2015 IEEE International Conference on Data Mining

Generative Models for Mining Latent Aspects and Their Ratings from Short Reviews

Huayu Li+, Rongcheng Lin+, Richang Hong* and Yong Ge+
+ *UNC Charlotte,* * *Hefei University of Technology*
{hli38,rlin4,yong.ge}@uncc.edu, hongrc@hfut.edu.cn

***Abstract*—A large number of online reviews have been accumulated on the Web, such as Amazon.com and Cnet.com. It is increasingly challenging to digest these reviews for both consumers and firms as the volume of reviews increases. A promising direction to ease such a burden is to automatically identify aspects of a product and reveal each individual's ratings on them from these reviews. The identified and rated aspects can help consumers understand the pros and cons of a product and make their purchase decisions, and help firms learn user feedbacks and improve their products and marketing strategy. While different methods have been introduced to tackle this problem in the past, few of them successfully model the intrinsic connection between aspect and aspect rating particularly in short reviews. To this end, in this paper, we first propose the Aspect Identification and Rating (AIR) model to model observed textual reviews and overall ratings in a generative way, where the sampled aspect rating influences the sampling of sentimental words on this aspect. Furthermore, we enhance AIR model to particularly address one unique characteristic of short reviews that aspects mentioned in reviews may be quite unbalanced, and develop another model namely AIRS. Within AIRS model, we allow an aspect to directly affect the sampling of a latent rating on this aspect in order to capture the mutual influence between aspect and aspect rating through the whole generative process. Finally, we examine our two models and compare them with other methods based on multiple real world data sets, including hotel reviews, beer reviews and app reviews. Experimental results clearly demonstrate the effectiveness and improvement of our models. Other potential applications driven by our results are also shown in the experiments.**

I. Introduction

With the rapid growth of the Internet, a large number of product reviews and ratings have been accumulated on the Web. For instance, Amazon.com, Cnet.com and Epinions.com are examples of the Web resources which contain such opinions contributed by worldwide consumers. These online reviews and ratings have become an increasingly important source of information that provides benefits both for the consumers and the firms that host markets. A consumer may not only pay attention to the (average) overall ratings of a product but also read the actual reviews when he is making purchase decision. The firm that owns a product also would like to learn customers' positive and negative feedbacks embedded in the online reviews as well. Both the availability of data and the practical needs have led to many studies on opinion summarization [1][2][3], sentiment analysis of online reviews [4][5], and aspect mining [6][7][8][9][10][11].

However, it is very challenging to digest the large number of online reviews and ratings for people. A promising direction to ease such a burden is to automatically identify multiple aspects of a product that customers have discussed about in reviews and reveal each individual's positive and negative ratings on those aspects. With the identified and rated aspects, customers could understand the pros and cons of a product in a more effective way when they are shopping online. A customer could even quickly locate the corresponding actual reviews which express positive or negative opinions on a particular aspect of product that is interesting to the customer. For instance, a potential camera buyer who cares the "stability" of camera very much may efficiently find out users' opinions on this aspect from the massive online reviews after we identify this aspect and users' opinions on it. Firms or product developers could also efficiently learn the important aspects of their products which are liked or disliked by their customers which will provide a good insight for them to market and improve their products. For instance, a camera company could better recommend a particular camera to different people who care the positively rated aspects of this product.

Although identifying and rating important aspects of products from online reviews has great benefit for both consumers and firms, it is a very challenging problem mainly due to the following three factors. First, there is no prior information about how many and what underlying aspects of a product are discussed in reviews. In particular, users may discuss about latent aspects of a product in implicit and diverse ways. Second, there is usually just one overall rating for each review at most Webs. The relationship between the overall rating and textual review could be quite dynamic among different people. Consequently, it is not easy to model both ratings and textual reviews together. Finally, different consumers may have different preference and emphasis on different aspects for one product. Consumers may or may not discuss all aspects of a product. Likewise, consumers may give overall ratings based on partial or all aspects of a product. All these factors together cause it very difficult to model both review and rating together for revealing aspects and each consumer's ratings on them.

In the literature, there are some works on aspect identification and (or) aspect rating. For example, [11][12][13]

1550-4786/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDM.2015.28

IEEE computer society

proposed LDA based models to mine the underlying aspects and estimate their corresponding aspect ratings based on the sentiment phrases. Also, [14] tried to enhance the coherence between the extracted topics and corresponding aspects, which basically threw light on the aspect identifications. And [8] combined LDA and a rating regression approach to automatically uncover the latent aspects and the ratings on each aspect with textual reviews and overall ratings. However, there are some limitations on these works. Many proposed models rely on some inputs which are expensive to obtain in many applications. For instance, [11][12][13] rely on pairs of aspect and sentiment; [14] requires the observed aspect ratings. Wang et.al. [8] presents a method to identify the aspects and predict aspect ratings based on textual reviews and overall ratings. However, it does not successfully model the intrinsic connection between aspect and aspect rating particularly in short reviews. Such intrinsic connection is that a higher or lower rating on one aspect often indicates more positive or negative words about the aspect because reviewers often form a rating for one aspect in their mind first and then choose positive or negative words to express their comments.

To address these limitations, we propose a unified framework which models both textual reviews and overall ratings in a generative process. In this framework, the generation of review and aspect rating influences each other through the overall generative process. Specifically, we first propose a new Aspect Identification and Rating model (AIR) for mining textual reviews and overall ratings, which aims at (1) identifying latent aspects (or topics) [1] of products, (2) predicting individual reviewer's ratings on each aspect, (3) capturing sentiments for each aspect. Within AIR model, we allow an aspect rating to influence the sampling of word distribution of the aspect for each review. Three Dirichlet distributions over words with neutral, positive and negative sentiments are used to characterize each topic. Moreover, as we observe that aspects mentioned in short reviews may be quite unbalanced, we further enhance AIR model to particularly handle this characteristic of short reviews and develop another model namely AIRS. Unlike AIR model, we sample aspect ratings from a Beta distribution based on the overall rating and the corresponding topic mixture weights in AIRS model. In other words, AIRS model captures the mutual influence between aspect and aspect rating through the whole generative process. Finally, experiments on multiple real word data sets including hotel review data, ratebeer review data and App review data show that our two models outperform the baseline methods on aspect identification and aspect rating prediction. We also demonstrate that that AIRS model performs better than AIR model on short reviews which have unbalanced aspects. In addition, we demonstrate the identified aspects and individual reviewer's ratings on them could be helpful with other applications, such as user behavior understanding and software/product improvement, with our app review data.

[1]The terms "topic" and "aspect" are used interchangeably in this paper.

Table I: A Summary of Notations

Symbol	Description
i	The review index in the corpus.
j	The word index in the review.
N_i	The number of words in the i-th review.
N	The number of reviews.
K	The number of latent topics.
R_i	The observed overall rating for the i-th review. We normalize R_i to (0,1) before training models.
w_{ij}	The j-th word in the i-th review.
z_{ij}	The topic assigned to the j-th word in the i-th review.
s_{ij}	The sentiment index assigned to the i-th word in the j-th review. 0, 1 and 2 means neutral, positive negative respectively.
Ω_{ik}	The predicted rating for the k-th latent topic in the i-th review.
θ_i	Topic distribution of the i-th review.
t_i	Neutral word ratio in the i-th review.
ϕ_k^0	Word distribution of the k-th topic given neutral sentiment.
ϕ_k^1	Word distribution of the k-th topic given positive sentiment.
ϕ_k^2	Word distribution of the k-th topic given negative sentiment.
α, β	Parameters of Dirichlet distribution.
γ	Parameters of Beta distribution.
λ	Tuning Parameter.

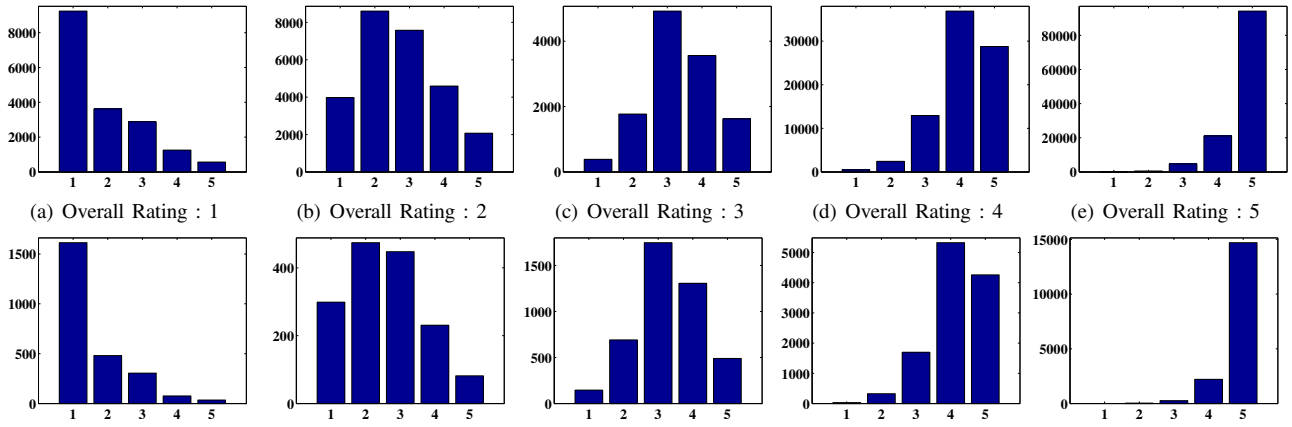
II. Methods

In this section, we first introduce the AIR model to extract aspects as well as positive and negative sentiments from reviews, and at the same time predict each reviewer's ratings on aspects. Furthermore, we extend AIR model to particularly address the challenge of unbalanced aspects in short reviews and develop the AIRS model. Some notations are shown in Table I.

A. The AIR Model

When users have a latent high (or low) rating on one aspect of a product, they are more likely to comment on the aspect with positive (or negative) words. For example, a review "**Good game, love it, so addicting.**" with a 5-star rating consists of more positive words like **good**, **love**, and **addicting**. But a review "**This is a stupid game.**" with a 1-star rating contains more negative words like **stupid**. In other words, for each review, the latent aspect rating implicitly influences the occurrence of negative or positive words which are used to comment on the aspect. Based on this assumption, we first develop our AIR model, where we capture the correlation between the aspect rating and the word distribution of aspect through the overall generative process.

Similar as conventional topic models, in AIR model we represent each review with a distribution over a set of latent aspects and denote each aspect as a distribution over a set of words. However, different from traditional topic models, the extraction of aspects (topics) and the sampling of words for each aspect are affected by the sampled latent aspect ratings which are dependent on the overall ratings given by reviewers. In other words, we argue that the probability of sampling a word for an aspect depends on different sentiments (i.e., latent aspect ratings). In this paper, we consider three types



(a) Overall Rating : 1 (b) Overall Rating : 2 (c) Overall Rating : 3 (d) Overall Rating : 4 (e) Overall Rating : 5

(f) Overall Rating : 1 (g) Overall Rating : 2 (h) Overall Rating : 3 (i) Overall Rating : 4 (j) Overall Rating : 5

Figure 1: X-axis denotes different values of aspect rating and Y-axis is the frequency of them. (a)∼(e) are the histograms with respect to different values of overall ratings. (f)∼(j) are the histograms of ratings on Service aspect with respect to different values of overall ratings.

of sentiments: neutral, positive and negative sentiments. Thus each topic is characterized by three different word distributions: ϕ_k^0, ϕ_k^1 and ϕ_k^2, which correspond to neutral, positive and negative sentiments respectively. As positive and negative words on an aspect are influenced more by the corresponding aspect rating. We first utilize a Beta distribution with prior γ to sample the ratio of neutral words for individual review. Then the sentiment orientation on an aspect is sampled from a Multinomial distribution, where the positive or negative sentiment is influenced by the rating on this aspect.

In addition, we show the histograms of aspect ratings with respect to different values of overall rating in Figure 1 with the TripAdvisor review data, the detail of which will be introduced in section III. For instance, we choose all reviews with overall rating as 1 and show the histogram of aspect ratings of these reviews in Figure 1. From these histograms, we obtain two observations: (1) These histograms of aspect rating are very close to the Beta distribution; (2) The value of aspect rating with maximum frequency is always equal to the value of overall rating, which indicates the mean of Beta distribution should be the overall rating. Motivated by these observations, we propose to leverage another Beta distribution with prior λR_i and $\lambda(1-R_i)$ to sample latent aspect ratings for each individual review based on its overall rating. The graphical model and general process for AIR are shown in Figure 2(b) and Table II, respectively.

Different algorithms have been proposed to estimate the parameters of generative processes such as Gibbs Sampling [15] and variational inference [16]. We choose to use Gibbs Sampling for inferring our model because it rapidly converges to the known ground-truth distribution [15]. Thus in this paper, Gibbs sampling method is used for obtaining parameter estimations. Six sets of unknown parameters: document distribution θ, neutral ratio t, predicted ratings Ω, words distribution ϕ, and two latent variables (i.e., topic z and sentiment index s) need to be estimated in the model. By Gibbs sampling, the transitions between successive states

Table II: The generative process for the AIR model.

1. For each review i,
   a. Draw the latent aspects $\theta_i \sim Dir(\alpha)$.
   b. Draw the neutral ratio $t_i \sim Beta(\gamma)$.
   c. For each aspect k, draw the predicted rating $\Omega_{ik} \sim Beta(\lambda R_i, \lambda(1-R_i))$.
2. For each aspect k,
   a. Draw the word distribution $\phi_k^0 \sim Dir(\beta)$ under neutral sentiment.
   b. Draw the word distribution $\phi_k^1 \sim Dir(\beta)$ under positive sentiment.
   c. Draw the word distribution $\phi_k^2 \sim Dir(\beta)$ under negative sentiment.
3. For each word w_{ij} in review d_i,
   a. Draw a topic $z_{ij} \sim Multinomial(\theta_i)$.
   b. Draw a sentiment index $s_{ij} \sim Multinomial(t_i, (1-t_i)\Omega_{iz_{ij}}, (1-t_i)(1-\Omega_{iz_{ij}}))$.
   c. Draw a word $w_{ij} \sim Multinomial(\phi_{z_{ij}}^{s_{ij}})$.

of the Markov chain results from repeatedly drawing latent variables z and s from their distribution conditioned on all other variables, integrating out θ, t, Ω and ϕ [17]. To apply the Gibbs sampling method, we first calculate the conditional distribution of z_{ij} and s_{ij}:

$$P(z_{ij}=k, s_{ij}=l \mid w_{ij}=v, \mathbf{w}_{-ij}, \mathbf{z}_{-ij}, \mathbf{s}_{-ij}) \propto \frac{C_{vkl}^{VKS}+\beta_v}{\sum_{v'} C_{v'kl}^{VKS}+\beta_{v'}} \times \frac{C_{ik}^{NK}+\alpha_k}{\sum_{k'} C_{ik'}^{NK}+\alpha_{k'}} \times \frac{C_{it}^{NB}+\gamma_t}{\sum_{t'} C_{it'}^{NB}+\gamma_{t'}} \times \left(\frac{C_{ikl}^{NKB}+\widetilde{\lambda}_l}{\sum_{l'} C_{ikl'}^{NKB}+\widetilde{\lambda}_{l'}}\right)^{I_l},$$

where $\widetilde{\lambda}_l = \begin{cases} \lambda R_i & \text{if } l=1 \\ \lambda(1-R_i) & \text{if } l=2 \end{cases}, t = \begin{cases} 1 & \text{if } l=0 \\ 2 & \text{if } l \neq 0, \end{cases}$ I_l is the indicator function that is equal to 1 if $l \neq 0$ and equal to 0 otherwise. $z_{ij}=k$ and $s_{ij}=l$ represent the assignment of the j-th word in the i-th review to topic k and sentiment index l respectively. $w_{ij}=v$ represents the mapping from the j-th word in i-th review to the v-th word in the vocabulary. $\mathbf{z}_{-ij}$ represents all topic assignments excluding the current instance. And $\mathbf{s}_{-ij}$ represents all sentiment assignments excluding the current instance. Furthermore, C_{vkl}^{VKS} is the number of times for word v assigned to topic k and sentiment l. C_{ik}^{NK} represents the number of times that topic k has occurred in the i-th review. C_{it}^{NB} represents the number of times neutral ($t=1$) or sentimental ($t=2$) words have occurred in the i-th review. C_{ikl}^{NKB} represents the number of times positive ($l=1$) or negative ($l=2$) words

assigned to topic k have occurred in the i-th review. In the burn-in process, C_{ik}^{NK}, C_{vkl}^{VKS}, C_{it}^{NB} and C_{ik1}^{NKB} are the counts that exclude current instance. Meanwhile, B equals to 2. After sampling each latent variable via Gibbs Sampling, we can estimate the parameters θ_{ik}, ϕ_{kv}^{l}, t_i, Ω_{ik} by:

$$\theta_{ik} = \frac{C_{ik}^{NK} + \alpha_k}{\sum_{k'} C_{ik'}^{NK} + \alpha_{k'}}, \phi_{kv}^{l} = \frac{C_{vkl}^{VKS} + \beta_v}{\sum_{v'} C_{v'kl}^{VKS} + \beta_{v'}},$$

$$t_i = \frac{C_{it}^{NB} + \gamma_1}{\sum_{t'} C_{it'}^{NB} + \gamma_{t'}}, \Omega_{ik} = \frac{C_{ik1}^{NKB} + \widetilde{\lambda}_1}{\sum_{l'} C_{ikl'}^{NKB} + \widetilde{\lambda}_{l'}},$$

where during the parameter estimation, C_{ik}^{NK}, C_{vkl}^{VKS}, C_{it}^{NB} and C_{ik1}^{NKB} are the counts including current instance, which is slightly different from previous sampling procedure.

As can be seen from the above inference, AIR model could (1) capture the correlation between the aspect rating and the word distribution of topic, and (2) learn latent aspects and aspect ratings with observed overall ratings and textual reviews.

B. The AIRS Model

In AIR model, we assume the aspect ratings are mainly influenced by the overall rating and thus draw the aspect ratings from a Beta distribution without the direct impact of aspect distribution of each review. However, in reality, many online reviews are very short as many reviewers often place emphasis on those aspects which they are concerned about. Therefore aspects mentioned in short reviews could be quite unbalanced. Thus the rating on one aspect that has been frequently commented in a review is naturally more correlated with the overall rating. Such unbalance in a review could be reflected directly in its topic distribution. Therefore, we enhance AIR model by introducing an explicit connection between the topic distribution of review and aspect ratings, and propose the AIRS model, where the aspect distribution of each review will explicitly influence the sampling of aspect ratings.

To further motivate our AIRS model, we explore our TripAdvisor review data to investigate the correlation between topic distribution and the impact of overall rating on aspect ratings. First we leverage the full set of keywords provided by [8] as prior to train LDA model with our data (more details will be introduced in Section III). For each topic (i.e., aspect) of a review, we get the corresponding aspect rating and check if its value matches the overall rating of this review. Then we compute the probability that the observed aspect rating matches the corresponding overall rating with respect to different values of topic probability in Figure 2(a). From the result, we can observe that when the topic probability increases, the rating on the aspect is much closer to the overall rating. In other words, if a reviewer talks more about a particular aspect, the rating on this aspect would be affected more by the overall rating.

Similar as AIR model, each review is represented by a mixture over K topics and a distribution over topics denoted by θ_i is sampled from a Dirichlet distribution for review i in AIRS model. Accordingly, θ_{ik} is the mixture weight of the k-th topic for review i. Different from AIR model, in AIRS model we sample aspect ratings for a review based on a Beta distribution which has the overall rating and the topic mixture weights of this review as prior. Specifically, based on the observed monotone increasing relationship between the topic mixture weight and the impact of overall rating on aspect rating as shown in Figure 2(a), we propose to draw the aspect rating on the k-th aspect for the i-th review from $Beta(\lambda\theta_{ik}R_i, \lambda\theta_{ik}(1-R_i))$. Consequently we develop another generative model, namely AIRS. Comparing with AIR mode, we strengthen the correlation between aspects and their ratings in an explicit way in AIRS model. Figure 2(c) shows the graphical model of AIRS. The generative process is similar as that of the AIR model except that for each aspect in a review we draw its rating Ω_{ik} from $Beta(\lambda\theta_{ik}R_i, \lambda\theta_{ik}(1-R_i))$.

Since θ is one part of the prior in Beta distribution, it leads to the no-close-form integration over multi-variables in either posterior distribution or expectation formula. Thus it is intractable to estimate the parameters of AIRS model. Neither Gibbs sampling nor Variational Inference could be used to estimate the posterior and distributions of interest. However, EM for MAP method is applicable for estimating θ_{ik}, ϕ_{kv}^{l}, t_i, Ω_{ik}, because MAP estimates these parameters directly after a set of random values are given.

Based on the general EM algorithm summarized by [18] and common MAP method, in the E-step, we use the current set of parameters $\{\theta, \phi, t, \Omega\}$ (denoted as Θ) to evaluate the posterior distribution of latent variables. Given the k-th topic and the l-th sentiment index for the j-th word of review i, the posterior probability is computed as:

$$P(z_{ij} = k, s_{ij} = l \mid w_{ij} = v, \Theta) = \frac{\phi_{kv}^{l}\theta_{ik}t_{il}\Omega_{ikl}}{\sum_{k,l}\phi_{kv}^{l}\theta_{ik}t_{il}\Omega_{ikl}}, \quad (1)$$

where variables t_{il} and Ω_{ikl} are defined as:

$$t_{il} = \begin{cases} t_i & \text{if } l = 0 \\ 1 - t_i & \text{if } l \neq 0 \end{cases}, \quad \Omega_{ikl} = \begin{cases} 1 & \text{if } l = 0 \\ \Omega_{ik} & \text{if } l = 1 \\ 1 - \Omega_{ik} & \text{if } l = 2. \end{cases}$$

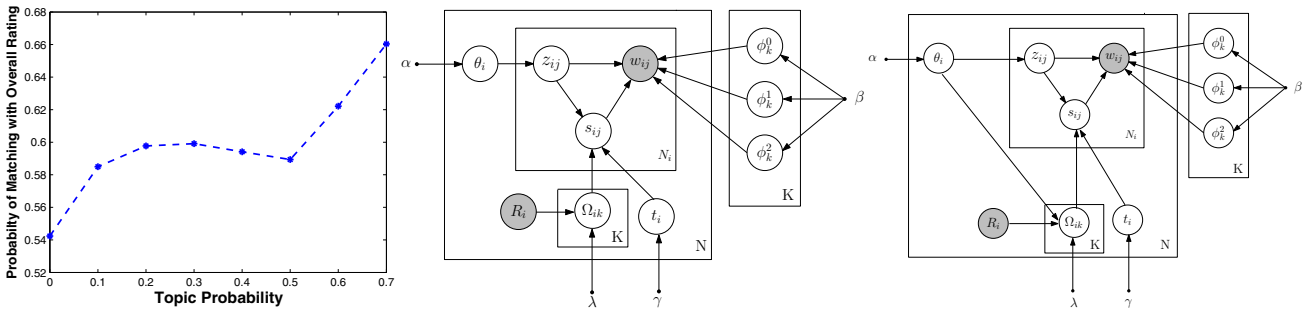
Let us denote $P(z_{ij} = k, s_{ij} = l \mid w_{ij} = v, \Theta)$ as P_{ijkl}. Then we further define two variables:

$$n_{ikl} = \sum_{j=1}^{N_i} P_{ijkl}, \quad n_{vkl}^{'} = \sum_{i=1}^{N}\sum_{j=1}^{N_i} I(w_{ij} = v)P_{ijkl},$$

where I(.) is the indicator function. Also we use the dot to denote the summation over an index, e.g., $n_{i.l} = \sum_{k=1}^{K} n_{ikl}$.

In M-Step, we obtain the new parameters by maximizing the lower bound $\mathcal{Q}$ given in Equation 2, and its full expression is given in Appendix.

$$\mathcal{Q} = \sum_{\mathbf{z},\mathbf{s}} P(\mathbf{z}, \mathbf{s} \mid \mathbf{w}, \Theta^{old}) log P(\mathbf{w}, \mathbf{z}, \mathbf{s} \mid \Theta) + log P(\Theta). \quad (2)$$



(a) (b) (c)

Figure 2: (a) The influence of overall ratings on aspect ratings for different topic probabilities. The topic probability is scaled by a approximating value, for example, if the topic probability falls into $[0.05, 0.15)$, then it is regarded as 0.1. (b) The Graphical Model for AIR. (c) The Graphical Model for AIRS.

As MAP has -1 offset [19], we simply remove the -1 offset during updating parameters. For example, if the value of α is originally set to 1.2, its value would be 0.2 after removing the -1 offset. Thus we can derive the updating equation of the neutral ratio t_i and the word distribution ϕ^l_{kv} by setting derivatives of the lower bound with respect to t_i and ϕ^l_{kv} to zero respectively. Specifically, we have their updating equations as follows:

$$t_i = \frac{n_{i.0} + \gamma_1}{N_i + \sum_{t'} \gamma_{t'}},\ \phi^l_{kv} = \frac{n'_{vkl} + \beta_v}{n'_{.kl} + \sum_{v'} \beta_{v'}}. \tag{3}$$

Due to θ is a part of prior for distribution Ω, it is intractable to get a close-form solution for θ. Therefore, we use the gradient-based optimization procedure to get the optimal solution for θ_{ik}, and θ_{ik} is given by:

$$\theta_{ik} = \arg\max_{\theta_{ik}} \{(n_{ik.} + \alpha_k) log\theta_{ik} + log\Gamma(\sum_{l=1}^{2} \theta_{ik}\widetilde{\lambda}_l) + \sum_{l=1}^{2} log\Gamma(\theta_{ik}\widetilde{\lambda}_l) + \sum_{l=1}^{2} \theta_{ik}\widetilde{\lambda}_l log\Omega_{ikl}\}, \tag{4}$$

$s.t. \forall k, 0 \leq \theta_{ik} \leq 1$ and $\sum_{k=1}^{K} \theta_{ik} = 1$. After obtaining the θ_i, the updating rule for Ω_{ik} shown in Equation 5 is derived by setting the derivative of lower bound with respect to Ω_{ik} to zero. Since there is Ω_{ik} in updating equation of θ as shown Equation 4, it is difficult to solve Ω and θ separately. Thus we use EM-style method to find the optimal solution of Ω and θ. Specifically we repeat updating θ and Ω until the convergence.

$$\Omega_{ik} = \frac{n_{ik1} + \theta_{ik}\widetilde{\lambda}_1}{\sum_{l=1}^{2} n_{ikl} + \sum_{l=1}^{2} \theta_{ik}\widetilde{\lambda}_l}. \tag{5}$$

To estimate the posterior probabilities and the parameters of interest, we first initialize all parameters randomly, and then utilize EM algorithm by executing E-Step and M-Step in each iteration until the log-likelihood given in the Appendix converges. E-Step and M-Step are described as:

1) (E-Step) For each word in each review, given the current parameters, evaluate the posterior distribution defined in Equation 1.
2) (M-Step) Given the posterior distribution, evaluate new parameters t, ϕ using Equations 3, and θ, Ω using EM-style method based on Equation 4 and 5.

III. Experimental Results

In this section, we evaluate the performance of AIR and AIRS models with multiple real-word data sets. Two kinds of short review data sets are leveraged to evaluate the performance of model fittings. As only hotel data sets have the observed aspect ratings (groundth), they are utilized to evaluate the accuracy of predicted ratings and the quality of mined topics. In addition, app data sets are applied for two different applications enabled by our models.

A. Experimental Setup

Datasets. Three types of online review data: the hotel review data collected from **TripAdvisor** [2], the beer review data collected from **RateBeer** [3], and the app review data crawled from **Applause** [4], are used in our experiments. Also we get two sets of app review data: one including reviews of several similar games which is denoted as **Applause 1** and another one including reviews of games *Temple Run* and *Temple Run II* which is denoted as **Applause 2**. Each review in the data sets has an overall rating. Specifically, the overall rating in RateBeer data ranges from 1 star to 20 stars, and the overall rating in other data sets ranges from 1 star to 5 stars. Especially, in the hotel data, there are also observed ratings on seven aspects of hotel, including *value*, *room*, *location*, *cleanliness*, *check in / front desk*, *service*, and *business service*. These aspect ratings also range from 1 star to 5 stars, which will be used as the ground-truth to evaluate the performance of our models on aspect rating predictions. We select those reviews containing all 7 aspect ratings and denote it as **Hotel**.

We do the following preprocessing on data sets: 1) converting all words into lower cases; 2) removing the punctuation; 3) removing stop words and the words that occur less than 10 times in the collection; 4) stemming each word to its root with *Porter Stemmer*[5]; 5) removing reviews

[2]http://www.tripadvisor.com/
[3]http://www.ratebeer.com/
[4]http://www.applause.com/
[5]http://tartarus.org/martin/PorterStemmer/

containing too less words; 6) normalizing each of overall ratings to the range of (0,1). We should note that we will recover the predicted aspect rating back to 5 star scale after model training.

In particular, in **TripAdvisor**, users are allowed to provide partial aspect ratings rather than all 7 aspect ratings. Aspects mentioned in those reviews with incomplete aspect ratings are very likely unbalanced. Therefore, we construct another hotel data set, denoted as **Incomplete Hotel**, where each review only contains partial aspect ratings for further evaluating the effectiveness of our models. Then we do the similar preprocessing. And the distribution of aspects covered in this data set is shown in Figure 3(a). Totally, we use five data sets in the experiments and their detailed statistics after being preprocessed are listed in Table III.

Table III: Statistics of Data Sets

Data Set	Review Number	Average Length
Hotel	39,586	113.04
Incomplete Hotel	9,339	31.50
RateBeer	94,963	23.40
Applause 1	77,465	14.91
Applause 2	15,606	17.55

Experiment Settings. In the experiments, we set α and β as $\frac{2.0}{K}$ and 0.01, respectively. γ is set as $(35, 13)$, and greedy algorithm is used to obtain the best parameter λ, where λ is finally set as 1.5 for AIR model and 15 for AIRS model. K is set as 7 for the evaluation of aspect ratings and qualities and K is 10 for the rest.

B. Evaluation Metrics

We will evaluate the performance of our models on aspect ratings and aspect identifications.

Aspect Rating Prediction Performance. Similar as [8], we adopt Mean Square Error (MSE) to evaluate the accuracy of aspect rating prediction. Aspect Pearson correlation (ρ_{aspect}), percentage of mis-ordered aspects inside reviews (Mis_{aspect}), and nDCG are leveraged to evaluate the ranking performance of the predicted aspect ratings. Let us denote N as the review number, and K_i as the observed aspect number in i-th review. MSE is then defined as $\frac{\sum_{i=1}^{N}\sum_{k=1}^{K_i}(R_{ik}-\hat{R}_{ik})^2}{\#total\ aspects}$, where R_{ik} and $\hat{R}_{ik}$ is the ground-truth and predict aspect rating respectively. ρ_{aspect} is given by $\frac{1}{N}\sum_{i=1}^{N}\rho(\mathbf{R}_i, \hat{\mathbf{R}}_i)$, where $\rho(\mathbf{R}_i, \hat{\mathbf{R}}_i)$ is the Pearson correlation between ground-truth and predict aspect rating vector. Mis_{aspect} is computed as $\sum_{i=1}^{N}\frac{\#discordant(i)}{\frac{1}{2}K_i(K_i-1)}$, where $\#discordant(i)$ is the discordant pairs between the predict and ground-truth aspect ratings in i-th review. DCG_i for i-th review is defined as $\sum_{k=1}^{K_i}\frac{2^{rel_k}-1}{log_2(1+k)}$, where rel_k is relevant score. Given the ideal DCG_i ($IDCG_i$), i.e. DCG_i of the ground-truth ratings, nDCG is defined as $\frac{1}{N}\sum_{i=1}^{N}\frac{DCG_i}{IDCG_i}$.

Aspect Identification Performance. Kullback Leibler (KL) Divergence is used to evaluate the quality of identified aspects, defined as $\frac{1}{K}\sum_x p(x)log\frac{p(x)}{q(x)}$, where $p(x)$ is the ground-truth word distribution, and $q(x)$ is the predicted word distribution.

C. Baseline Methods

To demonstrate the effectiveness of AIR and AIRS models, we adopt two baseline methods for comparisons. The first one is LARAM model proposed in [8] which assumes the overall rating is a weighted combination of aspect ratings for each individual review and then models the aspect ratings in a regression approach. The second one is LDA model [16] which has been widely used for uncovering the latent topics in a corpus. Specifically, LDA is used as a baseline to examine the model fitting of different methods. LARAM is used as a baseline to compare the aspect rating prediction performance for different approaches. And both LARAM and LDA are used as comparisons to evaluate the aspect identification performance.

D. Perplexity Comparison

In this subsection, we compute and compare the perplexity of a held-out test data set for different models with *RateBeer* data set and *Applause 1* data sets. Perplexity is a conventional metric for evaluating the performance of topic model. A lower perplexity indicates a better modeling of data. To compute the perplexity, we employ "Left-to-Right" evaluation [21], calculating perplexity in an incremental and "left-to-right" way for each review. Meanwhile, we hold out 10% of each review data set as the testing set to calculate the perplexity, and train models with the remaining 90%. Since AIRS model is implemented with MAP method in this paper, to make the comparison more fair we also implement both AIR and LDA with MAP methods similar as the section II-B. The perplexities versus the topic number for different methods are shown in Figure 3(b) and Figure 3(c).

Based on results, we can see the AIR model achieves the best performance among all models, indicating that the overall rating is helpful for model fitting. Compared to AIR and LDA implemented by MAP, AIRS is much superior, which illustrates that the introduced direct influence of topic distribution on the sampling of aspect ratings could lead to better performance on model fitting. In particular, when the topic number increases, AIRS outperforms them significantly. It happens mainly due to that with the increase of topic number, the characteristics of unbalanced aspects become more evident (i.e. the influence of aspect distribution on aspect ratings becomes more important). Therefore, the advantage of AIRS over LDA and AIR becomes more significant when the topic number is large. The performance of LDA with MAP is the worst, as it just simply models reviews with a set of topics without taking overall ratings into account. Totally, the methods implemented by MAP performs not good as the ones implemented by Gibbs sampling, because MAP method easily leads to over-fitting [19]. Therefore, We will use AIR model implemented by Gibbs

sampling method to mine latent aspects and predict their corresponding aspect ratings in the following experiments.

E. Performance of Aspect Rating

In this subsection, we evaluate the aspect rating performance for different models on both two hotel data sets. In order to ensure the discovered latent aspects by models are aligned with seven rated aspects of hotel, we utilize the full set of keywords provided by [8] as prior to guide the aspect modeling part. The detailed procedure could be found in [8]. We use the whole hotel data for both training and testing (i.e. no extra testing data is used here and we predict aspect ratings on the training reviews), and then quantitatively evaluate the performance of our models on aspect rating prediction. The performances of different methods with different validation metrics on both two hotel datasets are reported in the Table IV, where we only evaluate the predicted ratings on those aspects having groundtruth on Incomplete Hotel data.

From the results, we observe that the performances of AIR and AIRS are much better than LARAM. For example, our proposed methods have over 28% and 41% improvements in terms of MSE on *Hotel* and *Incomplete Hotel* data set respectively. Both our models and LARAM learn the aspect ratings with review texts and their overall ratings, but our models result in better performances because they appropriately model the correlation between aspect rating and overall rating. Specifically, our models sample aspect rating from a distribution related to the overall rating. And the aspect rating further influences the sampling of sentimental words. However, LARAM fails to capture such correlation in the modeling thus leads to bad performance on the prediction of aspect ratings. In addition, the AIRS model slightly outperforms the AIR model, particularly on *Incomplete Hotel* data set. For instance, the Mis_{aspect} of AIRS on *Incomplete Hotel* data is 0.130, while the result of AIRS is 0.144. It indicates that leveraging the topic mixture weight to sample aspect rating would benefit the prediction of aspect rating for short reviews where aspects mentioned may be quite unbalanced.

Table IV: Performances of Aspect Rating

	LARAM	AIR	AIRS
Hotel Data			
MSE	1.087	**0.782**	**0.782**
ρ_{aspect}	0.457	0.737	**0.738**
Mis_{aspect}	0.214	0.180	**0.178**
$nDCG$	**0.956**	**0.956**	**0.956**
Incomplete Hotel Data			
MSE	1.313	0.774	**0.765**
ρ_{aspect}	0.259	0.736	**0.737**
Mis_{aspect}	0.167	0.144	**0.130**
$nDCG$	0.966	0.967	**0.969**

F. Aspect Analysis

In this subsection, we first use KL Divergence metric to evaluate the quality of aspects extracted by our methods on both two hotel data sets, and then show some sentimental words learned by AIR model with *Applause 1* data set.

1) KL Divergence Performance: Similar to the procedure described in [8], in order to align latent aspects with the known seven aspects of hotel, we also use the same full set of keywords as prior to train LDA model with hotel review data. The learned latent topics by LDA are used as the ground-truth topics. Since we only have pre-defined 7 aspects, we set the number of topics as seven in our models. Then we calculate the KL divergence between the ground-truth topics and the topics learned by LDA without prior, LARAM, AIR, and AIRS. Particularly, to compare with the ground-truth topics, we sum the three topic distributions over words together for AIR and AIRS, because both models extract neutral, positive and negative words separately. The performance on both two hotel data sets are reported in Table V. Since we keep all sentiment words, such as isn't, aren't, and etc, the KL divergence results are a little larger than the results shown in [8].

From the results shown in Table V, we find that the topics learned by AIR are much closer to the ground-truth topics because the estimated aspect ratings are able to help extract sentiment, which leads to good sampling of words in each topic. LARAM performs the worst though it leverages the overall rating information to train the aspect model, which is consistent with the result in [8]. It illustrates that the sampling of aspect ratings based on their overall ratings and the sampling of positive/negative words upon aspect ratings would benefit aspect identification and rating prediction. AIRS performs a little worse than AIR because AIRS is implemented through MAP method which is prone to be over-fitting due to a single point estimation of parameters.

Table V: KL Divergence Performance

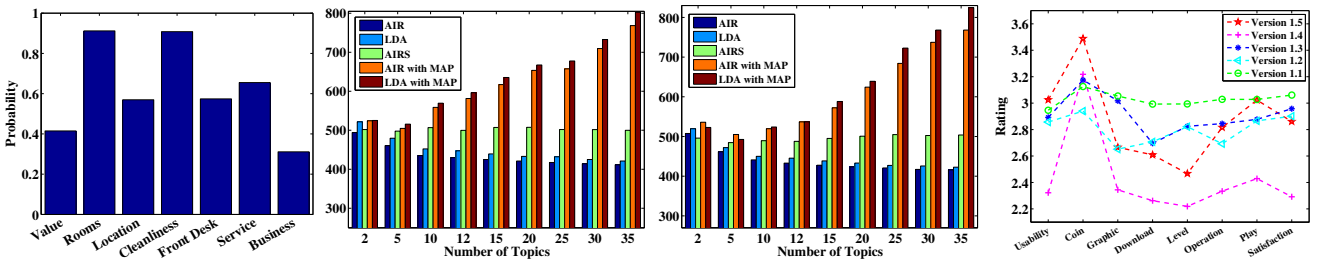
Hotel Data				
	LDA	LARAM	AIR	AIRS
7 topics	9.634	10.683	**8.735**	8.800
Incomplete Hotel Data				
7 topics	12.337	24.433	**8.912**	11.678

2) Aspect Sentiment Words: We train AIR model with app review data set *Applause 1* by using the bigram method which considers two words that co-occur in a sentence most frequently in the whole corpus as a term. The co-occurrence metric of two words is defined by a PMI similar method ($sPMI$) shown in the following:

$$sPMI = P(word_1 \wedge word_2) log_2 \frac{P(word_1 \wedge word_2)}{P(word_1)P(word_2)},$$

where $P(word)$ represents the frequency of $word$ occurring in the whole corpus, and $word_1 \wedge word_2$ represents $word_1$ and $word_2$ that co-occurs in one sentence. We consider two words co-occurring in one sentence as a term when their $sPMI$ is larger than a certain threshold. In the experiments, we set the threshold of $sPMI$ as e^{-4}.

Table VI shows top 10 positive and negative words in six different aspects when topic number is set as 10. We can



(a) Distribution of Aaspect Ratings (b) Perplexity on *Temple Run 2* (c) Perplexity on *Ratebeer* (d) Ratings on Different Versions

Figure 3: (a) The distribution of aspect ratings provided by users in Hotel-Incomplete data set. (b) $\sim$ (c) Perplexity comparisons on data set *Applause 1* and *Ratebeer*. X-axis is the number of topics and Y-axis is the perplexity. (d) Ratings on eight aspects for different versions of *Temple Run 2*.

Table VI: Top Sentiment Words Mined with AIR Model.

	Fighting	Usability	Sound & Graphic	Story	Platform	Control
Positive	god king	best	amazing graphic	great	amazing	awesome
	awesome	amazing	excellent	awesome	best ios	love
	best	amazing graphic	addicting	great graphic	best	cool
	beat	buy	beautiful	amazing	incredible	enjoy
	epic	awesome	perfect	play	fantastic	feel
	defeat	best play	control graphic	love	best play	hand
	avenge father	great graphic	incredible	wish more	cannot wait	action
	armor weapon	cannot wait	gorgeous	fun play	amazing graphic	addicting
	beat bloodline	addicting	fantastic	excellent	epic	sword fighting
	die	worth money	beautiful graphic	addicting	impressive	fight
Negative	fix	crash	sound effect	not	disappointed	dodge button
	save data	fix	no sound	boring	gameloft	fix
	bug	fix crash	lack	fight same	not	frustrating
	fix bug	cannot play	no	not worth	crap	control
	lost progress	crash start	sound	repetitive	no	problem
	lost	screen title	music sound	same	terrible	annoying
	play hour	crash play	frame rate	disappointed	horrible	bad
	save	fix bug	decent	epic citadel	sad	issue
	frustrating	crash open	missing	no	arena	unresponsive
	delete save	crash update	unfortunately	don't	garbage	dodge attack

see that even though some sentiment words are mixed up with neutral words, they are almost expressing sentimental information. In particular, positive (negative) words tend to express positive (negative) opinions. For example, for the aspect **Sound & Graphic**, the positive words **amazing graphic**, **excellent**, **addicting**, and **beautiful** reveal how much users like the sound and graphic of the app. Similarly the negative words **no sound**, **lack**, **no**, and **missing** explain the reason why users do not like this aspect. Also for aspect **Control**, the positive words, such as **awesome**, **love**, **cool** and **enjoy**, reflect that users like the **control** aspect of the app, and the negative words, such as **fix**, **frustrating** and **unresponsive**, on the other hand suggest users' negative opinions on this aspect.

G. Additional Applications

In this section, we introduce two additional applications enabled by AIR model with app review data set. One is utilizing AIR to analyze users' rating behaviors with the predicted aspect ratings. The other is to help developers to improve the quality of apps by analyzing aspect ratings on different versions of an app.

1) User Behaviors: A user's opinion could be decomposed into different aspect ratings which represent how much he likes the corresponding aspect of a product. AIR is employed to predict the aspect ratings on *Applause 1*. We summarize the predicted aspect ratings learned by AIR for two apps **Infinity Blade** and **Infinity Blade II** in Table VII. We can achieve two conclusions from the results. First, while users give the same overall rating on different apps, their opinions on each aspect sometimes are different. For example, user *Robert Rinehart* provides a **5** star rating for both two apps. However, the rating on aspect **Story** of **Infinity Blade** is only **2.277**, which can be reflected obviously from the review text, such as words **tedious**, **same monsters same places**; The rating on this aspect of **Infinity Blade II** is **5.054**. Thus it is possible to infer that *Robert Rinehart* does not like the **Story** aspect of **Infinity Blade**, but he is satisfied with this aspect of **Infinity Blade II**. Second, a user gives a low overall score for an app likely due to the his bad perception on one or several aspects. For instance, user *Wingbirdx*, provides different ratings on these two apps. Specifically, he gives an overall score **2** to **Infinity Blade II** probably because of his bad impression on aspect **Sound & Graphic**, on which the predicted rating is only **0.763**. It could be further justified by his review text, where words **lack of sounds** and **upsetting** express his opinion on this aspect directly.

2) Aspect Rating for Different Versions of App: We first train AIR model with *Applause 2* and learn the latent aspect ratings for different apps. After obtaining each user's aspect

Table VII: Rating Behavior Comparisons with Different Apps.

User Name	App	Overall Rating	Aspect Rating			Review Content
			Sound& Graphic	Story	Usability	
Robert Rinehart	Infinity Balde	5	4.978	**2.277**	5.093	Graphics are very good. Fun to play but it can get tedious. Same monsters same places. But overall as an app and not a Xbox game it's super.
	Infinity Balde II	5	5.01	**5.054**	5.336	Such a great game. Amazing graphics. Challenging but not impossible. It never gets old. Keep em coming.
Wingbirdx	Infinity Balde	5	**5.004**	5.178	5.182	Great graphics and fun game play can't wait for more.
	Infinity Balde II	2	**0.763**	1.992	1.836	The lack of sounds when swords clash and random bits of dialogue are very upsetting in an otherwise very polished game.
Someguy127	Infinity Baldex	5	5.058	5.051	**5.085**	The game is awesome! Infinty blade 2 better be as good! Simply the most impressive and breathtaking game in the app store!
	Infinity Balde II	4	3.909	4.043	**2.046**	Crashes at title screen still after entering a new rebirth. Still has bugs.

ratings, we average all users' aspect ratings for different versions of **Temple Run 2** and the result is shown in Figure 3(d). An interesting phenomenon is that ratings on most aspects tend to decrease as its version is updated. It is possible that more problems about **Temple Run 2** are identified by users as new versions are released. After obtaining users' aspect ratings for different versions, developers could know which aspect users like and which aspect users dislike, and they then could update their apps more effectively based on these useful feedbacks. For example, the rating on aspect **Coin** is higher than all other aspects in all versions, which suggests that users like coin collection the most when playing **Temple Run 2** and discuss it frequently in their reviews. Meanwhile, we can observe that when the version gets updated, the rating on aspect **Coin** also increases. It indicates that the design on **Coin** aspect is very received by users and developers should remain this design in the future version. In contrast, rating on aspect **Level** is the lowest among most of aspects. It reflects that users generally do not like upgrading or level settings of **Temple Run 2**. Thus analyzing latent ratings on each aspect for different versions could help developers to understand users' detailed opinions on their products. Based on this in-depth understanding, developers could effectively update their products to meet users' expectation and attract more users.

IV. Related Work

In this section, we mainly review the related work from three categories: summarization based aspect sentiment analysis, LDA based aspect sentiment techniques, and aspect rating prediction methods.

The first category focuses on summarization-based aspect sentiment extraction techniques [1, 2, 3]. For instance, [1, 2] provide an effective structured summary for product reviews. They first summarize product features (or aspects) extracted by natural language processing and data mining techniques, and then identify opinion sentences with corresponding opinion orientations. Positive (or negative) sentiment summarization is simply based on aggregating positive (or negative) review sentences. But different from them, our proposed models could not only capture aspects and sentiments, but also predict aspect ratings.

The second category is LDA based aspect sentiment techniques. LDA[16] is a generative probabilistic model for automatically uncovering latent topics from a large corpus, where each individual document is represented as a mixture over a set of latent topics, each of which is characterized by a distribution over words. Therefore, many extensions or variations are proposed to mine multiple aspects of a product from online reviews [8, 5, 22, 4]. For instance, [23] proposes MG-LDA model to study two types of topics: global topics capturing properties of reviewed objects, and local topics representing ratable aspects. As the coherence between topics and ratable aspects in MG-LDA model is not explicit, [14] incorporates the aspect ratings in the proposed MAS model in order to discover the coherence. Ester et.al. [11, 12, 13] develop several methods to extract aspects as well as predict their corresponding ratings from reviews. But these methods rely on opinion phrases (i.e., pairs of aspect and sentiment) as input which are often difficult to obtain. In addition, [24, 25, 26] propose to extract sentiment and topic from reviews. Specifically, [24] assumes each document has different topic-document distributions under each type of sentiment, i.e., positive, negative, and neutral sentiment. [26] as an extension of [24] models sentiments and topics in sentence level rather than document level. Furthermore, [25] reverses the sequence of sentiment and topic generation. However, compared to these models, we could extract aspects and sentiments more appropriately with the help of the overall ratings. Moreover, none of these works [24, 25, 26] output the aspect rating. In contrast, the focus of our models is on predicting the aspect rating.

The third category is about aspect rating prediction [6, 7, 8, 9, 10]. For instance, [7] first utilizes the bootstrapping-based method to extract aspects with some aspect keywords as seeds, then proposes a latent rating regression (LRR) approach to infer aspect ratings by assuming that the overall rating is the weighted combination of aspect ratings. The weight for each review reflects user's emphasis on each aspect. However, because it requires to specify some keywords in the aspect identification, this method is regarded as a supervised method and is not that practical.

To overcome this limitation, [8] furthermore proposes a LARAM model aiming at automatically learning aspects and at the same time predicting aspects. However, LARAM could not successfully model the intrinsic correlation (i.e., a higher or lower rating on one aspect often indicates more positive or negative words about the aspect) between aspect and aspect rating particularly in short reviews. In contrast, we model the inherent influence of aspect rating on word sampling in our models.

V. Conclusions

In this paper, we proposed two novel aspect identification and rating models which can model textual reviews and overall ratings at the same time to uncover latent aspects of products, each user's latent ratings on aspects and sentiments for each identified aspect. In our first model, i.e. AIR, we allow an aspect rating to influence the sampling of word distribution of the aspect for each review. Three Dirichlet distributions over words under neutral, positive and negative sentiments are used to characterize each topic. Moreover, based on the observations that many online reviews are short and the aspects mentioned in reviews are quite unbalanced, we proposed an enhanced model namely AIRS to better learn aspects and their ratings. In AIRS model, for each review, the sampling of an aspect rating will be directly influenced by the probability of this aspect. Thus, while AIR model only captures the one-way relationship between aspect and aspect rating, AIRS model could capture the mutual influence between aspect and aspect rating through the whole generative process. Gibbs sampling and EM for MAP were used to estimate parameters of AIR and AIRS model respectively. Finally, we conducted intensive experiments with three real-world review data sets to validate the performance of our models and demonstrate potential applications enabled by our methods.

VI. Acknowledgements

This research was supported in part by National Institutes of Health under Grant 1R21AA023975-01 and National Center for International Joint Research on E-Business Information Processing under Grant 2013B01035.

References

[1] M. Hu and B. Liu, "Mining and summarizing customer reviews." in *SIGKDD*, 2004a, pp. 168C–177.

[2] M. Hu and B. Liu, "Mining opinion features in customer reviews." in *AAAI*, 2004b.

[3] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews." in *HLT*, 2005, pp. 339–346.

[4] S. Brody and N. Elhadad, "An unsupervised aspect-sentiment model for online reviews." in *HLT*, 2010.

[5] B. Lu, M. Ott, C. Cardie, and B. K. Tsou, "Multi-aspect sentiment analysis with topic models." in *ICDMW*, 2011.

[6] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments." in *WWW*, 2009, pp. 131–140.

[7] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: A rating regression approach." in *KDD*, 2010.

[8] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis without aspect keyword supervision." in *KDD*, 2011.

[9] J. J. McAuley, J. Leskovec, and D. Jurafsky, "Learning attitudes and attributes from multi-aspect reviews." in *ICDM*, 2012.

[10] B. Snyder and R. Barzilay, "Multiple aspect ranking using the good grief algorithm." in *HLT-NAACL*, 2007, pp. 300–307.

[11] S. Moghaddam and M. Ester, "Ilda: interdependent lda model for learning latent aspects and their ratings from online product reviews." in *SIGIR*, 2011.

[12] S. Moghaddam and M. Ester, "On the design of lda models for aspect-based opinion mining." in *CIKM*, 2012.

[13] S. Moghaddam and M. Ester, "The flda model for aspect-based opinion mining: Addressing the cold start problem." in *WWW*, 2013.

[14] I. Titov and R. McDonald, "A joint model of text and aspect ratings for sentiment summarization." in *ACL*, 2008.

[15] T. L. Griffiths and M. Steyvers, "Finding scientific topics." *the National Academy of Sciences*, 2004.

[16] D. Blei, N. Andrew, and M. Jordan., "Latent dirichlet allocation," in *JMLR*, vol. 3, 2003, pp. 993–1022.

[17] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents." in *UAI*, 2004.

[18] C. M. Bishop, *Pattern Recognition And Machnie Learning.* Springer Science+Business Media, LLC, 2006.

[19] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh, "On smoothing and inference for topic models." in *UAI*, 2009.

[20] D. M. Blei and J. D. McAuliffe, "Supervised topic models," in *NIPS*, 2008.

[21] W. H. M, "Structured topic models for language." Ph.D. dissertation, University of Cambridge, 2008.

[22] W. X. Zhao, J. Jiang, H. Yan, and X. Li, "Jointly modeling aspects and opinions with a maxent-lda hybrid." in *EMNLP*, 2010.

[23] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models." in *WWW*, 2008.

[24] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis." in *CIKM*, 2009, pp. 375–384.

[25] C. Lin, Y. He, and R. Everson, "Weakly supervised joint sentiment-topic detection from text." in *TKDE*, 2012.

[26] Y. Jo and A. H. O. KAIST, "Aspect and sentiment unification model for online review analysis." in *WSDM*, 2011.

Appendix

Let us denote $P^{old}_{ijkl} = P(z_{ij} = k, s_{ij} = l \mid w_{ij}, \Theta^{old})$, then the full expression of lower bound defined in Section II-B is:

$$\begin{aligned}
&\mathcal{Q}(\theta, \phi, \Omega, t, \Theta^{old}) \\
&= \sum_{\mathbf{z},\mathbf{s}} P(\mathbf{z}, \mathbf{s} \mid \mathbf{w}, \Theta^{old}) log P(\mathbf{w}, \mathbf{z}, \mathbf{s} \mid \Theta) + log P(\Theta) \\
&= \sum_{i,j} \sum_{z_{ij}, s_{ij}} P(z_{ij}, s_{ij} \mid w_{ij}, \Theta^{old}) log P(w_{ij}, z_{ij}, s_{ij} \mid \Theta) + log P(\Theta) \\
&= \sum_{i,j}^{N,N_i} \sum_{k=1}^{K} \sum_{l=0}^{2} P^{old}_{ijkl} log(\phi^{l}_{k,w_{ij}} \theta_{ik} t_{il} \Omega_{ikl}) + \sum_{v=1}^{V} \sum_{k=1}^{K} \sum_{l=0}^{2} \beta_v log \phi^{l}_{kv} \\
&+ \sum_{i=1}^{N} \{\gamma_1 log t_i + \gamma_2 log(1 - t_i)\} + \sum_{i=1}^{N} \sum_{k=1}^{K} \{\alpha_k log \theta_{ik} \\
&+ log\Gamma(\sum_{l=1}^{2} \theta_{ik} \widetilde{\lambda}_l) - \sum_{l=1}^{2} log\Gamma(\theta_{ik} \widetilde{\lambda}_l) + \sum_{l=1}^{2} \theta_{ik} \widetilde{\lambda}_l log \Omega_{ikl}\}
\end{aligned}$$

$s.t. \forall k, 0 \leq \theta_{ik} \leq 1$ and $\sum_{k=1}^{K} \theta_{ik} = 1$. In addition, the log-likelihood is:

$$\begin{aligned}
log P(W|\Theta) &= \sum_{i,j}^{N,N_i} \sum_{k=1}^{K} \sum_{l=0}^{2} P(w_{ij} = v, z_{ij} = k, s_{ij} = l \mid \widetilde{\phi}, \widetilde{\theta}, \widetilde{t}, \widetilde{\Omega}) \\
&= \sum_{i,j}^{N,N_i} \sum_{k=1}^{K} \sum_{l=0}^{2} \widetilde{\phi}^{l}_{kv} \widetilde{\theta}_{ik} \widetilde{t}_{il} \widetilde{\Omega}_{ikl}
\end{aligned}$$

where $\widetilde{\phi}$, $\widetilde{\theta}$, $\widetilde{t}$, and $\widetilde{\Omega}$ are the estimated parameters of ϕ, θ, t, and Ω respectively.