

# Constrained LDA for Grouping Product Features in Opinion Mining

Zhongwu Zhai<sup>†</sup> Bing Liu<sup>‡</sup> Hua Xu<sup>†</sup> Peifa Jia<sup>†</sup>

<sup>†</sup>State Key Lab of Intelligent Tech. & Sys., Tsinghua National Lab for Info. Sci. and Tech., Dept. of Comp. Sci. & Tech., Tsinghua Univ.  
zhaizhongwu@gmail.com

<sup>‡</sup>Dept. of Comp. Sci., University of Illinois at Chicago  
liub@cs.uic.edu

**Abstract.** In opinion mining of product reviews, one often wants to produce a summary of opinions based on product features/attributes. However, for the same feature, people can express it with different words and phrases. To produce an effective summary, these words and phrases, which are domain synonyms, need to be grouped under the same feature. Topic modeling is a suitable method for the task. However, instead of simply letting topic modeling find groupings freely, we believe it is possible to do better by giving it some pre-existing knowledge in the form of automatically extracted constraints. In this paper, we first extend a popular topic modeling method, called LDA, with the ability to process *large* scale constraints. Then, two novel methods are proposed to extract two types of constraints automatically. Finally, the resulting *constrained-LDA* and the extracted constraints are applied to group product features. Experiments show that *constrained-LDA* outperforms the original LDA and the latest mLSA by a large margin.

**Keywords:** Opinion Mining, Product Feature Grouping, Constrained LDA

## 1 Introduction

One form of opinion mining in product reviews is to produce a feature-based summary [19]. In this model, product features are first identified, and positive and negative opinions on them are aggregated to produce a summary of opinions on the features. Product features are attributes and components of products, e.g., “picture quality” and “battery life” of a digital camera.

In reviews (or any writings), people often use different words and phrases to describe the same product feature. For example, “picture” and “photo” refer to the same feature for cameras. Grouping such synonyms is critical for opinion summary. Although WorldNet and other thesaurus dictionaries can help to some extent, they are insufficient because many synonyms are domain dependent. For example, “movie” and “picture” are synonyms in movie reviews, but they are not synonyms in camera reviews as “picture” is more likely to be synonymous to “photo” while “movie” to “video”. This paper deals with this problem, i.e., grouping domain synonym features. We assume that all the feature expressions have been identified by an existing algorithm [20-25, 29, 31, 36].

Topic modeling is a principled approach to solving this problem as it groups terms of the same topic into one group. This paper takes this approach. However, we believe instead of letting a topic modeling method to run completely unsupervised, some pre-existing knowledge

can be incorporated into the process to produce better results. The pre-existing knowledge can be inputted manually or extracted automatically. Here we extract such knowledge automatically.

Topic modeling methods can be seen as clustering algorithms that cluster terms into homogeneous topics (or clusters). In the classic clustering research in data mining, there is a class of semi-supervised clustering algorithms which allow constraints to be set as prior knowledge to restrict or to guide clustering algorithms to produce more meaningful clusters to human users [3, 38]. These constraints are in the forms of *must-links* and *cannot-links*. A must-link constraint specifies that two data instances must be in the same cluster. A cannot-link constraint specifies that two data instances cannot be in the same cluster.

In this paper, we incorporate these two types of constraints into the popular topic modeling method Latent Dirichlet Allocation (LDA) to produce a semi-supervised LDA method, called *constrained-LDA*. To the best of our knowledge, this is the first constrained LDA model which can process large scale constraints in the forms of must-links and cannot-links. There are two existing work by Andrzejewski and Zhu [1, 2] that are related to the proposed model. However, [1] only considers must-link constraints. In [2], the number of maximal cliques grow *exponentially* in the process of encoding constraints. Thus, [2] cannot process a large number of constraints (see Section 2.1). As we will also see in Section 2, our method of incorporating the two types of constraints is entirely different from the way that they did.

Although we call them must-link and cannot-link constraints, they are treated as “soft” rather than “hard” constraints in the sense that they can be violated or relaxed in the topic modeling process. The relaxation mechanism is needed because some constraints may not be correct especially when the constraints are extracted automatically. In our case, all constraints are extracted automatically with no human involvement. Thus, the constraints may be more appropriately called *probabilistic must-link and cannot-link* constraints.

On extracting must-link and cannot-link constraints for our application, we use two observations. First, we observed that a review sentence may comment on several product features, e.g., “*I like the picture quality, the battery life, and zoom of this camera*” and “*The picture quality is great, the battery life is also long, but the zoom is not good*”. From either of the sentences, we can see that the features, “picture quality”, “battery life” and “zoom” are unlikely to be synonyms or belonging to the same topic simply because people normally will not repeat the same feature in the same sentence. This observation allows us to extract many cannot-link constraints automatically. As for must-links, we observed that two noun phrases that shared one or more words are likely to fall into the same topic, e.g., “battery life” and “battery power”. Clearly, the two methods for identifying constraints are not perfect, i.e., they may find wrong constraints. The constraint relaxation mechanism comes to help to correct some of the cases.

In summary, this paper makes two main contributions:

- It proposes a general semi-supervised topic modeling method, called constrained-LDA. To our knowledge, this is the first time that a topic modeling method is enhanced with the ability to hand large scale must-link and cannot-link constraints.
- For our application of grouping product features. Two important observations were made, which allowed us to extract must-link and cannot-link constraints automatically.

Experiments show that the proposed constrained-LDA produces significantly better results than the original LDA and the latest mLSA [16] which also uses LDA.

## 2 Related Work

This study is related to two research areas, topic modeling and synonym grouping.

**Topic Modeling and LDA:** Blei *et al.* [5] proposed the original LDA using EM estimation. Griffiths and Steyvers [14] applied Gibbs sampling to estimate LDA’s parameters. Since these

works, many variations have been proposed [1, 2, 4, 6, 9, 10, 26, 27, 29, 30, 32, 37, 40]. In this paper, we only focus on the variations that add *supervised information* in the form of latent topic assignments.

Blei and McAuliffe [4] introduced a *supervised* latent Dirichlet allocation (sLDA). In sLDA, the authors added to LDA a response variable associated with each document, such as document's class label or document's rating. Ramage *et al.* [32] proposed a *labeled* LDA which considers the tag information of the documents. Chang and Blei [9] developed a relational topic model by adding the link information between documents. All these studies improve LDA by adding the labeled information of documents, whereas our constrained-LDA adds supervision to individual terms.

In [1], predefined topic-in-set knowledge (which means predefined terms for certain topics) was added to supervise the topic assignment for individual terms. Compared with our model, their model only used the *must-link* knowledge, not *cannot-links*. Moreover, our model's "topic-in-set knowledge" is updated dynamically after each Gibbs sampling, rather than fixed as predefined. Probability information is also introduced to the "topic-in-set knowledge".

In [2], must-link and cannot-link constraints were encoded with a Dirichlet Forest and were further incorporated into LDA. However, their model has a fatal limitation, as illustrated in Section 3.3 of [2], namely, the number of maximal cliques  $Q^{(r)}$  in a connected component of cannot-links' complementary graph can grow *exponentially*  $O(3^{|r|/5})$ , where  $|r|$  is the size of cannot-links' complementary graph. In our experiments (see Section 5), when **1/20** constraints in Table 2 are used,  $Q^{(r)}$  are 992 and 432 on camera and phone data sets, respectively; when **1/5** constraints are used,  $Q^{(r)}$  grow to 3,019,414 and 3,254,972, and then the program, downloaded from [2] authors' website<sup>1</sup>, crashed our server computer (2 Quad-Core AMD Opteron Processors, 2.70 GHz, 16GB Memory).

**Synonyms Grouping:** In [8], the authors proposed a method based on several similarity metrics to map discovered feature expressions to features in a given feature taxonomy of a domain. This is very different from our work as we do not have predefined feature taxonomy. The proposed method produces groupings automatically. [28] grouped product features using WordNet synonyms with poor results. [6] extracted and clustered semantic properties of *reviews* based on pros/cons annotations, which is different from our work of grouping *product features* (also we do not have pros/cons). In [39], a semi-supervised learning method is used. However, it requires the user to provide labeled examples, whereas this study does not need any pre-labeled examples. It thus solves a different problem.

In [16], product features were grouped using a multilevel latent semantic association technique, called *mLSA*. At the first level, all the words in product feature terms (each feature term can have more than one word) were grouped into a set of concepts/topics using LDA. The results are used to build latent topic structures for product feature terms. For example, we have four feature terms "day photos", "day photo", "daytime photos" and "daytime photo". If LDA groups the individual words "day" and "daytime" into topic10, and "photo" and "photos" into topic12, the system will group all four features into one group, call it "topic10-topic12", which is called a latent topic structure. At the second level, feature terms are grouped by LDA according to their latent topic structures produced at level 1 and context snippets in reviews. Following the above example, "day photos", "day photo", "daytime photos" and "daytime photo" in "topic10-topic12" combined with their surrounding words form a document. LDA runs such documents to produce the final result. The core idea of [16] is to re-organize and transform the input data for topic modeling, whereas we use the original reviews as the input. At any level of their multilevel algorithm the original LDA is directly applied. We propose constrained-LDA.

---

<sup>1</sup> [http://pages.cs.wisc.edu/~andrzej/research/df\\_lda.html](http://pages.cs.wisc.edu/~andrzej/research/df_lda.html)

### 3 The Proposed Algorithm

The original LDA is a purely unsupervised model, ignoring any pre-existing domain knowledge. However, as it is known in the semi-supervised clustering research [3, 38], the pre-existing knowledge can guide clustering algorithms to produce better and/or more meaningful clusters. We believe that they can help LDA as well, which is essentially a clustering algorithm. In our application domain, the prior knowledge about product features can help group domain synonym features, as explained in Section 1. In this section, we first give an introduction to LDA and then present the proposed constrained-LDA which can use pre-existing knowledge expressed as must-link and cannot-link constraints.

#### 3.1 Introduction to LDA

Instead of treating each document as “a-bag-of-words” as in many models dealing with text documents, topic modeling assumes that a document is “a-bag-of-topics”, and the aim of topic modeling is to group each term in each document into a proper topic. A variety of probabilistic topic models have been proposed [1, 4, 5, 9, 12-15, 17, 18, 34, 35], and LDA is one of the most popular topic modeling methods [35]. Similar to other methods, LDA’s input is a *term*×*document* matrix, and it outputs the *document*-*topic* distribution  $\theta$  and *topic*-*word* distribution  $\phi$ .

In order to obtain the distributions  $\theta$  and  $\phi$ , two main algorithms were proposed, EM [5] and Gibbs Sampling [14]. In this study, we use the Gibbs Sampling. For Gibbs sampling based LDA, the most important process is the updating of topic for each term in each document according to the probabilities calculated using Equation 1.

$$P(z_i = k | w_i = v, \mathbf{z}_{-i}, \mathbf{w}_{-i}) \propto \frac{C_{vk}^{WT} + \beta}{\sum_v C_{vk}^{WT} + V\beta} \frac{C_{dk}^{DT} + \alpha}{\sum_k C_{dk}^{DT} + K\alpha} \quad (1)$$

where  $z_i = k$  represents the assignment of the  $i^{\text{th}}$  term in a document to topic  $k$ ,  $w_i = v$  represents that the observed term  $w_i$  is the  $v^{\text{th}}$  term in the vocabulary of the text corpus, and  $\mathbf{z}_{-i}$  represents all the topic assignments excluding the  $i^{\text{th}}$  term.  $C_{vk}^{WT}$  is the number of times that term  $v$  is assigned to topic  $k$ , and  $C_{dk}^{DT}$  is the number of times that topic  $k$  has occurred in document  $d$ . Furthermore,  $K$  is the number of topics (which is an input given by the user),  $V$  is the size of the vocabulary,  $\alpha$  and  $\beta$  are the hyper-parameters for the document-topic and topic-word Dirichlet distributions, respectively. ( $\alpha$  and  $\beta$  are set to  $50/K$  and  $0.01$  by default.)

After  $N$  iterations of Gibbs sampling for all terms in all documents, *document*-*topic* distribution  $\theta$  and *topic*-*word* distribution  $\phi$  are finally estimated using Equations 2 and 3.

$$\theta_{dk} = \frac{C_{dk}^{DT} + \alpha}{\sum_k C_{dk}^{DT} + K\alpha} \quad (2)$$

$$\phi_{vk} = \frac{C_{vk}^{WT} + \beta}{\sum_v C_{vk}^{WT} + V\beta} \quad (3)$$

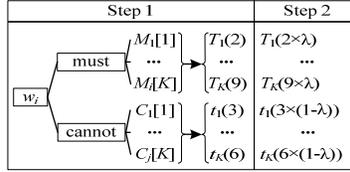
#### 3.2 Constrained-LDA

For constrained-LDA, constraints from the existing knowledge are added, and each term in the constraints is assumed to belong to only one topic. Compared with LDA, constrained-LDA has two more inputs, a set of must-link constraints and a set of cannot-link constraints. Recall a must-link constraint specifies that two terms should be in the same topic, and a cannot-link constraint specifies that two terms should not be in the same topic. The main idea of the

proposed approach is to revise the topic updating probabilities computed by LDA using the probabilities induced from the constraints. That is, in the topic updating process (shown in Equation 1), we compute an additional probability  $q(z_i=k)$  from the must-links and cannot-links for every candidate topic in  $\{1, 2, \dots, K\}$ , and then multiply it to the probability calculated by the original LDA model as the final probability for topic updating, see Equation 4.

$$P(z_i = k | w_i = v, \mathbf{z}_{-i}, \mathbf{w}_{-i}) \propto q(z_i = k) \frac{C_{vk}^{WT} + \beta}{\sum_{v'} C_{v'k}^{WT} + V\beta} \frac{C_{dk}^{DT} + \alpha}{\sum_{k'} C_{dk'}^{DT} + K\alpha} \quad (4)$$

As illustrated by Equations 1 and 4,  $q(z_i=k)$  plays a key role in constrained-LDA, because  $q(z_i=k)$  represents intervention or help from pre-existing knowledge of must-links and cannot-links. In this study,  $q(z_i=k)$  is computed as follows: For the given term  $w_i$ , if  $w_i$  is not constrained by any must-links or cannot-links,  $\{q(z_i=k) | k=1, \dots, K\}=1$ ; otherwise,  $\{q(z_i=k) | k=1, \dots, K\}$  is calculated using the following 4 steps in Figures 1 and 2.



**Fig.1.** Computing the weights for must-topics and cannot-topics

**Step 1** - get the must-topics and cannot-topics weights of  $w_i$ . Here must-topics mean the topics that the term  $w_i$  should be grouped into, while cannot-topics mean the topics that the term  $w_i$  should not be grouped into. For the given term  $w_i$ , its must-linked and cannot-linked *terms* are first found by querying must-links and cannot-links stores. Second, the *topics* of these terms are further obtained from the topic modeling. Then, we can obtain  $w_i$ 's must-topics and cannot-topics weights.

For example,  $w_i$ 's must-linked and cannot-linked *terms* are  $M_1, M_2$  and  $C_1, C_2, C_3$  respectively. Furthermore,  $M_1, M_2$  and  $C_1, C_2, C_3$  are assigned to topic  $k$  by LDA (denoted by  $M_1[k], M_2[k]$  and  $C_1[k], C_2[k], C_3[k]$ ). So, for topic  $k$ ,  $w_i$ 's must-topics and cannot-topics weights are  $weight(w_i, T_k(\{M_1, M_2\}))=weight(w_i, T_k(2))=2$  and  $weight(w_i, t_k(\{C_1, C_2, C_3\}))=weight(w_i, t_k(3))=3$ , respectively. Here,  $weight(w_i, T_k)$  or  $weight(w_i, t_k)$  is the weight that  $w_i$  should or should not be assigned to topic  $k$ ;  $T_k(2)$  represents there are 2 linked terms being assigned to topic  $k$  in the must category, and  $t_k(3)$  represents there are 3 linked terms being assigned to topic  $k$  in the cannot category.

**Step 2** - adjust the relative influences between *must-link category* and *cannot-link category*. In extracting the two types of constraints, the qualities of must-links and cannot-links may be different from each other. We use a damping factor  $\lambda$  to adjust the relative influences based on the constraint qualities. Specifically, all the must-topics' weights are multiplied by  $\lambda$ , while the cannot-topics' weights are multiplied by  $(1-\lambda)$ .

Following the above example,  $T_k(2)$  is adjusted to  $T_k(2 \times \lambda)$  while  $t_k(3)$  to  $t_k(3 \times (1-\lambda))$ . In this study, the default value of  $\lambda$  is empirically set to 0.3 (see Section 5.6).

Based on the results of above two steps, Steps 3 and 4 are further proposed to convert the *weights* of must-topics and cannot-topics to  $\{q(z_i=k) | k=1, \dots, K\}$ , as shown in Figure 2.

**Step 3** - aggregate the weights for each candidate topic. For the given term  $w_i$ , its candidate topics can fall into one of the three types, must-topics, unconstrained topics and cannot-topics. Recall must-topics mean the topics that  $w_i$  should be assigned to while cannot-link means the topics that  $w_i$  should not be assigned to. Thus, for calculating the probability that  $w_i$  will be assigned to candidate topic  $k$ , if  $k$  is in must-topics, we add  $weight(w_i, T_k)$  to  $q(z_i=k)$  in order to enhance the probability that  $w_i$  is assigned to topic  $k$ ; if  $k$  is in cannot-topics, we subtract  $weight(w_i, t_k)$  to  $q(z_i=k)$  in order to decrease the probability that  $w_i$  is assigned to topic  $k$  (lines 2 to 6 in Figure 2).

---

**Input:**  $w_i$ ;  
 $w_i$ 's must-topics' weights:  $\text{weight}(w_i, T_k), k=1,2,\dots,K$ ;  
 $w_i$ 's cannot-topics' weights:  $\text{weight}(w_i, t_k), k=1,2,\dots,K$ ;

**Output:**  $\{q(z_i=k)|k=1,2,\dots,K\}$

1. Initial all  $\{q(z_i=k)|k=1,2,\dots,K\}$  to zero
2. //Step 3 - Aggregate
3. **for** ( $k$  in  $\{1,2,\dots,K\}$ )
4.   **if** ( $k$  in  $\{w_i$ 's must-topics  $\}$ )  $q(z_i=k) += \text{weight}(w_i, T_k)$
5.   **if** ( $k$  in  $\{w_i$ 's cannot-topics  $\}$ )  $q(z_i=k) -= \text{weight}(w_i, t_k)$
- 6.
7. //Step 4 - Normalize and relax
8.  $\max = \{q(z_i=k)|k=1,2,\dots,K\}_{\max}$
9.  $\min = \{q(z_i=k)|k=1,2,\dots,K\}_{\min}$
10. **for** ( $k$  in  $\{1,2,\dots,K\}$ )
11.    $q(z_i = k) = \frac{q(z_i=k) - \min}{\max - \min}$
12.    $q(z_i = k) = q(z_i = k) \times \eta + (1 - \eta)$

---

**Fig. 2.** Probability aggregation and relaxation

In the above example, for the candidate topic  $k$ , the weight  $q(z_i = k)$  is:  $0 + \text{weight}(w_i, T_k(2 \times \lambda)) - \text{weight}(w_i, t_k(3 \times (1 - \lambda))) = 2 \times \lambda - 3 \times (1 - \lambda) = 5\lambda - 3$ .

**Step 4** - normalize and relax the weight of each candidate topic. Since the constraints are not guaranteed to be correct especially when the constraints are extracted automatically, there should be a parameter to adjust the constraint's strength to the model according to the quality of the constraints. When the constraints are completely correct, the model should treat these constraints as hard-constraints; when the constraints are all wrong, the model should discard them. In order to achieve this aim,  $\{q(z_i=k) | k = 1, \dots, K\}$  are adjusted by the relaxation factor  $\eta$  as follows:

Before being relaxed,  $\{q(z_i=k)|k=1,\dots,K\}$  are normalized to  $[0, 1]$  using Equation 5 (lines 8 to 11 in Figure 2). In Equation 5, max and min represent the maximum and minimum values of  $\{q(z_i=k)|k=1,\dots,K\}$ , respectively.

$$q(z_i = k) = \frac{q(z_i = k) - \min}{\max - \min} \quad (5)$$

Then,  $\{q(z_i=k)|k=1,\dots,K\}$  are relaxed by the relaxation factor  $\eta$  based on Equation 6 (line 12 in Figure 2). The default value of  $\eta$  is set to 0.9 in our study (see the evaluations in Section 5.6).

$$q(z_i = k) = q(z_i = k) \times \eta + (1 - \eta) \quad (6)$$

Note that, for our application of grouping product features, each product feature is considered as a term. Moreover, only  $\phi$  needs to be estimated by Equation 3 to output a set of topics and each topic contains a set of terms which belong to the topic.

## 4 Constraint Extraction

We now come back to our application and discuss how to extract constraints automatically. The general idea has been discussed earlier. For completeness, we briefly discuss them here again.

**Must-link:** If two product features  $f_i$  and  $f_j$  share one or more words, we assume them to form a must-link, i.e., they should be in the same topic, e.g., "battery power" and "battery life". Clearly, this method is not perfect. Then, the constraint relaxation mechanism comes to help.

**Cannot-link:** If two product features  $f_i$  and  $f_j$  occur in the same sentence and they are not connected by "and", the two features form a cannot-link. The reason for this is that people usually do not repeat the same feature in the same sentence. Features linked by "and" are not used as our experience showed that "and" can be quite unsafe. It frequently links features from the same topic, especially product names based features.

## 5 Experimental Evaluation

In this Section, we evaluate the proposed constrained-LDA model in a variety of settings, and compare it with the original LDA algorithm and the recent multilevel mLSA method for solving the same problem. We do not compare with the similarity based method in [11] because their technique requires a given feature taxonomy, which we do not use.

### 5.1 Data Sets

In order to demonstrate the generality of the proposed algorithm, experiments have been conducted in two domains: digital camera and cell phone. We used two data sets with feature annotations from the Customer Review Datasets<sup>2</sup>, which have been widely used by researchers for opinion mining. We selected the reviews for digital cameras and cell phones. Their feature annotations are used in our system. Since these two data sets are too small for topic modeling, we crawled many other camera and phone reviews from Amazon.com. The details of each data set are given in Table 1.

**Table 1.** Summary of the data sets

<b>Camera</b>	Number of Reviews	2,400	<b>Phone</b>	Number of Reviews	1,315
	Number of Sentences	20,628		Number of Sentences	18,393
	Number of Vocabulary	7,620		Number of Vocabulary	7,376

### 5.2 Gold Standard

Since the product features in the Customer Review Datasets have already been annotated by human annotators, these annotated product features are grouped manually to form a gold standard for each data set. For the digital camera data set, we group the features into 14 topics, according to the camera’s taxonomy published by Active Sales Assistant<sup>TM</sup>, a product of Active Decisions, which is one of the leading providers of Guided Selling Solutions, and is available at [www.activebuyersguide.com](http://www.activebuyersguide.com) [8]. For the cell phone data set, the topics published by Google products are adopted, and all the cell phone features are grouped into 9 topics.

### 5.3 Evaluation Measure

The performance of our product features grouping algorithm is evaluated using Rand Index [33], which has been used by several researchers [6, 7, 38]. Rand Index is also the evaluation measure used in [16]. We will compare our method with their mLSA method in Section 5.5.

Rand Index allows for a measure of agreement between two partitions,  $P_{answer}$  and  $P_{machine}$ , of the same instance set  $D$ . Each partition is viewed as a collection of  $C_n^2$  pair-wise decisions, where  $n$  is the size of  $D$ . For each decision of two instances  $I_j$  and  $I_k$  in  $D$ ,  $P_{answer}$  and  $P_{machine}$  either assigns  $I_j$  and  $I_k$  to the same cluster or to different clusters. Let  $a$  be the number of correct decisions where  $I_j$  and  $I_k$  are in the same cluster in both  $P_{answer}$  and  $P_{machine}$ . Let  $b$  be the number of correct decisions where the two instances are placed in different clusters in both partitions. The total agreement can be calculated using Equation 7. In our study, all the product features make up the instance set  $D$ ; the gold standard is the  $P_{answer}$ ; the experimental result is the  $P_{machine}$ .

$$RI(P_{answer}, P_{machine}) = \frac{a + b}{C_n^2} = \frac{a + b}{n \times (n - 1) / 2} \quad (7)$$

<sup>2</sup> <http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>

## 5.4 Compared with LDA

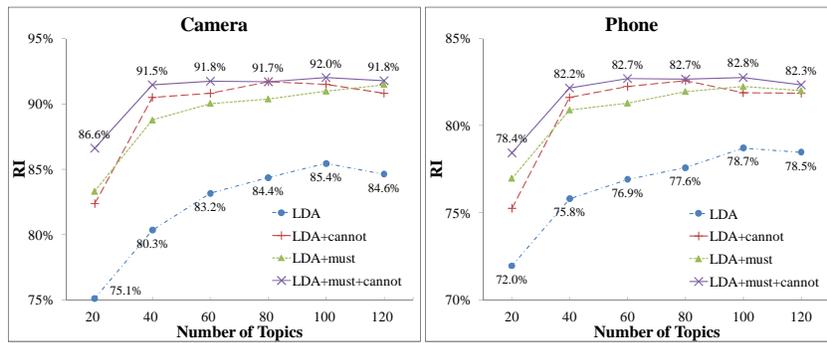
[2] proposed the most recent LDA model (called DF-LDA) that can consider must-link and cannot-link constraints. However, as explained in Section 2.1, DF-LDA *cannot* process a large number of constraints. When only 1/5 constraints in Table 2 are used, DF-LDA1 crashes the system. Thus DF-LDA cannot be applied to our task of grouping a large number of product features. Due to DF-LDA's limitation, we only report the comparison results with the original LDA.

Both the original LDA and the proposed constrained-LDA were run using different numbers of topics, 20, 40, 60, 80, 100 and 120, in the two domains. Note that LDA requires the number of topics to be specified by the user. Note also we do not report the results of using the original numbers of topics (14 and 9) for the two data sets as they were poorer (see the trends in Figure 3). Using only must-links, only cannot-links, and their combination were all experimented. The number of constraints extracted from each data set is given in Table 2, which are the number of unique pairs (pair (a, b) and pair (b, a) are the same in our case). All the results are shown in Figure 3. From Figure 3, we can see that the patterns are about the same for different methods on different data sets, which show that the results are consistent. Below we make some additional observations:

- All the constrained methods (*LDA+cannot*, *LDA+must* and *LDA+must+cannot*) perform much better than the original LDA model (*LDA*). For smaller numbers of topics, the improvements were more than 10% for the digital camera corpus, and around 7% for the cell phone corpus. With more topics, the improvements are slightly less, but still 7% for the digital camera and 4% for the cell phone.
- Both cannot-links (*LDA+cannot*) and must-links (*LDA+must*) perform well, although cannot-links are slightly more effective than must-links on average. This phenomenon indicates that our assumption about cannot-link is reasonable and the quality of the extracted cannot-links is good. When the number of topics is small or large, the must-links are slightly better than cannot-links. We believe the reason is that in these two ends, cannot-link terms were either forced into the same topics (for a small number of topics), or easily spread into too many topics. The original LDA also shows this behavior, which is fairly easy to understand.
- The combination of must-links and cannot-links (*LDA+must+cannot*) consistently outperforms each individual type of constraints alone (*LDA+cannot* and *LDA+must*). Although the margins of improvements were not very large, they were consistent. This also

**Table 2.** Number of the extracted constraints

Camera	Number of Must-links	300	Phone	Number of Must-links	184
		Number of Cannot-links		5172	



**Fig. 3.** RI results of constrained-LDA and the original LDA

indicates that the must-link and cannot-link constraints are already quite effective individually.

- In practice, it is often more effective to use a smaller number of topics, which are easy to understand and to handle by the users. In both cases, 40 topics seem to be optimal.

In summary, we can see that unsupervised topic modeling can be improved by adding must-link and cannot-link constraints. *Note that each feature expression is considered as a term in all our experiments.*

### 5.5 Comparing with mLSA

As mentioned earlier, the recent multilevel latent semantic association method *mLSA* [16] solves the same problem as we do. We reviewed this method in the related work section. It was shown that *mLSA* (which applies LDA) performs better than the existing methods, e.g., LDA-based and Kmeans-based algorithm. We thus only compare the proposed *constrained-LDA* with *mLSA*, but not other existing methods. The comparisons are made based on both the digital camera corpus and the cell phone corpus. The results are shown in Figure 4. We only used 40 topics, which appeared to be the optimal number among our tested topic numbers in Figure 3.

As demonstrated in Figure 4, *mLSA* (2: red bar) achieves encouraging results by transforming the input document content before applying LDA. Our constrained-LDA model does not make any efforts to re-organize or transform the input document content, and our input is the set of original reviews. However, the results produced by constrained-LDA (*LDA+cannot*, *LDA+must* and *LDA+must+ cannot*) are all substantially better than those of *mLSA*. This observation shows the positive influence of constraints.

### 5.6 Influence of Parameters

Compared to the original LDA model, the proposed constrained-LDA has two additional parameters, i.e., damping factor  $\lambda$  and relaxation factor  $\eta$ , and as mentioned in Section 3.2. In this section, we discuss their influences to the overall performance.

**Influence of the damping factor –  $\lambda$ :** Recall that damping factor  $\lambda$  is used to adjust the relative influences of must-links and cannot-links on the proposed model. In Figure 5,  $\lambda=0$  means the proposed model is only constrained by cannot-links, whereas  $\lambda=1$  means that the proposed model is only constrained by must-links. That is, larger  $\lambda$  values mean more influences of must-links and less influence of cannot-links.

As shown in Figure 5, with increased influence of must-links over cannot-links, the

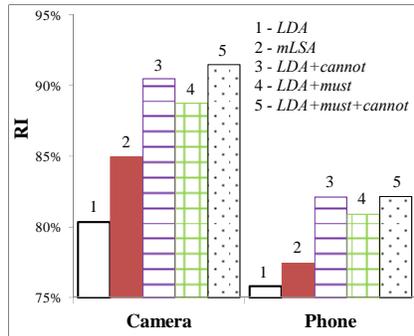


Fig. 4. Comparisons with mLSA

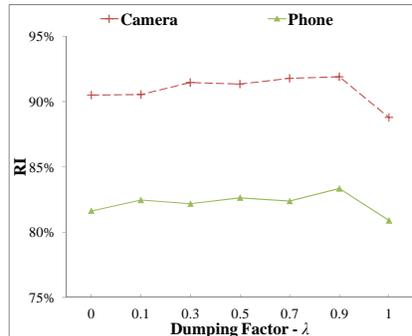


Fig. 5.  $\lambda$ 's influence on the overall performance

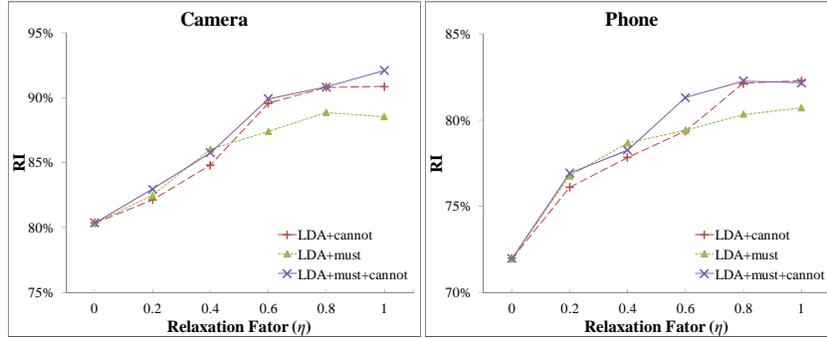


Fig. 6.  $\eta$ 's influence on the overall performance (#topics = 40)

performance of constrained-LDA improves slightly. However, when there is only must-links ( $\lambda=1$ ), the performance drops sharply to the lowest point. This illustrates the synergetic effect of must-links and cannot-links: they help each other.

Since the  $\lambda$  values after 0.3 produce very similar results, we used  $\lambda = 0.3$  as the default for  $\lambda$ . The experimental results in Figures 3 and 4 all used this default damping factor.

**Influence of the relaxation factor -  $\eta$ :** In this study, the relaxation factor  $\eta$  represents the strength of the constraints on the LDA model. When  $\eta=0$ , it means that no constraint is added to the LDA model. Then, constrained-LDA reduces to the original LDA. When  $\eta=1$ , it means that both must-link and cannot-link constraints become hard constraints and cannot be violated. The influence of  $\eta$  on the overall performance is shown in Figure 6.

As shown in Figure 6, with the growth of the strength of the constraints, the performances of *LDA+cannot*, *LDA+must* and *LDA+must+cannot* increase considerably. This observation not only shows that the constraints clearly help the performances of topic modeling (or LDA), but also shows that the qualities of the extracted must-links and cannot-links are quite good, especially the extracted cannot-links.

In fact, using both must-link and cannot-link constraints, when  $\eta = 1$ , the results are the best for both the digital camera data and cell phone data. We use  $\eta = 0.9$  as the default in the system as in general one may not be able to extract very high quality constraints. In our experiments reported earlier in Figures 3, 4 and 5, the default  $\eta = 0.9$  was used.

## 6 Conclusion and Future Work

This paper enhanced the popular topic modeling method LDA with the ability to consider existing knowledge in the form of must-link and cannot-link constraints. The resulting method is called *constrained-LDA*. Since the strength of constraints and the relative influences of must-links and cannot-links are designed to be adjustable, the proposed model is flexible to a variety of applications. Our experiment results show that our chosen default values perform quite well.

In our application, we experimented with two opinion mining data sets to group product feature synonyms. Constrained-LDA outperformed the existing methods by a large margin, which showed that constraints as prior knowledge can help unsupervised topic modeling.

Moreover, this paper also proposed two methods to extract the two types of constraints automatically. Experimental results showed that their qualities were high (see Figure 6).

In our future work, more techniques will be developed to expand the extracted must-links and cannot-links, and to further improve the accuracy of grouping product features.

## References

- [1] Andrzejewski D and Zhu X. Latent Dirichlet Allocation with topic-in-set knowledge. in *Proceedings of NAACL HLT*. 2009.43-48
- [2] Andrzejewski D, Zhu X, and Craven M. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. in *Proceedings of ICML*. 2009.25-32
- [3] Basu S, Davidson I, and Wagstaff K. Constrained clustering: Advances in algorithms, theory, and applications. 2008: Chapman & Hall/CRC.
- [4] Blei D and McAuliffe J. Supervised topic models. *Advances in Neural Information Processing Systems*, 2008. 20: 121-128.
- [5] Blei D, Ng A Y, and Jordan M I. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 2003. 3(3): 993-1022.
- [6] Branavan S R K, Chen H, Eisenstein J, and Barzilay R. Learning document-level semantic properties from free-text annotations. in *Proceedings of ACL*. 2008.569-603
- [7] Cardie C and Wagstaff K. Noun phrase coreference as clustering. in *Proceedings of the Eleventh National Conference on Artificial Intelligence*. 1999.82-89
- [8] Carenini G, Ng R, and Zwart E. Extracting knowledge from evaluative text. in *Proceedings of International Conference on Knowledge Capture*. 2005.11-18
- [9] Chang J and Blei D. Relational topic models for document networks. in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics(AISTATS)*. 2009. Clearwater Beach, Florida, USA
- [10] Chemudugunta C, Holloway A, Smyth P, and Steyvers M, Modeling documents by combining semantic concepts with unsupervised statistical learning. *The Semantic Web-ISWC 2008*: 229-244.
- [11] Clemons E K, Gao G G, and Hitt L M, When Online Reviews Meet Hyperdifferentiation: A Study of the Craft Beer Industry. *Journal of Management Information Systems*, 2006. 23(2): 149-171.
- [12] Griffiths T and Steyvers M. A probabilistic approach to semantic representation. in *In Proceedings of the 24th Annual Conference of the Cognitive Science Society*. 2002.381-386
- [13] Griffiths T and Steyvers M, Prediction and semantic association. *Advances in Neural Information Processing Systems*, 2003: 11-18.
- [14] Griffiths T and Steyvers M, Finding scientific topics. *Proceedings of the National Academy of Sciences*, 2004. 101(Suppl 1): 5228-5535.
- [15] Griffiths T, Steyvers M, Blei D, and Tenenbaum J, Integrating topics and syntax. *Advances in Neural Information Processing Systems*, 2005. 17: 537-544.
- [16] Guo H, Zhu H, Guo Z, Zhang X, and Su Z. Product feature categorization with multilevel latent semantic association. in *Proceedings of CIKM*. 2009.1087-1096
- [17] Hofmann T. Probabilistic latent semantic indexing. in *Proceedings of SIGIR*. 1999.50-57
- [18] Hofmann T, Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 2001. 42(1): 177-196.
- [19] Hu M and Liu B. Mining and summarizing customer reviews. in *Proceedings of SIGKDD*. 2004.168-177
- [20] Hu M and Liu B. Mining Opinion Features in Customer Reviews. in *Proceedings of AAAI*. 2004. San Jose, California.755-760
- [21] Jin W, Ho H, and Srihari R. OpinionMiner: a novel machine learning system for web opinion mining and extraction. in *Proceedings of KDD*. 2009.1195-1204
- [22] Kim S and Hovy E. Extracting opinions, opinion holders, and topics expressed in online news media text. in *Proceedings of EMNLP*. 2006.1065-1074
- [23] Kobayashi N, Inui K, and Matsumoto Y. Extracting aspect-evaluation and aspect-of relations in opinion mining. in *Proceedings of EMNLP*. 2007.1065-1074

- [24] Ku L, Ho H, and Chen H, Opinion mining and relationship discovery using CopeOpi opinion analysis system. *Journal of the American Society for Information Science and Technology*, 2009, 60(7): 1486-1503.
- [25] Ku L, Liang Y-T, and Chen H-H. Opinion Extraction, Summarization and Tracking in News and Blog Corpora. in *Proceedings of AAAI*. 2006.100-107
- [26] Lacoste-Julien S, Sha F, and Jordan M. DiscLDA: Discriminative learning for dimensionality reduction and classification. in *Proceedings of NIPS*. 2008
- [27] Li W and McCallum A. Pachinko allocation: DAG-structured mixture models of topic correlations. in *Proceedings of ICML*. 2006.577-584
- [28] Liu B, Hu M, and Cheng J. Opinion Observer: Analyzing and Comparing Opinions on the Web. in *Proceedings of WWW*. 2005.342-351
- [29] Mei Q, Ling X, Wondra M, Su H, and Zhai C. Topic sentiment mixture: Modeling facets and opinions in weblogs. in *Proceedings of WWW*. 2007.171-180
- [30] Mimno D, Wallach H, Naradowsky J, Smith D, and McCallum A. Polylingual topic models. in *Proceedings of EMNLP*. 2009.880-889
- [31] Popescu A-M and Etzioni O. Extracting Product Features and Opinions from Reviews. in *Proceedings of EMNLP*. 2005.339-346
- [32] Ramage D, Hall D, Nallapati R, and Manning C. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. in *Proceedings of EMNLP*. 2009.248-256
- [33] Rand W, Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 1971. 66(336): 846-850.
- [34] Rosen-Zvi M, Griffiths T, Steyvers M, and Smyth P. The author-topic model for authors and documents. 2004: AUAI Press Arlington, Virginia, United States.487-494
- [35] Steyvers M and Griffiths T, Probabilistic topic models. *Handbook of Latent Semantic Analysis*, 2007: 424-440.
- [36] Stoyanov V and Cardie C. Topic identification for fine-grained opinion analysis. in *Proceedings of COLING*. 2008.817-824
- [37] Titov I and McDonald R. Modeling online reviews with multi-grain topic models. in *Proceedings of WWW*. 2008.111-120
- [38] Wagstaff K, Cardie C, Rogers S, and Schroedl S. Constrained k-means clustering with background knowledge. in *In Proceedings of ICML*. 2001.577-584
- [39] Zhai Z, Liu B, Xu H, and Jia P. Grouping Product Features Using Semi-supervised Learning with Soft-Constraints. in *Proceedings of COLING*. 2010
- [40] Zhu J, Ahmed A, and Xing E. MedLDA: Maximum margin supervised topic models for regression and classification. in *Proceedings of ICML*. 2009.1257-1264