Learning Latent Topics from the Word Co-occurrence Network

Wu Wang[1,3], Houquan Zhou[1], Kun He[1,2(✉)], and John E. Hopcroft[2]

[1] Huazhong University of Science and Technology, Wuhan 430074, China
{U201110084,U201417183,brooklet60}@hust.edu.cn
[2] Computer Science Department, Cornell University, Ithaca, NY 14853, USA
jeh@cs.cornell.edu
[3] Shenzhen Research Institute of Huazhong University of Science and Technology, Shenzhen 518057, China

Abstract. Topic modeling is widely used to uncover the latent thematic structure in corpora. Based on the separability assumption, the spectral method focuses on the word co-occurrence patterns at the document-level and it includes two steps: anchor selection and topic recovery. Biterm Topic Model (BTM) utilizes the word co-occurrence patterns in the whole corpus. Inspired by the word-pair pattern in BTM, we build a Word Co-occurrence Network (WCN) where nodes correspond to words and weights of edges stand for the empirical co-occurrence probability of word pairs. We exploit existing methods to deal with the word co-occurrence network for anchor selection. We find a K-clique in the unweighted complementary graph, or the maximum edge-weight clique in the weighted complementary graph for the anchor word selection. Experiments on real-world corpora evaluated on topic quality and interpretability demonstrate the effectiveness of the proposed approach.

Keywords: Topic model · Word co-occurrence network · Maximum edge-weight clique · K-clique

1 Introduction

Topic modeling is an important technique used for text mining. The main idea of topic modeling is that documents arise as a distribution on a small number of topic vectors, where each topic vector is a distribution on words. Topic modeling can uncover the thematic structure from a large collection of text documents without human supervision. Topic models are of high importance in many machine learning applications, such as document analysis, classification and clustering, and it plays a key role on some specific context tasks. Topic modeling over short texts has attracted a lot of interests as short texts are prevalent on web application [15,17,18,26–28,30]. For a time-stamped document collection, discovering the evolution of topics over time has drawn much attention as well [4,14,29]. In practice, Targeted Topic Modeling (TTM) for focused analysis can perform more detailed analysis on some specific aspects [25]. The Web

D. Du et al. (Eds.): NCTCS 2017, CCIS 768, pp. 18–30, 2017.
https://doi.org/10.1007/978-981-10-6893-5_2

Search Stream Model (WSSM), a novel and highly practical probabilistic topic model, is delicately calibrated for handling two salient features of the web search data [13].

Existing topic modeling approaches can be divided into two main categories: probabilistic models [6,12] and spectral methods [2,3,21]. The first is based on some basic probabilistic models, such as Probabilistic Latent Semantic Indexing (PLSI) [12] and Latent Dirichlet Allocation (LDA) [6]. The LDA model is the most popular and frequently-used topic modeling method. As it is intractable to learn the parameters directly, the LDA model uses approximation inference techniques such as Markov Chain Monte Carlo (MCMC) [11] and variational inference [6] to learn the parameters. A number of variations of LDA have been studied. The Correlated Topic Model (CTM) [5] exhibits correlation of topics via the logistic normal distribution. Gibbs sampling [7,16] and parallel inference [8,19,28] have been widely studied to improve the performance and scalability.

The second category, spectral methods, suggests an algebraic recovery perspective and utilizes nonnegative matrix factorization (NMF) as a main technique. By making some reasonable assumption, these methods are able to provide provable polynomial-time algorithms. Instead of doing inference on document-word matrix, which is very sparse, they focus on the word co-occurrence matrix. Arora et al. [3] propose an approach to recover the topic distribution assuming that every topic contains at least one anchor word which occurs with non-zero probability only in that topic. They first select an anchor word for each topic by solving numerous linear programs; then, in the recovery step, reconstruct the topic distributions given these anchor words. Bittorf et al. [24] and Gillis et al. [9] reduce the number of linear programs. Gillis et al. [10] propose a linear projection instead of linear programming. Arora et al. [2] replace the linear programming with a combinatorial anchor selection algorithm that is efficient and scalable as compared with other approaches. Nguyen et al. [21] develop a new regularized algorithms to mathematically resemble the rich priors and improves the interpretability of topic models.

Inspired by the word-pair pattern of Biterm Topic Model (BTM) [26], we develop a word co-occurrence network (WCN) that utilizes the word co-occurrence information in the whole corpus. More specifically, rather than considering words as vectors in the high dimensional space, we regard each word as a node in the word co-occurrence network. Exploiting the word-pair patterns of BTM and the separability assumption, we conclude that the anchor words do not appear simultaneously in any single document. Based on this observation, we propose two new anchor word selection algorithms. The first, denoted as HardClique, finds K-clique in the unweighted complementary graph to select the anchor words. The second, denoted as SoftClique, finds a maximum edge-weight clique of size K in the weighted complementary graph as the alternative.

To the best of our knowledge, this is the first attempt to use graph theory for the anchor selection in topic modeling. It is a novel and interesting perspective by transforming the corpus into a word co-occurrence network and finding cliques in the network so as to find the anchor words in the original corpus. We perform

our methods on two types of real-world corpora and compare with two classical methods, the LDA method [6] and the anchor word method [2]. Experiments show that our method is very promising when evaluated on topic quality as well as interpretability. Our work relates the anchor selection problem in the area of data mining to the classic area of combinatorial optimization. In this way, other advanced methods for related problems in this classic area, such as theory and method for maximal independent set, can be transferred smoothly into the comparatively new area of topic modeling. Moreover, our method can be further investigated for many applications. For example, we might be able to determine the number of topics automatically by finding maximal cliques in the generated network.

2 Related Work

We review two related works: the Anchor Word Algorithm (AWA) [2] and the word-pair pattern of Biterm Topic Model (BTM) [26]. AWA is a provable learning algorithm for topic model inference. The algorithm can be divided into two steps: anchor selection and recovery. A better anchor word selection plays a key role in this algorithm. BTM is a probabilistic generative model that learns topics by directly modeling the generation of the word-pair patterns in the whole corpus.

2.1 The Anchor Word Algorithm

Learning Topics. The Anchor Word Algorithm (AWA) is based on the reasonable separability assumption. It assumes that there exists at least a special word to identify each topic. For example, "winger", "field", "shot" are likely to appear in articles related to football, but only "winger" is suitable to be the anchor word as it is most unambiguous.

Compared with the probabilistic models that work directly on the documents and words, AWA mainly focuses on the probability of the word co-occurrence. Denote the vocabulary size by V and the number of topics by K. At the first step, AWA generates the word co-occurrence matrix $\mathbf{Q}$ by applying a simple calculation [2]. $\mathbf{Q}$ is a $V \times V$ matrix whose element $Q_{i,j}$ stands for the probability of word i and word j simultaneously appear in a single document. The $\overline{\mathbf{Q}}$ matrix is the row normalized version of $\mathbf{Q}$ and its element stands for the conditional probability that word j appears given that word i appears in the same document.

AWA finds a set of anchor words from the co-occurrence matrix. Denote the indices of the anchor words as $S = \{s_1, s_2, ..., s_K\}$, the rows of $\overline{\mathbf{Q}}$ can be reconstructed as a convex combination of the rows corresponding to the anchor words [2]. We denote the coefficient of the reconstruction as a matrix C, where $C_{i,k} = p(z = k|w = i)$. More specifically,

$$\overline{Q}_{i,j} = \sum_{k=1}^{K} p(z = k|w = i)\overline{Q}_{s_k,j} = \sum_{k=1}^{K} C_{i,k}\overline{Q}_{s_k,j} \tag{1}$$

To find the topic distribution $p(w = i|z = k)$, the probability of word i given topic k, denoted as $A_{i,k}$, it is necessary to calculate $C_{i,k}$, the probability of topic k given word i. Once $C_{i,k}$ is learned, we can easily discover the topic distribution by the Bayes formula.

$$A_{i,k} \propto C_{i,k} p(w = i) = C_{i,k} \sum_j \overline{Q}_{i,j} \tag{2}$$

The reconstruction coefficients, $C_{i,k}$, can be uncovered with respect to two measures, RecoverKL and RecoverL2 [2]:

$$\text{RecoverKL: } C_{i,\cdot} = \underset{C_{i,\cdot}}{\operatorname{argmin}} D_{(KL)}(\overline{Q}_{i,\cdot} \| \sum_k C_{i,k} \overline{Q}_{s_k,\cdot}) \tag{3}$$

$$\text{RecoverL2: } C_{i,\cdot} = \underset{C_{i,\cdot}}{\operatorname{argmin}} D_{(L2)}(\overline{Q}_{i,\cdot} \| \sum_k C_{i,k} \overline{Q}_{s_k,\cdot}) \tag{4}$$

Finding Anchor Words. Figure 1 illustrates the conclusion drawn by Eq. (1) that normal words (in grey) lie in the convex hull of anchor words (in dark).

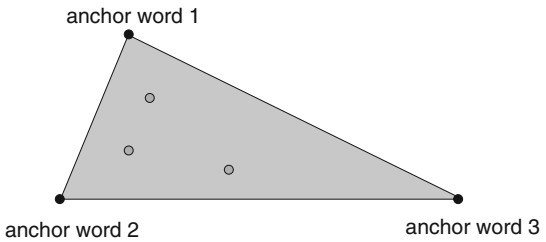

Fig. 1. Relations of anchor words and normal words. A anchor word is a vertex of the convex hull and the normal words lie in the convex hull.

Arora et al. propose a pure combinatorial algorithm, referred to as the FastAnchorWords algorithm [2], to find the anchor words in the anchor selection phase that totally avoids using the linear programming method. Each row of the matrix **Q** is regarded as a point in a V-dimensional space, and we can get V points as the input of the FastAnchorWords algorithm. FastAnchorWords algorithm then iteratively finds the furthest point from the subspace spanned by the set of anchor words found so far. Meanwhile, when faced with many choices for the next anchor word, this algorithm adopts a greedy strategy to find a point that is furthest to the set of points found so far. Arora et al. also provide analysis to guarantee the feasibility of the algorithm.

2.2 The Biterm Topic Model

The Biterm Topic Model (BTM) [26] is a probabilistic generative model that models the generation of the biterms (word co-occurrence pattern) in the whole corpus rather than documents. Figure 2 shows the graphical representation of BTM. The main purpose of BTM is to tackle the sparsity problem in topic modeling over short texts. Notice that the method yields great results even in normal text data. In BTM, any two distinct words extracted from a document is a biterm. The probability that a biterm drawn from a specific topic is further captured by the probability that two words in the biterm are drawn from the topic.

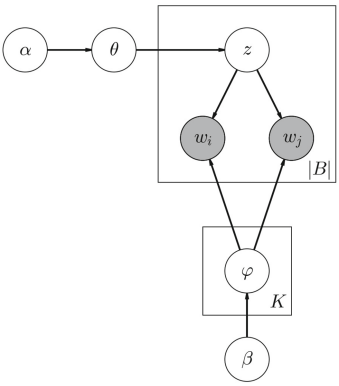

Fig. 2. Graphic representation of BTM. BTM models the generation procedure of biterms in a corpus rather than documents. α and β are the fixed hyperparameters.

For each biterm $b = (w_i, w_j)$, we should choose a topic first and draw the two words independently. Thus, the joint probability of a biterm can be written as [26]:

$$p(w_i, w_j) = \sum_z p(z)p(w_i|z)p(w_j|z) \tag{5}$$

where z is a topic.

Inspired by the word-pair pattern in BTM, we propose two novel anchor selection algorithms by regarding the corpus as a word co-occurrence network.

3 The Proposed Method

3.1 The Word Co-occurrence Network (WCN)

Generally speaking, a specific topic is represented as a group of semantically related words in topic models, while the correlation is revealed by word co-occurrence patterns in the documents. AWA and BTM both utilize the information of the word co-occurrence.

BTM assumes that the whole corpus is a mixture of topics, where each word-pair is independently drawn from a specific topic. The probability that a word-pair is drawn from a specific topic is further captured by the chance that two words in the word-pair are drawn from the same topic [26]. Inspired by the word-pair pattern, we build a word co-occurrence network by regarding words as nodes in the network and the word co-occurrence probability as the edge weights. The constructed word co-occurrence network can fully leverage the rich global word co-occurrence patterns to better reveal the latent topics. Meanwhile, from the probabilistic perspective, each entry of $\mathbf{Q}$ stands for the empirical word co-occurrence probability in the corpus. Figure 3 illustrates an example of the constructed word co-occurrence sub-network. The nodes in the network stand for words and the lighter color line indicates the lower-weight relationship. For example, edge between "layer" and "recognition" is of high weight while edge between "operate" and "redundancy" is of low weight.

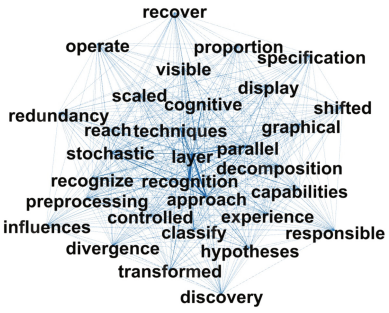

Fig. 3. An example of word co-occurrence sub-network. The nodes in the network stand for words and the lighter color line indicates the lower-weight relationship.

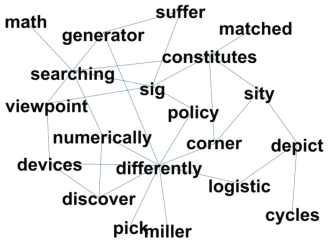

Fig. 4. An example of the complementary graph G. Edge exists between two words if their joint probability is smaller than a threshold ε.

3.2 Finding Cliques of Anchor Words

The separability assumption claims that an anchor word only appears in a specific topic, and it strongly indicates the corresponding topic. Thus, based on the main idea of BTM that the whole corpus is a mixture of topics, where each word-pair is independently drawn from a specific topic, we can draw an interesting conclusion that anchor words belonging to different topics will not appear simultaneously in the same document.

$$p(w_{s_k}, w_{s_{k'}}) = \sum_z p(z)p(w_{s_k}|z)p(w_{s_{k'}}|z) = 0 \tag{6}$$

Now consider the anchor selection phase and our goal is to find anchor words from the word co-occurrence network. Since we only have finitely many documents in the corpus, matrix $\mathbf{Q}$ is only an approximation to its expectation. Thus,

each entry of $\mathbf{Q}$ stands for the empirical word co-occurrence probability in the corpus. The constraint in Eq. (6) may not be strictly satisfied. Strictly speaking, anchor words belonging to different topics will not appear simultaneously or they appear together with a very low probability.

$$p(w_{s_k}, w_{s_{k'}}) = \sum_z p(z)p(w_{s_k}|z)p(w_{s_{k'}}|z) \xrightarrow{threshold:\varepsilon} 0 \tag{7}$$

Thus, the constraint in Eq. (6) need to be updated to Eq. (7). Accordingly, in the word co-occurrence network, the anchor word selection is to find a set of nodes which are connected with each other by very low-value edges. In practice, we propose two methods to find nodes corresponding to anchor words in the complementary graph. In this section,we describe the two proposed methods.

The HardClique Method. In the HardClique method, we allow the probability between anchor words to be lower than a small value ε. In order to find anchor words in the word co-occurrence network, we transform the network into a variant of complementary graph $G = (V, E)$ where V is the same set of nodes and $E = \{(v_i, v_j)|w_{ij} < \varepsilon\}$. If the joint probability of two words is sufficiently small, then there exists an edge connecting them in the complementary graph G. Figure 4 shows an example.

Thus, based on the complementary graph and the conclusion drawn from Eq. (6) or (7), we could choose the set of K nodes contained in a K-clique as the anchor words. The details are in Algorithm 1.

Algorithm 1. The HardClique method

Input: Corpus D, number of topics K, threshold ε
Output: Topic distribution matrix $\mathbf{A}$
1: Calculate the word co-occurrence matrix $\mathbf{Q}$ from corpus D
2: Transform the word co-occurrence network to graph G with respect to threshold ε
3: $S \leftarrow$ Find-K-Clique(G)
4: $\mathbf{A} \leftarrow$ RecoverL2($S, \mathbf{Q}$)

In practice, it is possible to find multiple sets of K-cliques, as one topic contains more than one anchor word. For example, "winger" and "goalkeeper" both strongly imply the football topic so they can both play the role of the anchor word. How to select the threshold ε plays a key role in the HardClique method. If the threshold is too small, we may not find any K-clique. On the contrary, if the threshold is too large, there could exist many candidates of anchor word cliques, and cause low topic quality. A trade-off between the quality and the feasibility should be carefully considered in practice. As nodes that stand for anchor words have at least $K-1$ degrees in the graph, we can recursively remove nodes whose degree is less than $K-1$ to reduce the size of graph and speed up the calculation.

The SoftClique Method. The other method we propose is with less restriction. As an alternative, we find K nodes whose sum of joint probability among each

node pair is the smallest. We first replace the weight of edges w_{ij} by $1 - w_{ij}$ to get the complementary graph. The problem turns out to be the maximum edge-weighted clique problem (MEWCP), a well-known NP-hard problem [1,23].

Algorithm 2. GENETIC-MEWCP

Input: Graph G, number of topics K, number of iterations T, population size N
Output: Maximum edge-weighted clique $S = \{v_1, v_2, \cdots, v_K\}$

```
maxweight ← 0
population ← A set of N k-cliques randomly chosen from G
do simple move to cliques in the population
for i ← 1 to T do
   exchange nodes between cliques in population
   do simple move to cliques in the population
   for i ← 1 to |population| do
      if weight(population[i]) > maxweight then
         S ← population[i]
         maxweight ← weight(population[i])
      end if
   end for
end for
```

Algorithm 3. The SOFTCLIQUE method

Input: Corpus D, number of topics K, number of iterations T, population size N
Output: Topic distribution matrix $\mathbf{A}$

```
Calculate word co-occurrence Q from corpus D
Replace weight of edges in word co-occurrence network G
S ← GENETIC-MEWCP(G, K, T, N)
A ← RECOVERL2(S, Q)
```

We could apply a simple local search method with the following genetic strategy. Find a clique by some greedy method, then define a simple move as swapping a node in the current clique that could increase the sum of the weights until a local maximum is achieved. To jump out of the local maximum, we exchange nodes between local maximum cliques and do a simple move again. This process is excuted iteratively to get a maximum edge-weight clique of size K. The GENETIC-MEWCP is given in Algorithm 2. The overall SOFTCLIQUE method is shown in Algorithm 3. Note that we could use any existing algorithm for the MEWCP.

Once the anchor words are found, the remaining of the procedure is the same as what was described in Sect. 2. We choose RecoverL2 measure in Eq. 4 to recover the topic distributions.

4 Experiments

In this section, we introduce the experiments conducted on real-world corpora to demonstrate the effectiveness of the proposed approaches. We take two typical methods as our baselines, namely LDA and AWA. All experiments are carried on a Linux server with Intel(R) Xeon(R) 2.00 GHz CPU and 56 GB memory.

4.1 Datasets

Two real-world corpora in different domains are used in the experiments, namely KOS and NIPS. The KOS corpus contains 3430 documents from the Daily Kos blog. The NIPS corpus contains 1500 papers from the NIPS conferences. All the corpora have been preprocessed into the bag of words format and stopwords are removed based on a stopword list.

We learn 10 or 20 topics on the corpora by four methods: LDA, AWA, and our HardClique and SoftClique methods. For LDA, we use an open-source python code implementation of LDA using Gibbs sampling[1]. The prior parameters are tuned via grid search: $\alpha = 0.1$ and $\beta = 0.01$. In our experiments, Gibbs sampling was run for 2000 iterations. For the HardClique method, we use the clique-finding algorithm proposed in [22]. For the SoftClique method, we use our own algorithm shown in Algorithm 3.

4.2 Evaluation Metrics

To compare the quality of topics found by different methods, we use a numeric metric called the coherence score [20]. Given a topic T and M words that appear most in topic $V^{(T)} = \{v_1, v_2, \cdots, v_M\}$, the coherence score is calculated as follows:

$$\text{Coherence}(T, V^{(T)}) = \sum_{i=2}^{M} \sum_{j=1}^{i-1} \log \frac{D(v_i, v_j) + 1}{D(v_j)} \tag{8}$$

$D(v_i, v_j)$ is the number of documents that v_i and v_j appear simultaneously and $D(v_i)$ is the number of documents v_i appears. 1 in the numerator of Eq. (8) is a smoothing constant to avoid zero value. As words strongly related to the same topic tend to co-occur in the same document, higher coherence score implies higher topic quality. It has been found that the coherence score is highly correlated with topic coherence.

To make a general evaluation, we compare the average of the coherence scores of K topics found by the four methods, namely $\frac{1}{K}\sum_k \text{Coherence}(T_k, V^{(T_k)})$. In the experiments, M is set to 10 and K is set to 10 or 20. As HardClique may find several sets of anchor words, we calculate the average value over all sets.

[1] https://pypi.python.org/pypi/lda.

Besides the coherence score, we also compute the held-out likelihood, the logarithm probability of previous unseen data. Higher held-out likelihood value indicates stronger ability to fit unseen data. As held-out likelihoods can not be calculated directly by the anchor methods, we use variational inference [6] with the topic distribution learned by the anchor methods. The parameter α of variational inference is set to 0.1.

4.3 Results

Coherence. The result on coherence is averaged over three random runs, as shown in Fig. 5. In general, our two methods outperform the two baselines. For the NIPS corpus, HardClique and SoftClique show the best coherence when K is set to 10 and 20 respectively. For the KOS corpus, when K is set to 10, our two methods both outperform the two baselines. When K is set to 20, SoftClique method produces the best coherence. In general, our methods achieve considerable improvement for the topic quality.

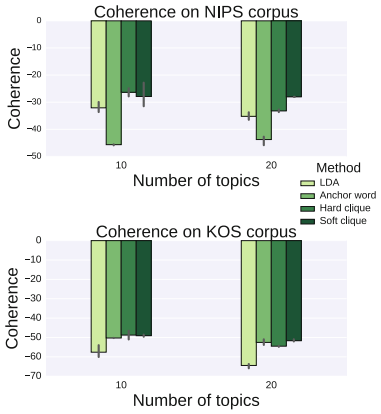

Fig. 5. Coherence score on two corpora. SoftClique performs the best on the two corpora when K is set to 20, and HardClique performs the best when K is set to 10.

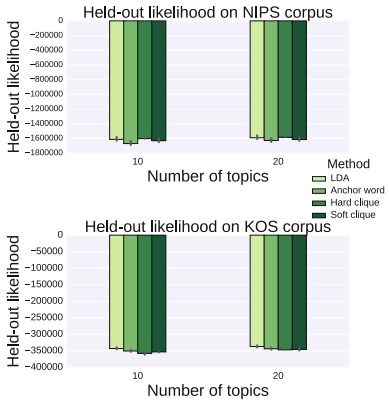

Fig. 6. Held-out likelihood on two corpora. The four methods show very similar performance and LDA is slightly better.

Recall that HardClique may find several sets of anchor words from the same corpus. For example, two cliques share 19 words and possess 1 exclusive word (in dark black) respectively. It indicates the exclusive words of two cliques are anchor words corresponding to the same topic. This gives reasonable interpretation for the result.

For more intuitive demonstration, we list top 10 words of some topics in Table 1. A more coherent topic is supposed to have more related top words, as

Table 1. The top 10 words given by 4 different methods on NIPS corpus for 5 topics. Top words of clique method are highly coherent.

Topic	LDA	Anchor word	HardClique	SoftClique
Topic 1	image, object images recognition features, feature pixel, view face, visual	network, set data, image system, neural images, training input, algorithm	data, image unit, input network, images output, hidden set, field	data, vector map, space point, feature set, input cluster, image
Topic 2	network, unit input, output weight, layer neural, hidden training, net	network, word training, input unit, set neural, system data, error	network, unit neural, weight hidden, input error, architecture output, algorithm	network, weight unit, input output, neural training, layer hidden, error
Topic 3	signal, filter information frequency noise, channel component system correlation auditory	speech, signal network, output recognition information layer, input speaker, acoustic	network, system pattern, neural output, training signal, recognition speech, input	word, signal speech, processing system, control training, subject recognition adaptation
Topic 4	image, object image recognition features, feature pixel, view face, visual	visual, signal object, image neuron, orientation map, spatial cortex, response	visual, processing direction, object motion, input activity, science spatial, field	system, image data, visual object, point images, direction recognition, feature
Topic 5	vector, data space, point function dimensional kernel, linear matrix, basis	architecture algorithm problem, network unit, weight vector, data set, gradient	algorithm, data set, distribution problem, method parameter, vector information, number	algorithm function vector, problem gradient, method architecture matrix error, space

top words are representative words for a topic. The result shows that the two clique methods produce highly coherent top words.

Held-out Likelihood. We choose 15% documents (165 documents for NIPS corpus and 515 documents for KOS corpus) for the held-out test. The result is averaged over three random runs and demonstrated in Fig. 6. We find that the HardClique method and LDA provide the best held-out likelihood on NIPS and KOS corpus respectively. In general, the difference is within the range of variability between documents.

5 Conclusion and Future Work

How to do a better anchor word selection is an open problem. We build a word co-occurrence network that considers words as nodes and the co-occurrence probability as edge weights. We propose two new anchor selection methods, namely finding K-clique in the unweighted complementary graph and finding maximum edge-weight clique in the weighted complementary graph. Experimental results on real-world corpora suggest that our methods outperform AWA and Gibbs sampling for LDA for the topic quality, the four methods provide comparable results for the held-out likelihood.

We apply a simple local search method with genetic strategy to get a maximum edge-weight clique of size K. It provides promising space for potential improvement utilizing more sophisticated algorithms in graph theory. Also, our method probably does not require the anchor words to form a clique for the anchor selection. In future work, we will conduct other combinatorial algorithm on the word co-occurrence graph.

Acknowledgments. This research work is supported by National Natural Science Foundation of China (61772219, 61472147), US Army Research Office (W911NF-14-1-0477) and Shenzhen Science and Technology Planning Project (JCYJ20170307154749425). We also thank Junru Shao for valuable discussions.

References

1. Alidaee, B., Glover, F., Kochenberger, G., Wang, H.: Solving the maximum edge weight clique problem via unconstrained quadratic programming. European Journal of Operational Research **181**(2), 592–597 (2007)
2. Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., Zhu, M.: A practical algorithm for topic modeling with provable guarantees. In: ICML, pp. 280–288 (2013)
3. Arora, S., Ge, R., Moitra, A.: Learning topic models-going beyond SVD. In: FOCS, pp. 1–10. IEEE, (2012)
4. Bhadury, A., Chen, J., Zhu, J., Liu, S.: Scaling up dynamic topic models. In: WWW, pp. 381–390 (2016)
5. Blei, D.M., Lafferty, J.D.: A correlated topic model of science. In: The Annals of Applied Statistics, pp. 17–35 (2007)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
7. Chen, J., Zhu, J., Wang, Z., Zheng, X., Zhang, B.: Scalable inference for logistic-normal topic models. In: NIPS, pp. 2445–2453 (2013)
8. Foulds, J., Boyles, L., DuBois, C., Smyth, P., Welling, M.: Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation. In: KDD, pp. 446–454. ACM, (2013)
9. Gillis, N.: Robustness analysis of Hottopixx, a linear programming model for factoring nonnegative matrices. SIAM Journal on Matrix Analysis and Applications **34**(3), 1189–1212 (2013)
10. Gillis, N., Vavasis, S.A.: Fast and robust recursive algorithmsfor separable nonnegative matrix factorization. IEEE transactions on pattern analysis and machine intelligence **36**(4), 698–714 (2014)
11. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National academy of Sciences **101**(Suppl. 1), 5228–5235 (2004)
12. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international conference on Research and development in information retrieval, pp. 50–57. ACM, (1999)
13. Jiang, D., Leung, K.W.T., Ng, W.: Fast topic discovery from web search streams. In: WWW, pp. 949–960. ACM, (2014)
14. Jo, Y., Hopcroft, J.E., Lagoze, C.The web of topics: discovering the topology of topic evolution in a corpus. InWWW, pp. 257–266. ACM, (2011)

15. Kataria, S., Agarwal, A.: Supervised Topic Models for Microblog Classification. In: ICDM, pp. 793–798. IEEE, (2015)
16. Li, A.Q., Ahmed, A., Ravi, S., Smola, A.J.: Reducing the sampling complexity of topic models. In: KDD, pp. 891–900. ACM, (2014)
17. Li, C., Wang, H., Zhang, Z., Sun, A., Ma, Z.: Topic modeling for short texts with auxiliary word embeddings. In: Proceedings of the 39th International conference on Research and Development in Information Retrieval, pp. 165–174. ACM, (2016)
18. Lin, T., Tian, W., Mei, Q., Cheng, H.: The dual-sparse topic model: mining focused topics and focused terms in short text. In: WWW, pp. 539–550. ACM, (2014)
19. Liu, X., Zeng, J., Yang, X., Yan, J., Yang, Q.: Scalable parallel EM algorithms for latent Dirichlet allocation in multi-core systems. In: WWW, pp. 669–679. (2015)
20. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: EMNLP, pp. 262–272. ACL, (2011)
21. Nguyen, T., Hu, Y., Boyd-Graber, J.L.: Anchors Regularized: Adding Robustness and Extensibility to Scalable Topic-Modeling Algorithms. In: ACL, pp. 359–369 (2014)
22. Palla, G., Dernyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. (2005)
23. Pullan, W.: Approximating the maximum vertex/edge weighted clique using local search. Journal of Heuristics **14**(2), 117–134 (2008)
24. Recht, B., Re, C., Tropp, J., Bittorf, V.: Factoring nonnegative matrices with linear programs. In: NIPS, pp. 1214–1222 (2012)
25. Wang, S., Chen, Z., Fei, G., Liu, B., Emery, S.: Targeted Topic Modeling for Focused Analysis. In: KDD, pp. 1235–1244 (2016)
26. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: WWW, pp. 1445–1456. ACM, (2013)
27. Yan, X., Guo, J., Liu, S., Cheng, X., Wang, Y.: Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In: Proceedings of the 2013 International Conference on Data Mining, pp. 749–757. SIAM, (2013)
28. Yang, S.H., Kolcz, A., Schlaikjer, A., Gupta, P.: Large-scale high-precision topic modeling on twitter. In: KDD, pp. 1907–1916. ACM, (2014)
29. Zhang, H., Kim, G., Xing, E.P.: Dynamic topic modeling for monitoring market competition from online text and image data. In: KDD, pp. 1425–1434. ACM, (2015)
30. Zuo, Y., Wu, J., Zhang, H., Lin, H., Wang, F., Xu, K., Xiong, H.: Topic Modeling of Short Texts: A Pseudo-Document View. In: KDD, pp. 2105–2114. ACM, (2016)

http://www.springer.com/978-981-10-6892-8

Theoretical Computer Science
35th National Conference, NCTCS 2017, Wuhan, China,
October 14-15, 2017, Proceedings
Du, D.; Lian, L.; En, Z.; He, K. (Eds.)
2017, XXI, 356 p. 165 illus., Softcover
ISBN: 978-981-10-6892-8