetric LDA varies. The last row of the table shows the average accuracy over all data sets. From the table we can observe that HLSM outperformed all other topic models on six data sets.

Table 3 presents the examples of top 12 extracted topics on data set Comp and Sci, there remained some topics with lower probability were not exhibited. We sorted the words with the learned topic-word probability. By examining the topical words, we can observe that the words in the same topic are always semantically relevant. For example, Topic 1 is about Mac hardware, and one domain in the data set Comp and Sci is comp.sys.mac.hardware, respectively. It is noteworthy that, some topics look similar, but they have distinction. For instance, words in Topic 2 and Topic 4 are semantically relevant but Topic 1 is more related to medical treatment, while Topic 2 probably describing some reports about disease. The result shows that our method can effectively identify the correlations between domain-specific features from different domains. Furthermore, our method can extracted narrow topics under the level of domain.

5. Conclusion

A topic model HLSM is presented in this paper to apply an approach from the area of community detection to topic generation. We apply the HLSM model to several document collections for document modeling and document clustering, and the experimental comparisons against state-of-the-art approaches demonstrate the promising performance. Especially the examples of words with top probability $p(word|topic)$ prove that topics generated by HLSM could be distinguished at a fine level.

Our work did not focus on the idea of the standard topic-model algorithms, which try to generated topics in a solution space with manually specified *rank*. HLSM automatically generates topics by revealing the struct of the network consists of words in the corpus. Particularly, plenty of work in the area of community detection focus on stochastic block models, which tries to reveal community structure in networks. We believe this work, which is similar to topic model in spirit, would offer new insights into topic modeling.

References

1. B. Barathi, "Cross-domain text classification using semantic based approach," in *Sustainable Energy and Intelligent Systems (SEISCON 2011), International Conference on*, July 2011, pp. 820–825.
2. R. Zhao and K. Mao, "Supervised adaptive-transfer plsa for cross-domain text classification," in *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*, Dec 2014, pp. 259–266.
3. A. Ghazifard, Z. Shamaee, and M. Shams, "Topic word set-based text clustering," in *e-Commerce in Developing Countries: With Focus on e-Security (ECDC), 2013 7th Intenational Conference on*, April 2013, pp. 1–10.
4. H.-C. Chang and C.-C. Hsu, "Using topic keyword clusters for automatic document clustering," in *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*, vol. 1, July 2005, pp. 419–424 vol.1.
5. J. Wilson, S. Chaudhury, and B. Lall, "Improving collaborative filtering based recommenders using topic modelling," in *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, vol. 1, Aug 2014, pp. 340–346.
6. J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008. [Online]. Available: http://dx.doi.org/10.1007/s11263-007-0122-4
7. T. Hofmann, "Probabilistic latent semantic indexing," *SIGIR*, pp. 50–57, 1999.
8. A. Y. Blei, D. M.; Ng and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, pp. 993–1022, 2003.
9. D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 113–120. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143859
10. D. M. Blei, T. L. Griffiths, and M. I. Jordan, "The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies," *J. ACM*, vol. 57, no. 2, pp. 7:1–7:30, Feb. 2010. [Online]. Available: http://doi.acm.org/10.1145/1667053.1667056
11. H. M. Wallach, D. M. Mimno, and A. McCallum, "Rethinking lda: Why priors matter," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1973–1981. [Online]. Available: http://papers.nips.cc/paper/3854-rethinking-lda-why-priors-matter.pdf
12. O. C. Martin, R. Monasson, and R. Zecchina, "Statistical mechanics methods and phase transitions in optimization problems," *Theoretical Computer Science*, vol. 265, no. 12, pp. 3–67, 2001, phase Transitions in Combinatorial Problems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397501001499
13. S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 35, pp. 75–174, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0370157309002841
14. B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," *Phys. Rev. E*, vol. 83, p. 016107, Jan 2011. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.83.016107
15. A. Lancichinetti, M. I. Sirer, J. X. Wang, D. Acuna, K. Körding, and L. A. N. Amaral, "High-reproducibility and high-accuracy method for automated topic classification," *Phys. Rev. X*, vol. 5, p. 011007, Jan 2015. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevX.5.011007
16. M. Rosvall and C. T. Bergstrom, "Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems," *PLoS ONE*, vol. 6, no. 4, p. e18209, 04 2011. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0018209
17. T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004. [Online]. Available: http://www.pnas.org/content/101/suppl_1/5228.abstract
18. R. Nallapati, W. Cohen, and J. Lafferty, "Parallelized variational em for latent dirichlet allocation: An experimental evaluation of speed and scalability," in *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, Oct 2007, pp. 349–354.
19. D. Blei, L. Carin, and D. Dunson, "Probabilistic topic models," *Signal Processing Magazine, IEEE*, vol. 27, no. 6, pp. 55–65, Nov 2010.