A Joint Model of Product Properties, Aspects and Ratings for Online Reviews

Ying Ding
School of Information Systems
Singapore Management University
ying.ding.2011@smu.edu.sg

Jing Jiang
School of Information Systems
Singapore Management University
jingjiang@smu.edu.sg

Abstract

Product review mining is an important task that can benefit both businesses and consumers. Lately a number of models combining collaborative filtering and content analysis to model reviews have been proposed, among which the Hidden Factors as Topics (HFT) model is a notable one. In this work, we propose a new model on top of HFT to separate product properties and aspects. Product properties are intrinsic to certain products (e.g. types of cuisines of restaurants) whereas aspects are dimensions along which products in the same category can be compared (e.g. service quality of restaurants). Our proposed model explicitly separates the two types of latent factors but links both to product ratings. Experiments show that our proposed model is effective in separating product properties from aspects.

1 Introduction

Online product reviews and the numerical ratings that come with them have attracted much attention in recent years. During the early years of research on product review mining, there were two separate lines of work. One focused on content analysis using review texts but ignored users, and the other focused on collaborative filtering-based rating prediction using user-item matrices but ignored texts. However, these studies do not consider the identifies of reviewers, and thus cannot incorporate user preferences into the models. In contrast, the objective of collaborative filtering-based rating prediction is to predict a target user's overall rating on a target product without referring to any review text (e.g. Salakhutdinov and Mnih (2007)). Collaborative filtering makes use of past ratings of the target user, the target item and other user-item ratings to predict the target user's rating on the target item.

Presumably if review texts, numerical ratings, user identities and product identities are analyzed together, we may achieve better results in rating prediction and feature/aspect identification. This is the idea explored in a recent work by McAuley and Leskovec (2013), where they proposed a model called Hidden Factors as Topics (HFT) to combine collaborative filtering with content analysis. HFT combines latent factor models for recommendation with Latent Dirichlet Allocation (LDA). In the joint model, the latent factors play dual roles: They contribute to the overall ratings, and they control the topic distributions of individual reviews.

While HFT is shown to be effective in both predicting ratings and discovering meaningful latent factors, we observe that the discovered latent factors are oftentimes not "aspects" in which products can be evaluated and compared. In fact, the authors themselves also pointed out that the topics discovered by HFT "are not similar to aspects" (McAuley and Leskovec, 2013). Here we use "aspects" to refer to criteria that can be used to compare all or most products in the same category. For example, we can compare restaurants by how well they serve customers, so *service* is an aspect. But we cannot compare restaurants by how well they serve Italian food if they are not all Italian restaurants to begin with, so *Italian food* cannot be considered an aspect; It is more like a feature or property that a restaurant either possesses or does not possess.

Identifying aspects would help businesses see where they lose out to their competitors and consumers to directly compare different products under the same criteria. In this work, we study how we can modify the HFT model to discover both properties and aspects. We use the term "product properties" or simply "properties" to refer to latent

Proceedings of Recent Advances in Natural Language Processing, pages 131–137,
Hissar, Bulgaria, Sep 7–9 2015.

factors that can explain user preferences but are intrinsic to only certain products. Besides types of cuisines, other examples of properties include brands of products, locations of restaurants or hotels, etc. Since a product's rating is related to both the properties it possesses and how well it scores in different aspects, we propose a joint model that separates product properties and aspects but links both of them to the numerical ratings of reviews.

We evaluate our model on three data sets of product reviews. Based on human judgment, we find that our model can well separate product properties and aspects while at the same time maintaining similar rating prediction accuracies as HFT. In summary, the major contribution of our work is a new model that can identify and separate two different kinds of latent factors, namely product properties and aspects.

2 Related Work

Research on modeling review texts and the associated ratings or sentiments has attracted much attention. In the pioneering work by Hu and Liu (2004), the authors extracted product aspects and predicted sentiment orientations. While this work was mainly based on frequent pattern mining, recent work in this direction pays more attention to modeling texts with principled probabilistic models like LDA. Wang et al. (2011a) modeled review documents using LDA and treated ratings as a linear combination of topic-word-specific sentiment scores. Sauper et al. (2011) modeled word sentiment under different topics with a topic-sentiment word distribution. While these studies simultaneously model review documents and associated ratings, they do not consider user identity and item identity, which makes them unable to discover user preference and item quality. There have been many studies on the extraction of product aspects (Qiu et al., 2011; Titov and McDonald, 2008b; Mukherjee and Liu, 2012). These studies use either linguistic patterns or a topic modeling approach, or a combination of both, to identify product features or aspects. However, they do not distinguish between aspects and properties.

More recent work has started paying attention to taking user and product identity into consideration. McAuley and Leskovec (2013) used a principled model similar to that of Wang and Blei (2011) to map each latent factor to a topic learned by LDA from review documents. Two variations of this model were proposed by Bao et al. (2014), which also took each review's helpfulness score into consideration. The latest work in this direction is a model proposed by Diao et al. (2014). This work further modeled the generation of sentiment words in review text, which was controlled by the estimated sentiment score of the corresponding aspect. However, in all the work discussed above, there was no separation and joint modeling of product properties and aspects.

3 Model

In this section, we will describe our join model for product properties, aspects and ratings.

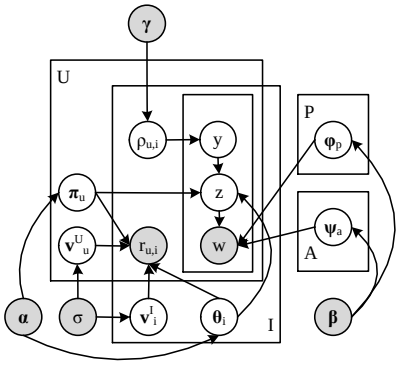

Figure 1: Plate notation of our PAR model. Circles in gray indicate hyperparameters and observations.

3.1 Our Model

3.1.1 Generation of Ratings

As we have pointed out in Section 1, many of the latent factors learned by HFT are product properties such as brands, which cannot be used to compare all products in the same category. In order to explicitly model both product properties and aspects, we first assume that there are two different sets of latent factors: There is a set of P product properties, and there is another set of A product aspects. Both are latent factors that will influence ratings.

Next, we assume that each product has a distribution over product properties and each user has a real-valued vector over product properties. Because properties generally model features that a product either possesses or does not possess, it makes sense to associate a distribution over properties with a product. For example, if each type of cuisines corresponds to a property, then a Mexican restaurant should have a high probability for

the property *Mexican food* but low or zero probabilities for properties such as *Japanese food*, *Italian food*, etc. On the other hand, a user may like and dislike certain product properties, so it makes sense to use real numbers that can be positive or negative to indicate a user's preferences over different properties. For example, if a user does not like Japanese food, she is likely to give low ratings to Japanese restaurants, and therefore it makes sense to model this as a negative value associated with the property *Japanese food* in her latent vector.

Analogically, it makes sense to assume that a product has a real-valued latent vector over aspects, where a positive value means the product is doing well in that aspect and a negative value means the product is poor in that aspect. For example, a restaurant may get a negative score for the aspect *service* but a positive score for the aspect *price*. On the other hand, we assume that a user has a distribution over aspects to indicate their relative weight when the user rates a product. For example, if service is not important to a user but price is, she will have a low or zero probability for the aspect *service* in her vector but a high probability for the aspect *price*.

Formally, let $\boldsymbol{\theta}_i$ denote the property distribution of product i, $\boldsymbol{v}_u^U$ denote the property vector of user u, $\boldsymbol{\pi}_u$ denote the aspect distribution of user u and $\boldsymbol{v}_i^I$ denote the aspect vector of item i. Based on the assumptions above, it makes sense to model the rating of user u given to item i to be close to $(\boldsymbol{\theta}_i \cdot \boldsymbol{v}_u^U + \boldsymbol{\pi}_u \cdot \boldsymbol{v}_i^I)$. If we compare this formulation with standard ways of modeling ratings such as in HFT, we can see that the major difference is the following. In standard models, the latent vectors of both users and items are unconstrained, i.e. both positive and negative values can be taken. This may cause problem interpreting the learned vectors. For example, when user u has a negative value for the k^{th} latent factor and item i also has a negative value for the k^{th} latent factor, the product of these two negative values results in a positive contribution to the rating of item i given by user u. But how shall we interpret these two negative values and their combined positive impact to the rating? In our model, we separate the latent factors into two groups. For one group of latent factors (product properties), we force the items to have non-negative values, while for the other group of latent factors (product aspects), we force the users to have non-negative values. By doing this, we improve the interpretability of the learned latent vectors.

3.1.2 Generation of Review Texts

In our model, for each latent factor, which can be either a product property or an aspect, there is a word distribution associated with it, which we denote by $\boldsymbol{\phi}_p$ for property p and $\boldsymbol{\psi}_a$ for aspect a.

We assume that a review of a product given by a particular user mainly consists of two types of information: properties this product possesses and evaluation of this product in the various aspects that this user cares about. Content related to product properties is mainly controlled by the property distribution of the product. For example, reviews on a Mexican restaurant may contain much information about Mexican food. Content related to aspects are mainly controlled by the user's aspect preference distribution. A user who values service more may comment more about a restaurant's service. Based on these assumptions, in the generative process of reviews, each word in a review document is sampled either from a product property or an aspect.

3.1.3 The Generative Process

Our model is shown in Figure 1. and the description of the generative process is as follows:

- For each product property p, sample a word distribution $\boldsymbol{\phi}_p \sim \text{Dirichlet}(\boldsymbol{\beta})$.
- For each aspect a, sample a word distribution $\boldsymbol{\psi}_a \sim \text{Dirichlet}(\boldsymbol{\beta})$.
- For each item
  - Sample a product property distribution $\boldsymbol{\theta}_i \sim \text{Dirichlet}(\boldsymbol{\alpha})$.
  - Sample an A-dimensional vector $\boldsymbol{v}_i^I$ where $v_{i,a}^I \sim \text{Normal}(0, \sigma^2)$.
  - Sample an item rating bias $b_i \sim \mathcal{N}(0, \sigma^2)$.
- For each user
  - Sample an aspect distribution $\boldsymbol{\pi}_u \sim \text{Dirichlet}(\boldsymbol{\alpha})$.
  - Sample a P-dimensional vector $\boldsymbol{v}_u^U$ where $v_{u,p}^U \sim \text{Normal}(0, \sigma^2)$.
  - Sample a user rating bias $b_u \sim \mathcal{N}(0, \sigma^2)$.
- For a user-item pair where a review and a rating exist
  - Sample the rating $r_{u,i} \sim \text{Normal}(\boldsymbol{\theta}_i \cdot \boldsymbol{v}_u^U + \boldsymbol{\pi}_u \cdot \boldsymbol{v}_i^I + b_i + b_u + b, \sigma^2)$
  - Sample the parameter for a Bernoulli distribution $\boldsymbol{\rho}_{u,i} \sim \text{Beta}(\boldsymbol{\gamma})$
  - For each word in the review
    - Sample $y_{u,i,n} \sim \text{Bernoulli}(\boldsymbol{\rho}_{u,i})$.
    - Sample $z_{u,i,n} \sim \text{Discrete}(\boldsymbol{\theta}_i)$ if $y_{u,i,n} = 0$ and $z_{u,i,n} \sim \text{Discrete}(\boldsymbol{\pi}_u)$ if $y_{u,i,n} = 1$.
    - Sample $w_{u,i,n} \sim \text{Discrete}(\boldsymbol{\phi}_{z_{u,i,n}})$ if $y_{u,i,n} = 0$ and $w_{u,i,n} \sim \text{Discrete}(\boldsymbol{\psi}_{z_{u,i,n}})$ if $y_{u,i,n} = 1$.

Here, $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are hyper-parameters for Dirichlet distribution, σ is the standard deviation for Gaussian distribution, $\boldsymbol{\rho}_{u,i}$ is the switching probability distribution for review of user u on item i, $y_{u,i,n}$ and $z_{u,i,n}$ are the switching variable and topic assignment for word at position n of review on itme i from user u. We refer to our model as the Property-Aspect-Rating (PAR) model.

3.2 Parameter Estimation

Our goal is to learn the parameters that can maximize the log-likelihood of both review texts and ratings simultaneously. Formally speaking, we are trying to estimate the parameters $\boldsymbol{V}^U$, $\boldsymbol{V}^I$, $\boldsymbol{B}_U$, $\boldsymbol{B}_I$, $\boldsymbol{\pi}_U$, $\boldsymbol{\theta}_I$, $\boldsymbol{\rho}$, $\boldsymbol{\phi}_P$ and $\boldsymbol{\psi}_A$ that can optimize the following posterior probability.

$$P(\boldsymbol{V}^U, \boldsymbol{V}^I, \boldsymbol{B}_U, \boldsymbol{B}_I, \boldsymbol{\pi}_U, \boldsymbol{\theta}_I, \boldsymbol{\rho}, \boldsymbol{\phi}_P, \boldsymbol{\psi}_A | \boldsymbol{W}, \boldsymbol{R}).$$

Here $\boldsymbol{V}^U$ and $\boldsymbol{V}^I$ refer to all latent vectors for items and users, $\boldsymbol{B}_U$ and $\boldsymbol{B}_I$ refer to all the bias terms, $\boldsymbol{W}$ refers to all the words in the reviews and $\boldsymbol{R}$ refers to all the ratings. The hyperparameters are omitted in the formula. Equivalently, we will use the loglikelihood as our objective function. As there is no closed form solution for it, we use Gibbs-EM algorithm (Wallach, 2006) for parameter estimation.

E-step: In the E-step, we fix the parameters $\boldsymbol{\pi}_U$ and $\boldsymbol{\theta}_I$ and collect samples of the hidden variables $\boldsymbol{Y}$ and $\boldsymbol{Z}$ to approximate the distribution $P(\boldsymbol{Y}, \boldsymbol{Z} | \boldsymbol{W}, \boldsymbol{R}, \boldsymbol{\pi}_U, \boldsymbol{\theta}_I)$.

M-step: In the M-step, with the collected samples of $\boldsymbol{Y}$ and $\boldsymbol{Z}$, we seek values of $\boldsymbol{\pi}_U$, $\boldsymbol{\theta}_I$, $\boldsymbol{V}^U$, $\boldsymbol{V}^I$, $\boldsymbol{B}_U$ and $\boldsymbol{B}_I$ that maximize the following objective function:

$$\mathcal{L} = \sum_{(\boldsymbol{Y}, \boldsymbol{Z}) \in \mathcal{S}} \log P(\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{R} | \boldsymbol{\pi}_U, \boldsymbol{\theta}_I, \boldsymbol{V}^U, \boldsymbol{V}^I, \boldsymbol{B}_U, \boldsymbol{B}_I)$$

where $\mathcal{S}$ is the set of samples collected in the E-step.

In our implementation, we perform 600 runs of Gibbs EM. Because Gibbs sampling is time consuming, in each run we only perform one iteration of Gibbs sampling and collect that one sample. We then have 60 iterations of gradient descent. The gradient descent algorithm we use is L-BFBS, which is efficient for large scale data set.

4 Experiments

In this section, we present the empirical evaluation of our model.

Data Set	#Reviews	#W/R	Voc	#Users	#Items
SOFT	54,330	84.6	16,653	43,177	8,760
MP3	20,689	103.9	8,227	18,609	742
REST	88,865	86.5	21,320	8,230	3,395

Table 1: Statistics of our data sets.*#W/R stands for #Word/Review.

4.1 Data

We use three different review data sets for our evaluation. The first one is a set of software reviews, which was used by McAuley and Leskovec (2013). We refer to this set as SOFT. The second one is a set of reviews of MP3 players, which was used by Wang et al. (2011b). We refer to this set as MP3. The last one is a set of restaurant reviews released by Yelp[1] in Recsys Challenge 2013[2], which was also used by McAuley and Leskovec (2013). We refer to it as REST. Based on common practice in previous studies (Titov and McDonald, 2008a; Titov and McDonald, 2008b; Wang and Blei, 2011), we processed these reviews by first removing all stop words and then removing words which appeared in fewer than 10 reviews. We then also removed reviews with fewer than 30 words. Some statistics of the processed data sets are shown in Table 1.

4.2 Experiment Setup

As we have discussed in Section 1, the focus of our study is to modify the HFT model to capture both product properties and aspects. Note that HFT model is designed for both predicting ratings and discovering meaningful latent factors. Therefore, the goal of our evaluation is to test whether our PAR model can perform similarly to HFT in terms of rating prediction and latent factor discovery, and on top of that, whether our PAR model can well separate product properties and aspects, which HFT cannot do. In the rest of this section, we present our evaluation as follows. We first compare PAR with HFT in terms of finding meaningful latent factors. We then evaluate how well PAR separates properties and aspects. Finally, we compare PAR with HFT for rating prediction. Note that when we compare PAR with HFT in the first and the third tasks, we do not expect PAR to outperform HFT but we want to make sure PAR performs comparably to HFT.

In all our experiments, we use the same number

[1] http://www.yelp.com

[2] https://www.kaggle.com/c/yelp-recsys-2013

	Product Properties		Aspects	
	Number	Avg. # Relevant Words	Count	Avg. # Relevant Words
SOFT	18	11.3	9	9.2
MP3	6	5.0	13	9.9
REST	13	10.4	5	7.8

Table 2: Summary of the Ground Truth Latent Factors.

of latent factors for PAR and HFT. For PAR, the number of latent factors is the number of properties plus the number of aspects, i.e. $P + A$. After some preliminary experiments, we set the total number of latent factors to 30 for both models. For PAR, based on observations with the preliminary experiments, we empirically set P to 10 and A to 20. Although these settings may not be optimal, by using the same number of latent factors for both models, no bias is introduced into the comparison.

For other hyperparameters, we empirically tune the parameters using a development set and use the optimal settings. For PAR, we set $\alpha = 2$, $\beta = 0.01$, $\sigma = 0.1$ and $\gamma = 1$. For HFT, we set $\mu = 10$ for MP3 and SOFT and $\mu = 0.1$ for REST. All results reported below are done under these settings.

4.3 Annotation of Ground Truth

The major goal of our evaluation is to see how well the PAR model can identify and separate product properties and aspects. However, in all three data sets we use, there is no ground truth and we are not aware of any data set with ground truth labels we can use for our task. Therefore, we have to annotate the data ourselves.

Instead of asking annotators to come up with product properties and aspects, which would require them to manually go through all reviews and summarize them, we opted to ask them to start from latent factors discovered by the two models. We randomly mixed the latent factors learned by PAR and HFT. The top 15 words of each latent factor were shown to two annotators, and each annotator independently performed the following three steps of annotations. In the first step, an annotator had to determine whether a latent factor was meaningful or not based on the 15 words. In the second step, for latent factors labeled as meaningful, an annotator had to decide whether it was a product property or an aspect. In the third step, an annotator had to pick relevant words from the given list of 15 words for each latent factor. After the three-step independent annotation, the two annotators compared and discussed their results to come to a consensus. During this discussion, duplicate latent factors were merged and word lists for each latent factor were finalized. The annotators were required to exclude general words such that no two latent factors share a common relevant word. In the end, the annotators produced a set of product properties and another set of aspects for each data set. For each latent factor, a list of highly relevant words was also produced. Table 2 shows the numbers of ground truth properties and aspects as labeled by the annotators and the average numbers of relevant words per latent factor of the three data sets.

4.4 Discovery of Meaningful Latent Factors

In the first set of experiments, we would like to compare PAR and HFT in terms of how well they can discover meaningful latent factors. Here latent factors include both product properties and aspects.

4.4.1 Results

We show three numbers for each data set and each method. The first is the number of "good" latent factors discovered by a method. Here a good latent factor is one that matches one of the ground truth latent factors. A learned latent factor matches a ground truth latent factor if the top-15 words of the learned latent factor cover at least 60% of the ground truth relevant words of the ground truth latent factor. We find the 60% threshold reasonable because most matching latent factors appear to be meaningful.

We use Precision and Recall as the evaluation metric. We would like to point out that the recall defined in this way is higher than the real recall value, because our ground truth latent factors all come from the discovered latent factors, but there may exist meaningful factors that are not discovered by either HFT or PAR at all. Nevertheless, we can still use this recall to compare PAR with HFT. The results are shown in Table 3. As we

	SOFT			MP3			REST		
	# Good LF	Prec	Rec	# Good LF	Prec	Rec	# Good LF	Prec	Rec
PAR	20	0.67	0.74	14	0.47	0.74	10	0.33	0.56
HFT	20	0.67	0.74	12	0.40	0.63	10	0.33	0.56

Table 3: Results for Identification of Meaningful Latent Factors

can see from the table, PAR and HFT performed similarly in terms of discovering meaningful latent factors. PAR performed slightly better than HFT on the MP3 data set. Overall, between one-third to two-thirds of the discovered latent factors are meaningful for both methods, and both methods can discover more than half of the ground truth latent factors.

4.5 Separation of Product Properties and Aspects

In this second set of experiments, we would like to evaluate how well PAR can separate product properties and aspects. In order to focus on this goal, we first disregard the discovered latent topics that are not considered good latent topics according to the criterion used in the previous experiment.

We then show the 2×2 confusion matrix between the labeled two types of latent factors and the predicted two types of latent factors by PAR for each data set. The results are in Table 4. As we can see, our model does a very good job in separating the two types of latent factors for MP3 and REST. For SOFT, our model mistakenly labeled 4 product properties as aspects. Although this result is not perfect, it still shows that our model can separate properties from aspects well in different domains.

We find that properties in the software domain are mostly functions and types of software such as games, antivirus software and so on. Aspects of software include software version, user interface, online service and others. In the MP3 data set, properties are mainly about MP3 brands such as Sony and iPod while aspects are about batteries, connections with computers and some others. Properties of the restaurant data set are all types of cuisines and aspects include ambiance and service.

4.6 Rating Prediction

Finally we compare our model with HFT for rating prediction in terms of root mean squared error. The results are shown in Table 5. We can see that PAR outperforms HFT in two real data sets (SOFT, MP3) and gets the same performance for the data set REST. This means separating properties and aspects in the model did not compromise rating prediction performance, which is important because otherwise the learned latent factors might not be the best ones explaining the ratings.

Prediction	Ground Truth					
	SOFT		MP3		REST	
	P	A	P	A	P	A
P	8	2	3	0	8	0
A	4	6	1	10	0	2

Table 4: Confusion Matrices of PAR for all Data Sets. *P stands for property and A stands for aspect.

	SOFT	REST	MP3
PAR	1.394	1.032	1.401
HFT	1.399	1.032	1.404

Table 5: Performance in Rating Prediction.

5 Conclusion and Future Work

We presented a joint model of product properties, aspects and numerical ratings for online product reviews. The major advantage of the proposed model is its ability to separate product properties, which are intrinsic to products, from aspects that are meant for comparing products in the same category. To achieve this goal, we combined probabilistic topic models with matrix factorization. We explicitly separated the latent factors into two groups and used both groups to generate both review texts and ratings. Our evaluation showed that compared with HFT our model could achieve similar or slightly better performance in terms of identifying meaningful latent factors and predicting ratings. More importantly, our model is able to separate product properties from aspects, which HFT and other existing models are not capable of.

References

Yang Bao, Hui Zhang, and Jie Zhang. 2014. TopicMF: Simultaneously exploiting ratings and reviews for

recommendation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 2–8.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 193–202.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 165–172.

Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 339–348.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37:9–27.

Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic matrix factorization. In *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*, pages 1257–1264, Vancouver, British Columbia, Canada. Curran Associates, Inc.

Christina Sauper, Aria Haghighi, and Regina Barzilay. 2011. Content models with attitude. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 350–358.

Ivan Titov and Ryan T. McDonald. 2008a. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 308–316.

Ivan Titov and Ryan T. McDonald. 2008b. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web*, pages 111–120.

Hanna M. Wallach. 2006. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 977–984.

Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 448–456.

Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011a. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 618–626.

Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011b. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 618–626.