Sentence Clustering using PageRank Topic Model

Kenshin Ikegami
Department of Systems Innovation
The University of Tokyo
Tokyo, Japan
kenchin110100@gmail.com

Yukio Ohsawa
Department of Systems Innovation
The University of Tokyo
Tokyo, Japan
ohsawa@sys.t.u-tokyo.ac.jp

Abstract

The clusters of review sentences on the viewpoints from the products' evaluation can be applied to various use. The topic models, for example Unigram Mixture (UM), can be used for this task. However, there are two problems. One problem is that topic models depend on the randomly-initialized parameters and computation results are not consistent. The other is that the number of topics has to be set as a preset parameter. To solve these problems, we introduce PageRank Topic Model (PRTM), that approximately estimates multinomial distributions over topics and words in a vocabulary using network structure analysis methods to Word Co-occurrence Graphs. In PRTM, an appropriate number of topics is estimated using the Newman method from a Word Co-occurrence Graph. Also, PRTM achieves consistent results because multinomial distributions over words in a vocabulary are estimated using PageRank and a multinomial distribution over topics is estimated as a convex quadratic programming problem. Using two review datasets about hotels and cars, we show that PRTM achieves consistent results in sentence clustering and an appropriate estimation of the number of topics for extracting the viewpoints from the products' evaluation.

1 Introduction

Many people buy products through electronic commerce and Internet auction site. Consumers have to use products' detailed information for decision making in purchasing because they cannot see the real products. In particular, reviews from other consumers give them useful information because reviews contain consumers' experience in practical use. Also, reviews are useful for providers of products or services to measure the consumers' satisfaction.

In our research, we focus on generating clusters of review sentences on the viewpoints from the products' evaluation. For example, reviews of home electric appliance are usually written based on the following the viewpoints: performance, design, price, etc. If we generate clusters of the review sentences on these viewpoints, the clusters can be applied to various uses. For example, if we extract representative expressions from clusters of sentences, we can summarize reviews briefly. This is useful because some products have thousands of reviews and hard to be read and understood.

There are various methods to generate clusters of sentences. Among several methods, we adopt probabilistic generative models for sentence clustering because the summarizations of clusters can be represented as word distributions. Probabilistic generative models are the methods that assume underlying probabilistic distributions generating observed data, and that estimate the probabilistic distributions from the observed data. In language modeling, these are called topic models.

Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is a well-known topic model used in document clustering. LDA represents each document as a mixture of topics. A topic means a multinomial distribution over words in a vocabulary.

Unigram Mixture (UM) (Nigam et al., 2000) as-

sumes that each document is generated by a multinomial distribution over words in a vocabulary, $\boldsymbol{\phi}_k = (\phi_{k1}, \cdots, \phi_{kV})$, where V denotes the size of vocabulary and ϕ_{kv} denotes the appearance probability of v-th term in the k-th topic. UM estimates a multinomial distribution over topics, $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_K)$, where θ_k denotes the appearance probability of k-th topic. After all, K+1 multinomial distributions, $\boldsymbol{\theta}$ and $\boldsymbol{\phi} = (\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_K)$ are estimated from the observed data, where K denotes the number of topics.

Using estimated $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, the probability that a document is generated from $\boldsymbol{\phi}_k$ is calculated. This probability determines the clusters of the sentences.

In UM, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ can be estimated by iterative computation. However, since $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are initialized randomly, computation results are not consistent. In addition to this, the number of topics K has to be set as a preset parameter.

To estimate the appropriate number of topics, the average cosine distance ($AveDis$) of each pair of topics can be used (Cao et al., 2009). This measure is based on the assumption that better topic distributions have fewer overlapping words. However, to estimate the appropriate number of topics based on this measure, we need to set several numbers of topics and it takes much time to calculate.

In this paper, we introduce PageRank Topic Model (PRTM) to consistently estimate $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ using Word Co-occurrence Graphs. PRTM consists of 4 steps as follows:

1. Convert corpus W into a Word Co-occurrence Graph G_w.
2. Divide graph G_w into several communities.
3. Measure PageRank in each community and estimate multinomial distributions over words in a vocabulary $\boldsymbol{\phi}$.
4. Estimate a multinomial distribution over topics $\boldsymbol{\theta}$ as a convex quadratic programming problem assuming the linearity of $\boldsymbol{\phi}$.

Network structures have been applied to several Natural Language Processing tasks (Ohsawa et al., 1998) (Bollegala et al., 2008). For example, synonyms can be identified using network community detection method, e.g. the Newman method (Clauset et al., 2004) (Sakaki et al., 2007). In this research, we also apply the Newman method to detect communities of co-occurrence words in step 2. In step 3, we calculate the appearance probability of nodes using PageRank (Brin and Page, 1998). PageRank is the appearance probability of nodes in a network. In Word Co-occurrence Graph G_w, each node represents a word. Therefore, we regard a set of PageRank of nodes as $\boldsymbol{\phi}$. After that, $\boldsymbol{\theta}$ is estimated using a convex quadratic programming problem based on the assumption of the linearity of $\boldsymbol{\phi}$ in step 4. From these steps, reproducible $\boldsymbol{\phi}$, $\boldsymbol{\theta}$ and clustering results can be obtained because the Newman method, PageRank and the convex quadratic programming problem are not depending on random initialization of parameters.

There is another advantage to identify communities of co-occurrence words using the Newman method. The Newman method yields an optimized number of communities K in the sense it extracts communities to maximize Modularity Q. Modularity Q is one measure of the strength of division of a network structure into several communities. When modularity Q is maximized, the graph is expected to be divided into an appropriate number of communities.

Our main contributions are summarized as follows:

- Using PRTM, we estimate consistent multinomial distributions over topics and words. It enables us to get consistent computation results of sentence clustering.
- PRTM yields an appropriate number of topics, K, as well as the other parameters. It is more suitable to estimate the number of viewpoints from the products' evaluation than the average cosine distance measurement.

In this paper, we first explain our proposed method, PRTM, in section 2. We show the experimental results in section 3 and compare with related works in section 4. At last, we discuss our conclusions in section 5.

2 Proposed Method

In this section, we explain the Newman method and PageRank in subsection 2.1, 2.2. After that, we

show our proposed method, PageRank Topic Model, in subsection 2.3.

2.1 Newman method

The Newman method is a method to detect several communities from a network structure (Clauset et al., 2004). The method puts together nodes to maximize Modularity Q. Modularity Q is defined as follows:

$$Q = \sum_{i=1}^{K} (e_{ii} - a_i^2) \quad (1)$$

where K is the number of communities, e_{ii} is the ratio of the number of edges in the i-th community to the total number of edges in the network, a_i is the ratio of the number of edges the i-th community from the other communities to the total number of edges in the network.

Modularity Q represents the density of connections between the nodes within communities. Therefore, the higher the Modularity Q is, the more accurately the network is divided into communities. In the Newman method, communities are extracted by the following steps:

1. Assign each node to a community.
2. Calculate the increment in Modularity ΔQ when any two communities are merged into one community.
3. Merge the two communities, that score the highest ΔQ in the previous process, into one community.
4. Repeat step 2 and step 3 as long as Q increases.

2.2 PageRank

PageRank (Brin and Page, 1998) is the algorithm to measure the importance of each node in a network structure. It has been applied to evaluating the importance of websites in the World Wide Web. In PageRank, the transition probability matrix $H \in \mathbb{R}_+^{V \times V}$ is generated from network structure, where V denotes the number of nodes. H_{ij} represents the transition probability from node n_i to node n_j, a ratio of the number of edges from node n_i to node n_j to the total number of edges from node n_i. However, if node n_i does not have outgoing edges (dangling node), node n_i does not have transition to any other nodes. To solve this problem, matrix H is extended to matrix $G \in \mathbb{R}_+^{V \times V}$ as follows:

$$G = dH + (1-d)\frac{1}{V}\mathbf{1}^T\mathbf{1} \quad (2)$$

where d is a real number within [0, 1] and $\mathbf{1} \in \{1\}^V$. PageRank of node n_i, i.e. $PR(n_i)$, is calculated using matrix G as follows:

$$\boldsymbol{R}^T = \boldsymbol{R}^T G \quad (3)$$

where $\boldsymbol{R} = (PR(n_i), \cdots, PR(n_V))^T$. Equation (3) can be solved with the simultaneous linear equations or the power method.

2.3 PageRank Topic Model

In this subsection, we explain our proposed method, PageRank Topic Model (PRTM), to estimate a multinomial distribution over topics $\boldsymbol{\theta}$ and words in a vocabulary $\boldsymbol{\phi}$ using a Word Co-occurrence Graph. PRTM consists of 4 steps as shown in section 1. We explain them by following these steps.

Step 1: First, we convert a dataset into a bag of words. Each bag represents a sentence in the dataset. We define Word Co-occurrence Graph $G_w(V, E)$ as an undirected weighted graph where each vocabulary v_i is represented by a node $n_i \in V$. An edge $e_{ij} \in E$ is created between node n_i and node n_j if v_i and v_j co-occur in the bag of words.

Step 2: We apply the Newman method to graph G_w to extract communities $Com^{(k)}$, where $k = 1, \cdots, K$ and K denotes the number of communities. $Com^{(k)}$ is a set of nodes in G_w. From this results, we generate Word Co-occurrence SubGraph $G_w^{(k)}(V^{(k)}, E^{(k)})$. Although $V^{(k)}$ is the same as V of G_w, an edge $e_{ij}^{(k)} \in E^{(k)}$ is created if node n_i or n_j exists in $Com^{(k)}$. Figure 1 shows the relationship between $Com^{(k)}$ and $G_w^{(k)}$.

Step 3: We measure the importance of each node in $G_w^{(k)}$ with PageRank. Page et al. (1999) explained PageRank by the random surfer model. A random surfer is a person who opens a browser to any page and starts following hyperlinks. PageRank can be interpreted as the probability of a random surfer existence in nodes. In this case, a node $n_i^{(k)}$ represents vocabulary v_i. Therefore $PR(n_i^{(k)})$ represents the

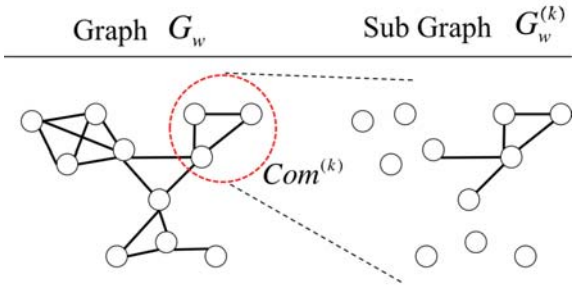

Figure 1: The relationship between $Com^{(k)}$ and $G_w^{(k)}$

appearance probability of word v_i in $G_w^{(k)}$. We regard $G_w^{(k)}$ as k-th topic and define multinomial distributions over words in a vocabulary $\boldsymbol{\phi}_k$ as follows:

$$\begin{aligned}\boldsymbol{\phi}_k &= (\phi_{k1}, \cdots, \phi_{kV}) \\ &= (PR(n_1^{(k)}), \cdots, PR(n_V^{(k)}))\end{aligned} \tag{4}$$

Step 4: We estimate a multinomial distribution over topics $\boldsymbol{\theta}$ using $\boldsymbol{\phi}$, that is estimated in Step 3. To estimate $\boldsymbol{\theta}$, we assume the linearity of $\boldsymbol{\phi}$ as follows:

$$\phi_{\cdot v} = \sum_{k=1}^{K} \phi_{kv}\theta_k \tag{5}$$

where $\phi_{\cdot v}$ denotes the appearance probability of v-th term in graph G_w. However, it is impossible to estimate a θ_k that satisfies Equation (5) in all of words in a vocabulary because each $\boldsymbol{\phi}_k$ is independently estimated using PageRank.

Therefore, we estimate θ_k minimizing the following equation:

$$\begin{aligned}&\arg\min_{\boldsymbol{\theta}} L \\ &= \arg\min_{\boldsymbol{\theta}} \sum_{v}^{V} (\phi_{\cdot v} - \sum_{k=1}^{K} \phi_{kv}\theta_k)^2 \\ &\text{s.t. } \|\boldsymbol{\theta}\| = 1, \boldsymbol{\theta} \geq 0\end{aligned} \tag{6}$$

By reformulating Equation (6), the following equation can be obtained:

$$\begin{aligned}&\arg\min_{\boldsymbol{\theta}} L \\ &= \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\boldsymbol{\theta}^T Q \boldsymbol{\theta} + c^T \boldsymbol{\theta} \\ &\text{s.t. } \|\boldsymbol{\theta}\| = 1, \boldsymbol{\theta} \geq 0\end{aligned} \tag{7}$$

where the (i, j)-th element of matrix $Q \in \mathbb{R}^{K \times K}$ denotes $2{\boldsymbol{\phi}_i}^T \boldsymbol{\phi}_j$ and the i-th element of vector c denotes $-2{\boldsymbol{\phi}_{\cdot}}^T \boldsymbol{\phi}_i$.

Equation (7) is formulated as a convex quadratic programming problem, of which a global optimum solution should be obtained.

The probability that document d is generated from k-th topic, i.e. $p(z_d = k|w_d)$, is calculated as follows:

$$\begin{aligned}p(z_d = k|w_d) &= \frac{p(w_d|k)p(k)}{\sum_{k'=1}^{K} p(w_d|k')p(k')} \\ &= \frac{\theta_k \prod_{v=1}^{V} \phi_{kv}^{N_{dv}}}{\sum_{k'=1}^{K} \theta_{k'} \prod_{v=1}^{V} \phi_{k'v}^{N_{dv}}}\end{aligned} \tag{8}$$

where N_{dv} denotes the number of v-th term in document d.

3 Experiments

In this section, we show the evaluation results of PRTM using real-world text data in comparison with UM and LDA. In subsection 3.1, we explain our test datasets and the measure used to evaluate sentence clustering accuracy. Furthermore, we present the conditions of UM and LDA in the same subsection. We show topic examples estimated by PRTM, UM, and LDA in subsection 3.2. In subsection 3.3, we compare the sentence clustering accuracy of PRTM with that of UM and LDA. In addition, we compare the estimated number of topics of PRTM with that of the average cosine distance measurement in subsection 3.4.

3.1 Preparation for Experiment

In the experiments, we used the following two datasets:

Hotel Reviews: This is Rakuten Travel[1] Japanese review dataset and has been published by Rakuten, Inc. In this dataset, there are 4309 sentences of 1000 reviews. We tokenized them using Japanese morphological analyzer, mecab[2], and selected nouns and adjectives. It contains a vocabulary of 3780 words and 19401 word tokens. During preprocessing, we removed high-frequency words appearing more than 300 times and low frequency words appearing less

[1] http://travel.rakuten.co.jp/

[2] http://taku910.github.io/mecab/

than two times. The sentences of this dataset were classified by two annotators. The annotators (humans) were asked to classify each sentence into six categories; "Service", "Room", "Location", "Facility and Amenity", "Bathroom", and "Food". We adopted these six categories because Rakuten Travel website scores hotels by these six evaluation viewpoints. In evaluation of sentence clustering accuracy, we used 2000 sentences from the total sentences which both the annotators classified into the same category.

Car Reviews: This is Edmunds[3] Car English review dataset and has been published by the "Opinion Based Entity Ranking" project (Ganesan and Zhai, 2011). In this dataset, there are 7947 reviews in 2009, out of which we randomly selected 600 reviews consisting of 3933 sentences. We tokenized them using English morphological analyzer, Stanford CoreNLP[4], and selected nouns, adjectives and verbs. It contains a vocabulary of 3975 words and 27385 word tokens. During preprocessing, we removed high-frequency words appearing more than 300 times and low frequency words appearing less than two times. All of the 3922 sentences were classified into eight categories by two annotators; "Fuel", "Interior", "Exterior", "Build", "Performance", "Comfort", "Reliability" and "Fun". We adopted these eight categories for the same reason as Hotel Review. There are 1148 sentences which both annotators classified into the same category and we used them in the evaluation of sentence clustering accuracy.

Evaluation: We measured Purity, Inverse Purity and their F_1 score for sentence clustering evaluation (Zhao and Karypis, 2001). Purity focuses on the frequency of the most common category into each cluster. Purity is calculated as follows:

$$Purity = \sum_i \frac{|C_i|}{n} \max_j Precision(C_i, L_j) \quad (9)$$

where C_i is the set of i-th cluster, L_j is the set of j-th given category and n denotes the number of samples. $Precision(C_i, L_j)$ is defined as:

$$Precision(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|} \quad (10)$$

[3]http://www.edmunds.com/

[4]http://stanfordnlp.github.io/CoreNLP/

However, if we make one cluster per sample, we reach a maximum purity value. Therefore we also measured Inverse Purity. Inverse Purity focuses on the cluster with maximum recall for each category and is defined as follows:

$$InversePurity = \sum_j \frac{|L_j|}{n} \max_i Precision(L_j, C_i) \quad (11)$$

In this experiment, we used the harmonic mean of Purity and Inverse Purity, F_1 score, as clustering accuracy. F_1 score is calculated as follows:

$$F_1 = \frac{2 \times Purity \times InversePurity}{Purity + InversePurity} \quad (12)$$

Estimation of number of topics: To estimate the appropriate number of topics, we used the average cosine distance measurement ($AveDis$) (Cao et al., 2009). $AveDis$ is calculated using the multinomial distributions $\boldsymbol{\phi}$ as follows:

$$corre(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = \frac{\sum_{v=0}^{V} \phi_{iv}\phi_{jv}}{\sqrt{\sum_{v=0}^{V}(\phi_{iv})^2}\sqrt{\sum_{v=0}^{V}(\phi_{jv})^2}}$$
$$AveDis = \frac{\sum_{i=0}^{K}\sum_{j=i+1}^{K} corre(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)}{K \times (K-1)/2} \quad (13)$$

where V denote the number of words in a vocabulary and K denotes the number of topics.

If topic i and j are not similar, $corre(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$ becomes smaller. Therefore, when the appropriate number of topics K is preset, that is all the topics have different word distributions, $AveDis$ becomes smaller.

Comparative Methods and Settings: We compared PRTM with UM and LDA in the experiments. UM can be calculated using several methods: EM algorithm (Dempster et al., 1977), Collapsed Gibbs sampling (Liu, 1994) (Yamamoto and Sadamitsu, 2005), or Collapsed Variational Baysian (Teh et al., 2006). In our experiments, topic and word distributions $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ were estimated using Collapsed Gibbs sampling for both the UM and LDA models. The hyper-parameter for all the Dirichlet distributions were set at 0.01 and were updated at every iteration. We stopped iterative computations when the difference of likelihood between steps got lower than 0.01.

cluster1			cluster2		
PRTM	UM	LDA	PRTM	UM	LDA
breakfast	breakfast	breakfast	bath	bath	breakfast
satisfaction	meal	satisfaction	wide	wide	service
very	satisfaction	support	care	care	absent
service	delicious	convenient	comfortable	good	location
meal	delicious	absent	big bath	absent	satisfaction
cluster3			cluster4		
PRTM	UM	LDA	PRTM	UM	LDA
good	station	breakfast	support	support	breakfast
location	convenient	reception	reception	reception	good
station	close	support	feeling	staff	satisfaction
cheap	location	satisfaction	reservation	check-in	very
fee	convenience-store	bath	good	kindness	shame
cluster5			cluster6		
PRTM	UM	LDA	PRTM	UM	LDA
different	reservation	support	absent	satisfaction	breakfast
bathing	plan	satisfaction	other	opportunity	wide
bathroom	non-smoking	breakfast	people	wide	station
difficult	preparation	reception	preparation	business-trip	absent
illumination	breakfast	very	voice	very	care

Table 1: Top 5th terms in each topic by PRTM, UM, and LDA. Each term has been translated from Japanese to English using Google translation.

3.2 Topic Examples

We used Hotel Reviews dataset and estimated words distributions ϕ by PRTM, UM, and LDA. All of the PRTM, UM, and LDA were given the number of topics $K = 6$.

In Table 1, we show the terms of top fifth appearance probabilities in each topic estimated. As we can see, PRTM and UM contain similar terms in cluster 1, 2, 3, and 4. For example, in cluster 1, both of PRTM and UM have terms, "breakfast" and "meal". Therefore its topic seems to be "Food." On the other hand, there are the same terms, "support" and "reception", in cluster 4. This topic seems to represent "Service." However, in LDA, the estimation seems to fail because all of the topics have similar words (e.g. the word "breakfast" exists in all the topics.) For these reasons, it is more suitable to assume that each sentence has one topic than to assume that it has multiple topics.

3.3 Sentence Clustering Accuracy

We evaluated sentence clustering accuracy comparing PRTM with UM and LDA on Hotel Review and Car Review datasets. By changing the number of topics K from 3 to 20, we trained topics and word distributions θ, ϕ with PRTM, UM, and LDA. We generated clusters of sentences by Equation (8) in PRTM and UM. In LDA, we decided the cluster of sentence using topic distributions of each sentence. The sentence clustering accuracy was evaluated by F_1 score on Purity and Inverse Purity. F_1 scores of UM and LDA were the mean values of the tests running ten times, because the computation results vary depending on randomly initialized θ and ϕ.

We present sentence clustering accuracy for all the PRTM, UM, and LDA in Figure 2. As shown in Figure 2, PRTM outperformed UM when the number of topics is more than six in both the Hotel and Car Review datasets. For UM, F_1 score became highest when K was small and gradually decreased when K became larger. On the other hand,

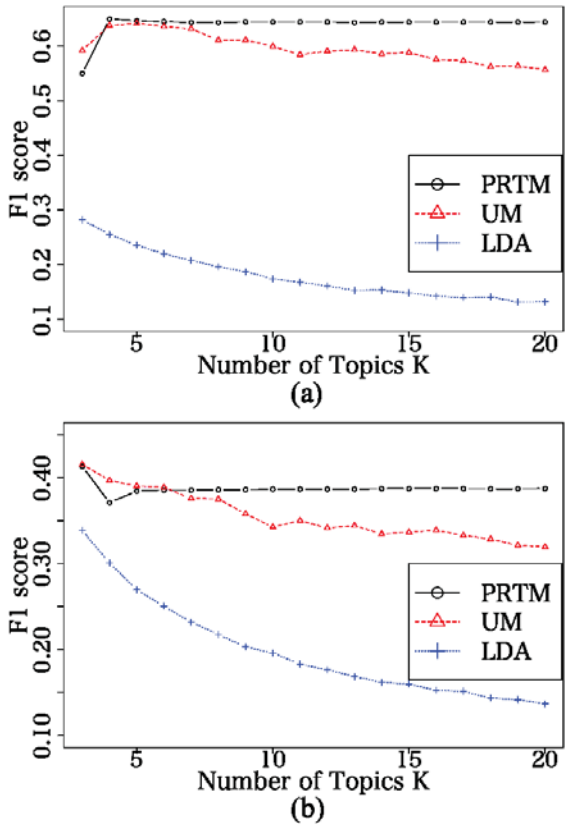

Figure 2: F_1 score comparison with different numbers of topics. (a) Hotel Reviews. (b) Car Reviews.

Number of Topics $K = 6$		
θ_k	PRTM	UM
θ_1	2.58×10^{-1}	3.11×10^{-1}
θ_2	2.54×10^{-1}	1.77×10^{-1}
θ_3	2.24×10^{-1}	1.71×10^{-1}
θ_4	1.68×10^{-1}	1.40×10^{-1}
θ_5	7.04×10^{-2}	1.27×10^{-1}
θ_6	2.70×10^{-2}	7.39×10^{-2}
Number of Topics $K = 12$		
θ_k	PRTM	UM
θ_1	2.52×10^{-1}	2.20×10^{-1}
θ_2	2.50×10^{-1}	1.23×10^{-1}
θ_3	2.17×10^{-1}	1.14×10^{-1}
θ_4	1.65×10^{-1}	9.58×10^{-2}
θ_5	6.94×10^{-2}	9.58×10^{-2}
θ_6	2.13×10^{-2}	7.34×10^{-2}
θ_7	1.79×10^{-2}	6.35×10^{-2}
θ_8	7.62×10^{-3}	6.03×10^{-2}
θ_9	2.28×10^{-4}	5.54×10^{-2}
θ_{10}	1.58×10^{-5}	4.02×10^{-2}
θ_{11}	1.28×10^{-5}	3.90×10^{-2}
θ_{12}	2.93×10^{-6}	1.83×10^{-2}

Table 2: The appearance probabilities θ_k comparison with $K = 6$ and $K = 12$. Sorted in descending order.

with PRTM, F_1 score did not decrease if K became larger. The F_1 scores of LDA were lower than PRTM and UM because it is not suitable for review sentence clustering as mentioned in subsection 3.2.

Table 2 shows the comparison of the appearance probabilities θ_k with the number of topics $K = 6$ and $K = 12$. Similar θ_k was estimated by PRTM and UM with $K = 6$. However, with $K = 12$, PRTM had the larger deviation of the θ_k from 2.93×10^{-6} to 2.52×10^{-1}. On the other hand, UM with $K = 12$ had the more uniform θ_k than PRTM. This large deviation of θ of PRTM prevents sentences in the same category from being divided into several clusters. This is the reason why the F_1 score of UM gradually decreased and PRTM achieved invariant sentence clustering accuracy.

3.4 Appropriate Number of Topics

PRTM yields an appropriate number of topics by maximization of Modularity Q. On the other hand, the appropriate number of topics in UM and LDA can be estimated using the average cosine distance ($AveDis$) measurement. Therefore, we compared Modularity of PRTM with $AveDis$ of UM and LDA with different numbers of topics. We trained topic and word distributions $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and estimated the optimal number of topics K with both of Hotel Reviews and Car Reviews. The $AveDis$ scores of UM and LDA were the mean values of the tests running three times for the same reason as subsection 3.3.

Figure 3 shows the experimental results. The $AveDis$ of UM got the smallest scores in $K = 47$ with Hotel Reviews and in $K = 47$ in Car Reviews. Furthermore, $AveDis$ of LDA decreased monotonically in the range of $K = 3$ to $K = 60$. On the other hand, the Modularity of PRTM got largest in $K = 7$ with Hotel Reviews and in $K = 6$ with Car Reviews. When we consider that Rakuten Travel website scores hotels by six viewpoints and that Edmunds website scores cars by eight viewpoints, the Modularity of PRTM estimates more appropriate number of topics than $AveDis$ of UM in review

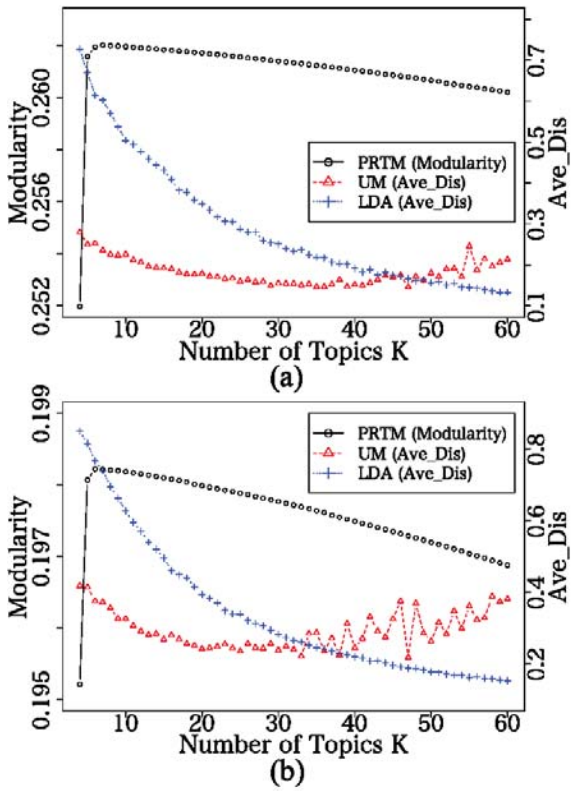

Figure 3: Modularity and Ave-Dis comparison with different numbers of topics. (a) Hotel Reviews. (b) Car Reviews.

datasets.

4 Related Work

There are several previous works of probabilistic generative models. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) estimates topic distributions for each document and word distributions for each topic. On the other hand, Unigram Mixtures (UM) (Nigam et al., 2000) estimates a topic distribution for all the documents and word distributions for each topic. In both papers, their models are tested at document classification task using WebKB datasets which contain 4199 web sites and 23830 words in a vocabulary. Twitter-LDA (Zhao et al., 2011) has been presented to estimate more coherent topic from tweets which consist of less than 140 letters. In Twitter-LDA model, it is hypothesized that one tweet is regarded to be generated from one topic such as UM. Twitter-LDA is tested using over 1 million tweets which have over 20000 words in a vocabulary.

There are several benefits of using probabilistic generative models for sentence clustering as described in section 1. However, these probabilistic generative models need much amount of datasets to get consistent computation results. In our experiments, we used about 4000 sentences of reviews which are the same number of documents as in WebKB datasets. However, there are few words in a vocabulary since a sentence of reviews has fewer words than a website. Therefore, in UM and LDA, the computation results seriously depended on randomly-initialized parameters, and lower clustering accuracy was obtained than PRTM in our experiment. To get consistent computation results from short sentence corpus with probabilistic generative models, over 1 million sentences are needed for like the experiment in Twitter-LDA. However, our proposed method, PageRank Topic Model (PRTM), can get consistent multinomial distributions over topics and words with few datasets because the network structure analysis methods are not dependent on randomly-initialized parameters. Therefore, PRTM achieved higher sentence clustering accuracy than UM and LDA with few review datasets.

5 Conclusion

In this paper, we have presented PageRank Topic Model (PRTM) to estimate a multinomial distribution over topics θ and words ϕ applying the network structure analysis methods and the convex quadratic programming problem to Word -Co-occurrence Graphs. With PRTM, the consistent computation results can be obtained because PRTM is not denpendent on randomly-initialized θ and ϕ. Furthermore, compared to other approaches at the task of estimations of the appropriate number of topics, PRTM estimated more appropriate number of topics for extracting the viewpoints from reviews datasets.

Acknowledgments

This research was partially supported by Core Research for Evolutionary Science and Technology (CREST) of Japan Science and Technology Agency (JST).

References

Aaron Clauset, Mark EJ Newman and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (methodological)*, 39(1): 1–38.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2008. A Co-occurrence Graph-based Approach for Personal Name Alias Extraction from Anchor Texts. *In Proceedings of International Joint Conference on Natural Language Processing*: 865–870.

David M. Blei, Andrew Y. Ng and Michael I. Jordan. 2003. Latent Dirichlet allocation. *the Journal of machine Learning research*, 3: 993–1022.

Juan Cao, Tian Xia, Jintao Li, Yongdong Zhang, and Sheng Tang. 2009. A density-based method for adaptive LDA model selection. *Neurocomputing*, 72: 1775–1781.

Jun S. Liu. 1994. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427): 958–966.

Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3): 61–67.

Kavita Ganesan and ChengXiang Zhai. 2011. Opinion-Based Entity Ranking. *Information Retrieval*.

Larry Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. *Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120*.

Mikio Yamamoto and Kugatsu Sadamitsu. 2005. Dirichlet Mixtures in Text Modeling. *CS Technical report CS-TR-05-1, University of Tsukuba, Japan*.

Sergey Brin and Larry Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(17):107–117.

Takeshi Sakaki, Yutaka Matsuo, Koki Uchiyama and Mitsuru Ishizuka 2007. Construction of Related Terms Thesauri from the Web. *Journal of Natural Language Processing*, 14(2):3–31.

Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. *The annual European Conference on Information Retrieval*:338–349.

Yee W. Teh, David Newman, and Max Welling. 2006. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. *In Advances in Neural Information Processing Systems*: 1353–1360.

Ying Zhao and George Karypis. 2001. Criterion functions for document clustering: Experiments and analysis. *Technical Report TR 01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN*.

Yukio Ohsawa, Nels E. Benson, and Masahiko Yachida. 1998. KeyGraph: Automatic Indexing by Co-occurrence Graph based on Building Construction Metaphor. *In Proceedings of Advanced Digital Library Conference*: 12–18.