Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach

Hongning Wang, Yue Lu, Chengxiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana IL, 61801 USA
{wang296, yuelu2, czhai}@uiuc.edu

ABSTRACT

In this paper, we define and study a new opinionated text data analysis problem called Latent Aspect Rating Analysis (LARA), which aims at analyzing opinions expressed about an entity in an online review at the level of topical aspects to discover each individual reviewer's latent opinion on each aspect as well as the relative emphasis on different aspects when forming the overall judgment of the entity. We propose a novel probabilistic rating regression model to solve this new text mining problem in a general way. Empirical experiments on a hotel review data set show that the proposed latent rating regression model can effectively solve the problem of LARA, and that the detailed analysis of opinions at the level of topical aspects enabled by the proposed model can support a wide range of application tasks, such as aspect opinion summarization, entity ranking based on aspect ratings, and analysis of reviewers rating behavior.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Text Mining

General Terms

Algorithms, Experimentation

Keywords

Opinion and sentiment analysis, review mining, latent rating analysis

1. INTRODUCTION

With the emergence and advancement of Web 2.0, more and more people can freely express opinions on all kinds of entities such as products and services. These reviews are useful to other users for making informed decisions and to merchants for improving their service. However, the volume of reviews grows so rapidly that it is becoming increasingly difficult for users to wade through numerous reviews to find the needed information. Much work has been done to alleviate this problem including extracting information from reviews [18, 16, 26], summarizing users' opinions, categorizing reviews according to opinion polarities [20, 6, 7], and extracting comparative sentences from reviews [12, 13]. Nevertheless, with the current techniques, it is still hard for users to easily digest and exploit the large number of reviews due to inadequate support for understanding each individual reviewer's opinions at the fine-grained level of topical aspects.

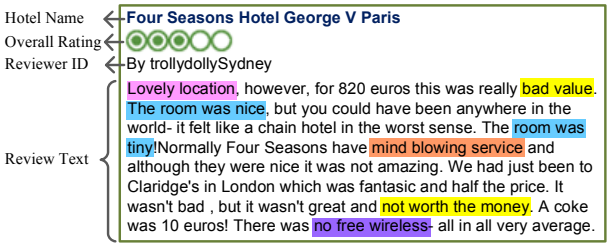

Figure 1: A Sample Hotel Review

Consider a typical hotel review shown in Figure 1. This review discusses multiple aspects of the hotel, such as price, room condition, and service, but the reviewer only gives an overall rating for the hotel; without an explicit rating on each aspect, a user would not be able to easily know the reviewer's opinion on each aspect. Going beyond the overall rating to know the opinions of a reviewer on different aspects is important because different reviewers may give a hotel the same overall rating for very different reasons. For example, one reviewer may have liked the location, but another may have enjoyed the room. In order to help users tell this difference, it is necessary to understand a reviewer's rating on each of the major rating aspects (i.e., rating factors) of a hotel. Furthermore, even if we can reveal the rating on an aspect such as "price", it may still be insufficient because "cheap" may mean different price ranges for different reviewers. Even the same reviewer may use a different standard to define "cheap" depending on how critical other factors (e.g. location) are; intuitively, when a reviewer cares more about the location, the reviewer would tend to be more willing to tolerate a higher price. To understand such subtle differences, it is necessary to further reveal the relative importance weight that a reviewer placed on each aspect when assigning the overall rating.

To achieve such deeper and more detailed understanding of a review, we propose to study a novel text mining problem called Latent Aspect Rating Analysis (LARA). Given a

KDD'10, July 25–28, 2010, Washington, DC, USA.

set of reviews with overall ratings, LARA aims at analyzing opinions expressed in each review at the level of topical aspects to discover each individual reviewer's latent rating on each aspect as well as the relative importance weight on different aspects when forming the overall judgment.

Revealing the latent aspect ratings and aspect weights in each individual review would enable a wide range of application tasks. For example, the revealed latent ratings on different aspects can immediately support aspect-base opinion summarization; aspect weights are directly useful for analyzing reviewers' rating behaviors; and the combination of latent ratings and aspect weights can support personalized aspect-level ranking of entities by using only those reviews from the reviewers with similar aspect weights to those preferred by an individual user.

While existing work on opinion summarization has addressed the LARA problem to certain extent, no previous work has attempted to infer the latent aspect rating at the level of each individual review, nor has it attempted to infer the weights a reviewer placed on different aspects. (See Section 2 for a more detailed review of all the related work.)

To solve this new mining problem, we propose a two-stage approach based on a novel latent rating regression model. In the first stage, we employ a bootstrapping-based algorithm to identify the major aspects (guided by a few seed words describing the aspects) and segment reviews. In the second stage, we propose a generative Latent Rating Regression (LRR) model which aims at inferring aspect ratings and weights for each individual review based only on the review content and the associated overall rating. More specifically, the basic idea of LRR is to assume that the overall rating is "generated" based on a weighted combination of the latent ratings over all the aspects, where the weights are to model the relative emphasis that the reviewer has placed on each aspect when giving the overall rating. We further assume that latent rating of each aspect depends on the content in the segment of a review discussing the corresponding aspect through a regression model. In other words, we may also view that the latent rating on each aspect as being "generated" by another weighted sum of word features where the weights indicate the corresponding sentimental polarities. Since we do not observe the ratings on different aspects, the response variable of this regression model (i.e., aspect rating) is latent.

We evaluate the proposed LRR model on a hotel data set crawled from TripAdviser (`www.tripadvisor.com`). Experiment results show that the proposed LRR model can effectively decompose the overall rating of a given review into ratings on different aspects and reveal the relative weights placed on those aspects by the reviewer. We also show that the results obtained from the LRR model can support several application tasks, including aspect opinion summarization, personalized entity ranking, and rating behavior analysis of reviewers.

2. RELATED WORK

To the best of our knowledge, no previous work has studied the proposed LARA problem, but there are several lines of related work.

Analysis of the overall sentiment of review text has been extensively studied. Related research started from a definition of binary classification of a given piece of text into the positive or negative class [6, 7, 20, 5, 14]. Later, the definition is generalized to a multi-point rating scale [19, 9]. Many approaches have been proposed to solve the problem, including supervised, un-supervised, and semi-supervised approaches, but they all attempt to predict an *overall* sentiment class or rating of a review, which is not so informative as revealing aspect ratings as we attempt to do.

Since an online review usually contains multiple opinions on multiple aspects, some recent work has started to predict the aspect-level ratings instead of one overall rating. For example, Snyder et al. [23] show that modeling the dependencies among aspects using good grief algorithm can improve the prediction of aspect ratings. In [24], Titov et al. propose to extract aspects and predict the corresponding ratings simultaneously: they use topics to describe aspects and incorporate a regression model fed by the ground-truth ratings. However, they have assumed that the aspect ratings are *explicitly* provided in the training data. In contrast, we assume the aspect ratings are *latent*, which is a more general and more realistic scenario.

Summarization is a generally useful technique to combat information overload. A recent human evaluation [15] indicates that sentiment informed summaries are strongly preferred over non-sentiment baselines, suggesting the usefulness of modeling sentiment and aspects when summarizing opinions. However, existing works on aspect-based summarization [10, 21, 18, 26] only aimed at aggregating all the reviews and representing major opinions on different aspects for a given topic. While aggregated opinions can present a general picture of a topic, the details in each review are lost; furthermore, the differences among reviews/reviewers are not considered, thus the aggregated sentiment is based on reviewers with different tastes. Recent work by Lu et al. [17] is the closest to ours, but their goal is still to generate an *aggregated* summary with aspect ratings inferred from overall ratings. Most importantly, none of the previous work considers the reviewer's emphasis on different aspects, i.e. aspect weight. Our work aims at inferring both the aspect ratings and aspect weights at the level of individual reviews; the result can be useful for multiple tasks, including opinion-based entity ranking, analysis of user rating behavior in addition to "rated aspect summarization".

3. PROBLEM DEFINITION

In this section, we formally define the problem of Latent Aspect Rating Analysis (LARA).

As a computational problem, LARA assumes that the input is a set of reviews of some interesting entity (e.g., hotel), where each review has an overall rating. Such a format of reviews is quite common in most of the merchants web site, e.g. Amazon (`www.amazon.com`) and Epinions (`www.epinions.com`), and the number of such reviews is growing constantly.

Formally, let $D = \{d_1, d_2, \ldots, d_{|D|}\}$ be a set of review text documents for an interesting entity or topic, and each review document $d \in D$ is associated with an overall rating r_d. We also assume that there are n unique words in the vocabulary $V = \{w_1, w_2, \ldots, w_n\}$.

Definition (Overall Rating) An overall rating r_d of a review document d is a numerical rating indicating different levels of overall opinion of d, i.e. $r_d \in [r_{min}, r_{max}]$, where r_{min} and r_{max} are the minimum and maximum ratings respectively.

We further assume that we are given k aspects, which are rating factors that potentially affect the overall rating of the given topic. For example, for hotel reviews, possible aspects may include "price" and "location." An aspect is specified through a few keywords, and provides a basis for latent aspect rating analysis.

Definition (Aspect) An aspect A_i is a (typically very small) set of words that characterize a rating factor in the reviews. For example, words such as "price", "value", and "worth" can characterize the price aspect of a hotel. We denote an aspect by $A_i = \{w|w \in V, A(w) = i\}$, where $A(.)$ is a mapping function from a word to an aspect label.

Definition (Aspect Ratings) Aspect rating $\mathbf{s}_d$ is a k dimensional vector, where the i-th dimension is a numerical measure, indicating the degree of satisfaction demonstrated in the review d toward the aspect A_i, and $\mathbf{s}_{di} \in [r_{min}, r_{max}]$. A higher rating means a more positive sentiment towards the corresponding aspect.

Definition (Aspect Weights) Aspect weight α_d is a k dimensional vector, where the i-th dimension is a numerical measure, indicating the degree of emphasis placed by the reviewer of review d on aspect A_i, where we require $\alpha_{di} \in [0, 1]$ and $\sum_{i=1}^{k} \alpha_{di} = 1$ to make the weights easier to interpret and comparable across different reviews. A higher weight means more emphasis is put on the corresponding aspect.

Definition (Latent Aspect Rating Analysis (LARA)) Given a review collection D about a topic T where each review document d is associated with an overall rating r_d, and k aspects $\{A_1, A_2, \ldots, A_k\}$ to be analyzed, the problem of Latent Aspect Rating Analysis (LARA) is to discover each individual review's rating $\mathbf{s}_{di}$ on each of the k aspects as well as the relative emphasis α_{di} the reviewer has placed on each aspect.

Informally, LARA aims at discovering the latent aspect ratings and aspect weights in each individual review d based on all the review contents and the associated overall ratings. While the most general setup of the LARA problem would consist of also discovering the possibly unknown aspects in addition to discovering the latent ratings/weights on different aspects, in this paper, we assume that we are given a few keywords describing each aspect. This assumption is realistic as for any given entity type, it is feasible to manually specify the major aspects in this way; besides, such a setup also gives a user control over what aspects to be analyzed.

4. METHODS

A major challenge in solving the problem of LARA is that we do not have detailed supervision about the latent rating on each aspect even though we are given a few keywords describing the aspects. Another challenge is that it is unclear how we can discover the relative weight placed by a reviewer on each aspect. To solve these challenges, we propose a novel Latent Rating Regression (LRR) model to tie both latent ratings and latent weights with the contents of a review on the one hand and the overall rating of the review on the other. Specifically, we assume that the reviewer generates the overall rating of a review based on a weighted combination of his/her ratings on all aspects, and the rating on each aspect is generated based on another weighted combination of the words in the review that discusses the corresponding aspect. After fitting such a two-fold regression model to all the review data, we would be able to obtain the latent aspect ratings and weights, thus solving the problem of LARA.

Since the LRR model assumes that we know which words are discussing which aspects in a review, we first perform aspect segmentation in a review document based on the given keywords describing aspects to obtain text segment(s) for each aspect. Thus, our overall approach consists of two stages, which we will further discuss in detail.

4.1 Aspect Segmentation

The goal of this first step is to map the sentences in a review into subsets corresponding to each aspect. Since we assume that only a few keywords are specified to describe each aspect, we design a boot-strapping algorithm to obtain more related words for each aspect.

Algorithm: Aspect Segmentation Algorithm

Input: A collection of reviews $\{d_1, d_2, \ldots, d_{|D|})\}$, set of aspect keywords $\{T_1, T_2, \ldots, T_k\}$, vocabulary V, selection threshold p and iteration step limit I.
Output: Reviews split into sentences with aspect assignments.
Step 0: Split all reviews into sentences, $X = \{x_1, x_2, \ldots, x_M\}$;
Step 1: Match the aspect keywords in each sentence of X and record the matching hits for each aspect i in $Count(i)$;
Step 2: Assign the sentence an aspect label by $a_i = argmax_i\ Count(i)$. If there is a tie, assign the sentence with multiple aspects.
Step 3: Calculate χ^2 measure of each word (in V);
Step 4: Rank the words under each aspect with respect to their χ^2 value and join the top p words for each aspect into their corresponding aspect keyword list T_i;
Step 5: If the aspect keyword list is unchanged or iteration exceeds I, go to ***Step 6***, else go to ***Step 1***;
Step 6: Output the annotated sentences with aspect assignments.

Figure 2: Boot-strapping method for aspect segmentation.

Specifically, the basic workflow of the proposed *Aspect Segmentation Algorithm* is as follows: given the seed words for each aspect and all the review text as input, we assign each sentence to the aspect that shares the maximum term overlapping with this sentence; based on this initial aspect annotation, we calculate the dependencies between aspects and words by Chi-Square (χ^2) statistic [25], and include the words with high dependencies into the corresponding aspect keyword list. These steps are repeated until the aspect keyword list is unchanged or the number of iterations exceeds the limit. The full description of the algorithm is in Figure 2. The χ^2 statistic to compute the dependencies between a term w and aspect A_i is defined as follows:

$$\chi^2(w, A_i) = \frac{C \times (C_1C_4 - C_2C_3)^2}{(C_1 + C_3) \times (C_2 + C_4) \times (C_1 + C_2) \times (C_3 + C_4)}$$

where C_1 is the number of times w occurs in sentences belonging to aspect A_i, C_2 is the number of times w occurs in sentences not belonging to A_i, C_3 is the number of sentences of aspect A_i that do not contain w, C_4 is the number of sentences that neither belong to aspect A_i, nor contain word w, and C is the total number of word occurrences.

After aspect segmentation, we would get k partitions of each review d, and represent them as a $k \times n$ feature matrix $\mathbf{W}_d$, where $\mathbf{W}_{dij}$ is the frequency of word w_j in the text assigned to aspect A_i of d normalized by the total counts of words in the text of that aspect.

4.2 Latent Rating Regression Model (LRR)

In the second stage, based on the aspect segmentation results in each review, we apply a novel Latent Rating Regression (LRR) model to analyze both aspect ratings $\mathbf{s}_d$ and aspect weights α_d.

4.2.1 The Generation Assumption

Our assumption of reviewer's rating behavior is as follows: to generate an opinionated review, the reviewer first decides the aspects she wants to comment on; and then for each aspect, the reviewer carefully chooses the words to express her opinions. The reviewer then forms a rating on each aspect based on the sentiments of words she used to discuss that aspect. Finally the reviewer assigns an overall rating depending on a weighted sum of all the aspect ratings, where the weights reflect the relative emphasis she has placed on each aspect.

4.2.2 The LRR Model

The LRR model is a regression model that formally captures the generation process discussed above. Recall that after aspect segmentation, for each review d, we have a word frequency matrix $\mathbf{W}_d$ which gives normalized frequency of words in each aspect. The LRR model treats $\mathbf{W}_d$ as independent variables (i.e., features of review d) and the overall rating r of the review as the response variable (i.e., variable to predict). In order to model the latent ratings on different aspects and the latent weights in the aspects, the LRR model further assumes that the overall rating is not directly determined by the word frequency features, but rather, based on a set of latent ratings on different aspects which are more directly determined by the word frequency features.

Formally, as we have defined in Section 3, $\mathbf{s}_d$ and α_d are review-level k-dimensional aspect weight vector and aspect rating vector, respectively. The reviewer for d would be assumed to first generate an aspect rating for each A_i as a linear combination of $\mathbf{W}_{di}$ and β_i, i.e.

$$\mathbf{s}_i \quad \sum_{j=1}^{n} \beta_{ij} \mathbf{W}_{dij} \tag{1}$$

where $\beta_i \in \Re$ indicates the word sentiment polarities on aspect A_i.

Then, the reviewer would generate the overall rating based on the weighted sum of α_d and $\mathbf{s}_d$, i.e. $\alpha_d^T \mathbf{s}_d = \sum_{i=1}^{k} \alpha_{di} \mathbf{s}_{di}$. Specifically, the overall rating is assumed to be a sample drawn from a Gaussian distribution with mean $\alpha_d^T \mathbf{s}_d$ and variance δ^2, which indicates the uncertainty of the overall rating predictions. Thus putting all together, we have

$$r_d \sim N(\sum_{i=1}^{k} \alpha_{di} \sum_{j=1}^{n} \beta_{ij} \mathbf{W}_{dij}, \delta^2) \tag{2}$$

Intuitively, the key idea here is to bridge the gap between the observed overall rating and the detailed text descriptions through introducing the latent aspect weight α_d and term sentiment weight β, which enable us to model the overall rating based on ratings of specific aspects.

Looking further into the rating behaviors, we find that reviewers' emphasis on different aspects can be complicated: 1) different reviewers might have different preferences for the aspects, e.g. business travelers may emphasize on internet service while honeymoon couples may pay more attention to rooms; 2) aspects are not independent, especially when the aspects have overlaps, e.g. an emphasis on cleanliness would indicate a preference to room too. In order to take the diversity of reviewer's preference into consideration, we further treat the aspect weight α_d in each review d as a set of random variables drawn from an underline prior distribution for the whole corpus. Furthermore, to capture the dependencies among different aspects, we employ a multivariate Gaussian distribution as the prior for aspect weights, i.e.

$$\alpha_d \sim N(\mu, \mathbf{\Sigma}) \tag{3}$$

where μ and $\mathbf{\Sigma}$ are the mean and variance parameters.

Combining Eq (2) and (3), we get a Bayesian regression problem. The probability of observed overall rating in a given review in our LRR model is given by:

$$\begin{aligned} P(r|d) &= P(r_d|\mu, \mathbf{\Sigma}, \delta^2, \beta, \mathbf{W}_d) \\ &= \int p(\alpha_d|\mu, \mathbf{\Sigma}) p(r_d|\sum_{i=1}^{k} \alpha_{di} \sum_{j=1}^{n} \beta_{dij} \mathbf{W}_{dij}, \delta^2) \mathrm{d}\alpha_d \end{aligned} \tag{4}$$

where r_d and $\mathbf{W}_d$ are the observed data in review d, $\mathbf{\Theta} = (\mu, \mathbf{\Sigma}, \delta^2, \beta)$ are the set of *corpus-level* model parameters, and α_d is the latent aspect weight for review d. Note that we assume that δ^2 and β do not depend on individual reviewers, and are thus also *corpus-level* model parameters. A graphical model illustration of LRR model is given in Figure 3.

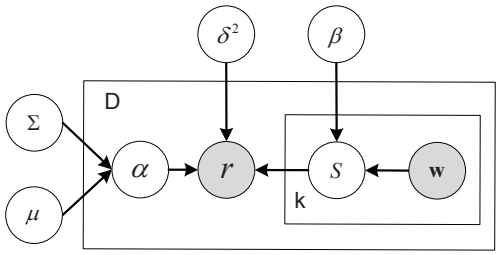

Figure 3: Graphical Representation of LRR. The outer box represents reviews, while the inner box represents the composition of latent aspect ratings and word descriptions within a review.

Suppose we are given the LRR model parameters $\mathbf{\Theta} = (\mu, \mathbf{\Sigma}, \delta^2, \beta)$, we can apply the model to get the aspect ratings and weights in each review as follows: (1) the latent aspect rating $\mathbf{s}_d$ in a particular review d could be calculated by Eq (1); (2) we appeal to the maximum a posteriori (MAP) estimation method to retrieve the most probable value of α_d in the given review. The object function of MAP estimation

for review d is defined as:

$$\mathcal{L}(d) = \log p(\alpha_d|\mu, \mathbf{\Sigma})p(r_d|\sum_{i=1}^{k}\alpha_{di}\sum_{j=1}^{n}\beta_{dij}\mathbf{W}_{dij}, \delta^2) \quad (5)$$

We expand this object function and associate all the terms with respect to α_d in each review (denote as $\mathcal{L}(\alpha_d)$) as follows:

$$\begin{aligned}\hat{\alpha}_d &= \arg\max \mathcal{L}(\alpha_d) \quad (6)\\ &= \arg\max\left[-\frac{(r-\alpha_d^T\mathbf{s_d})^2}{2\delta^2} - \frac{1}{2}(\alpha_d-\mu)^T\mathbf{\Sigma}^{-1}(\alpha_d-\mu)\right]\end{aligned}$$

subject to

$$\sum_{i=1}^{k}\alpha_{di} = 1$$
$$0 \le \alpha_{di} \le 1 \text{ for } i = 1, 2, \ldots, k$$

To address above constraint non-linear optimization problem, we apply the conjugate-gradient-interior-point method with the following formula for the derivatives with respect to α_d:

$$\frac{\partial\mathcal{L}(\alpha_d)}{\partial\alpha_d} = -\frac{(\alpha_d^T\mathbf{s}_d - r_d)\mathbf{s}_d}{\delta^2} - \mathbf{\Sigma}^{-1}(\alpha_d-\mu)$$

4.2.3 Discussion

LRR is neither a purely supervised nor a purely unsupervised model, but it has interesting connections with several existing supervised and unsupervised models.

On the one hand, in terms of its objective function, LRR is similar to a supervised regression model since both are to fit the observed overall ratings (see Eq (2)). However, unlike a regular supervised model, LRR is not to learn a model for prediction of the overall rating of a review; instead, in LRR, we are more interested in analyzing the hidden ratings and weights on each aspect implied by the observed overall ratings (though LRR can also be used to predict the overall rating). From another perspective, according to Eq (1), there is another regression model embedded (with aspect rating as the response variable), but the aspect ratings are not observed directly, thus the only supervision we have is the observed overall rating, which we assume is a weighted-sum of these aspect ratings. This is a major distinction between our LRR model and traditional supervised regression models.

On the other hand, the LRR model also behaves similarly as unsupervised methods in the sense that we do not require availability of training data with known aspect ratings and yet can infer the latent aspect ratings. Specifically, to analyze the latent aspect ratings in a set of reviews, we need to first find the optimal model parameters $\mathbf{\Theta} = (\mu, \mathbf{\Sigma}, \delta^2, \beta)$ for the data set, and then predict the latent ratings $\mathbf{s}_d$ using the learned parameters. In addition, when new data comes, we need to update the parameters accordingly. However, LRR is not a traditional unsupervised method either because we do have the indirect supervision from the overall ratings.

It is also interesting to compare our LRR model with standard topic models, such as LDA[2]. In LDA, we are interested in the latent word distributions that can characterize topics, while in LRR, we attempt to discover word weights that can characterize linguistic patterns associated with aspect ratings. One significant difference between those two models is that LDA is fully unsupervised, but LRR is partially supervised: although we do not have direct supervision on each aspect rating, the overall rating imposes constraints on aspect ratings and thus provides indirect supervision.

4.3 LRR Model Estimation

In the previous section, we discuss how to apply our LRR model to infer aspect weight α_d in each review d when given the model $\mathbf{\Theta} = (\mu, \mathbf{\Sigma}, \delta^2, \beta)$. In this section, we discuss how to estimate these model parameters using the Maximum Likelihood (ML) estimator, i.e., how to find the optimal $\hat{\mathbf{\Theta}} = (\hat{\mu}, \hat{\mathbf{\Sigma}}, \hat{\delta^2}, \hat{\beta})$ that can maximize the probability of observing all the overall ratings.

The log-likelihood function on the whole set of reviews is:

$$\mathcal{L}(D) = \sum_{d\in D}\log p(r_d|\mu, \mathbf{\Sigma}, \delta^2, \beta, \mathbf{W}_d) \quad (7)$$

Thus the ML estimate is

$$\hat{\mathbf{\Theta}} = \arg\max_{\mathbf{\Theta}}\sum_{d\in D}\log p(r_d|\mu, \mathbf{\Sigma}, \delta^2, \beta, \mathbf{W}_d).$$

To compute this ML estimation, we would first randomly initialize all the parameter values to obtain $\mathbf{\Theta}_{(0)}$ and then use the following EM-style algorithm to iteratively update and improve the parameters by alternatively executing the E-step and then M-step in each iteration:

E-Step: For each review d in the corpus, infer aspect rating $\mathbf{s}_d$ and aspect weight α_d based on the current parameter $\mathbf{\Theta}_{(t)}$ (the subscript t indicates the iteration) by using Eq (1) and (6).

M-Step: Given the inferred aspect rating $\mathbf{s}_d$ and aspect weight α_d based on the current parameters $\mathbf{\Theta}_{(t)}$, update the model parameters and obtain $\mathbf{\Theta}_{(t+1)}$ by maximizing the "complete likelihood", i.e., the probability of observing all the variables including the overall ratings r_d and the inferred aspect ratings $\mathbf{s}_d$ and aspect weights α_d for all the reviews.

First, we look at the case of updating the two parameters of the Gaussian prior distribution of the aspect weight α_d. Here our goal is to maximize the probability of observing all the α_d computed in the ***M-Step:*** for all the reviews, thus we have the following updating formulae based on the ML estimation for a Gaussian distribution.

$$\begin{aligned}\mu_{(t+1)} &= \arg\max_{\mu} -\sum_{d\in D}(\alpha_d-\mu)^T\mathbf{\Sigma}^{-1}(\alpha_d-\mu)\\ &= \frac{1}{|D|}\sum_{d\in D}\alpha_d \quad (8)\end{aligned}$$

$\mathbf{\Sigma}_{(t+1)}$ is given by

$$\arg\max_{\mathbf{\Sigma}}\left[-|D|\log\mathbf{\Sigma} - \sum_{d\in D}(\alpha_d-\mu_{(t+1)})^T\mathbf{\Sigma}^{-1}(\alpha_d-\mu_{(t+1)})\right]$$

That is,

$$\mathbf{\Sigma}_{(t+1)} = \frac{1}{|D|}\sum_{d\in D}(\alpha_d-\mu_{(t+1)})(\alpha_d-\mu_{(t+1)})^T \quad (9)$$

Second, we look at how to update β and δ^2. Since α_d is assumed to be known, we can update δ^2 and β to maximize $P(r_d|\alpha_d, \delta^2, \beta, \mathbf{W}_d)$ (defined in Equation 2). Solving this optimization problem, we have the following updating formulae:

$$\delta^2_{(t+1)} = \arg\max_{\delta^2} \left[-|D| \log \delta^2 - \frac{\sum_{d\in D}(r_d - \alpha_d^T \mathbf{s}_d)^2}{\delta^2} \right]$$
$$= \frac{1}{|D|} \sum_{d\in D} (r_d - \alpha_d^T \mathbf{s}_d)^2 \quad (10)$$

$$\beta_{(t+1)} = \arg\max_{\beta} \sum_{d\in D} - \frac{(r_d - \sum_{i=1}^{k} \alpha_{di} \beta_i^T \mathbf{W}_{di})^2}{2\delta^2_{(t+1)}} \quad (11)$$

The closed-from solution for β requires an inversion on a $|V| \times |V|$ matrix, which is expensive to directly compute. To avoid this, we apply the gradient-based method to find the optimal solution of β with the following gradients:

$$\frac{\partial \mathcal{L}(\beta)}{\partial \beta_i} = \sum_{d\in D} (\sum_{i=1}^{k} \alpha_{di} \beta_i^T \mathbf{W}_{di} - r_d) \alpha_{di} \mathbf{W}_{di}$$

The E-step and M-step are repeated until the likelihood value of Eq (7) converges.

5. EXPERIMENT RESULTS

In this section, we first describe the review data set we used for evaluating the LRR model and then discuss the experiment results.

5.1 Data Set and Preprocessing

We crawled 235,793 hotel reviews from *TripAdvisor* in one month period (from February 14, 2009 to March 15, 2009). We chose this data set for evaluation because in addition to the overall rating, reviewers also provided 7 aspect ratings in each review: *value, room, location, cleanliness, check in/front desk, service, business service* ranging from 1 star to 5 stars, which can serve as ground-truth for quantitative evaluation of latent aspect rating prediction. The data is available at `http://times.cs.uiuc.edu/~wang296/Data`.

We first perform simple pre-processing on these reviews: 1) converting words into lower cases; 2) removing punctuations, stop words defined in [1], and the terms occurring less than 5 times in the corpus; 3) stemming each word to its root with *Porter Stemmer* [22].

Since we only have ground-truth aspect ratings on the pre-defined 7 aspects, we have to ensure the same aspects are used in our prediction. Therefore, we manually select a few seed words for each pre-defined aspect and use them as input to the aspect segmentation algorithm described in Section 4.1, where we set the selection threshold p=5 and iteration step limit I=10 in our experiments. Table 1 shows the initial aspect terms used.

Table 1: Aspect Seed Words

Aspects	Seed words
Value	value, price, quality, worth
Room	room, suite, view, bed
Location	location, traffic, minute, restaurant
Cleanliness	clean, dirty, maintain, smell
Check In/Front Desk	stuff, check, help, reservation
Service	service, food, breakfast, buffet
Business service	business, center, computer, internet

After aspect segmentation, we discarded those sentences that fail to be associated with any aspect. If we require all the reviews contain all the 7 aspect descriptions, there would be only 780 reviews left covering 184 hotels. To avoid spareness and missing aspect descriptions in the review, we thus concatenated all the reviews commenting on the same hotel together as a new "review" (we call it "*h-review*") and average the overall/aspect ratings over them as the ground-truth ratings. After these processings, we have a corpus with 1,850 hotels ("*h-review*") and 108,891 reviews; the details are illustrated in Table 2.

Table 2: Evaluation Corpus Statistics

Number of Hotels	1850
Number of Reviews	108891
Sentences per Review	8.21±4.02
Words per Aspect	9.57±6.21

5.2 Qualitative evaluation

We first show three sample results generated by the proposed LRR model for qualitative evaluation.

Aspect-level Hotel Analysis: One simple method to judge the quality of a given hotel is to check its overall rating. However, this rough criterion would lose the detailed assessments about the quality of different aspects: it fails to tell the differences among the hotels in the aspect level. To examine the capability of our LRR model to make this distinction, we randomly select 3 hotels with the same overall rating 4.2 (on average) but different aspect ratings, and apply LRR to predict the hidden aspect ratings. The prediction results are shown in Table 3 with ground-truth ratings in parenthesis (due to the space limitation, we only show the result for the first four aspects).

Table 3: Aspect rating prediction for different hotels

Hotel	Value	Room	Location	Cleanliness
Grand Mirage Resort	4.2(4.7)	3.8(3.1)	4.0(4.2)	4.1(4.2)
Gold Coast Hotel	4.3(4.0)	3.9(3.3)	3.7(3.1)	4.2(4.7)
Eurostars Grand Marina Hotel	3.7(3.8)	4.4(3.8)	4.1(4.9)	4.5(4.8)

We can see that although these three hotels have the same overall ratings, they differ in detailed aspects: *Grand Mirage Resort* and *Gold Coast Hotel* both have better prices (high ratings for "value"), while *Eurostars Grand Marina Hotel* has a better location and room conditions. This information is valuable to the users who have different requirements on aspects.

Reviewer-level Hotel Analysis: Even for the same hotel, different reviewers may hold different opinions on an aspect. The LRR model can further support such detailed analysis by predicting aspect rating at the individual review level. To demonstrate this function, we select the subset of reviews including all 7 aspect descriptions (780 reviews and 184 hotels) to examine the variances across different types of reviewers. In Table 4, two reviewers both give *Hotel Riu Palace Punta Cana* an overall rating of 4 stars, but they do not agree on every aspect: reviewer 1 evaluated the hotel's cleanliness better than other aspects while reviewer 2 thought its value and location were the best part. Identifying such disagreement and providing the evidence (aspect

ratings) would better help users make informed decisions based on reviews.

Table 4: Aspect rating prediction for different reviewer of Hotel Riu Palace Punta Cana

Reviewer	Value	Room	Location	Cleanliness
Mr.Saturday	3.7(4.0)	3.5(4.0)	3.7(4.0)	5.8(5.0)
Salsrug	5.0(5.0)	3.0(3.0)	5.0(4.0)	3.5(4.0)

Corpus Specific Word Sentimental Orientation: In addition to predicting the latent aspect ratings for the whole text, LRR can also identify the word's sentimental orientations. Being different from traditional unsupervised sentiment classification methods, which rely on a predefined lexicon, LRR can uncover such sentimental information directly from the given data. In Table 5, we show some interesting results of LRR by listing the top 5 words with positive weights and top 5 words with negative weights for each aspect, and we compare them with the opinion annotation in *SentiWordNet* [8]. (Due to the space limitation, we only show term weights for the first 4 aspects.) We can find some interesting results: the word "ok" is positive as defined by *SentiWordNet*, but in our corpus reviewers use this word to comment on something barely acceptable; words "linen", "walk" and "beach" do not have opinion annotations in *SentiWordNet* since they are nouns, while LRR assigns them positive sentimental orientations likely because "linen" may suggest the "cleanliness" condition is good and "walk" and "beach" might imply the location of a hotel is convenient. Thus, LRR can provide us with word orientation information that is specific to the given domain, which may be useful for augmenting an existing sentiment lexicon for specific domains.

Table 5: Term weight under aspects

Value	Rooms	Location	Cleanliness
resort 22.80	view 28.05	restaurant 24.47	clean 55.35
value 19.64	comfortable 23.15	walk 18.89	smell 14.38
excellent 19.54	modern 15.82	bus 14.32	linen 14.25
worth 19.20	quiet 15.37	beach 14.11	maintain 13.51
quality 18.60	spacious 14.25	perfect 13.63	spotlessly 8.95
bad -24.09	carpet -9.88	wall -11.70	smelly -0.53
money -11.02	smell -8.83	bad -5.40	urine -0.43
terrible -10.01	dirty -7.85	mrt -4.83	filthy -0.42
overprice -9.06	stain -5.85	road -2.90	dingy -0.38
cheap -7.31	ok -5.46	website -1.67	damp -0.30

5.3 Quantitative Evaluation

Baseline Algorithms: To the best of our knowledge, no previous work has attempted to solve the same problem as ours. The closest work is [17], in which the authors proposed two methods, i.e. **Local prediction** and **Global prediction**, to solve a similar problem. Therefore we take these two methods as our baseline methods for comparison. We also include another baseline approach, in which we take the overall rating of review as the aspect ratings for the review to train a supervised model. We implement this method using the Support Vector Regression (SVR) model [3] and name it as **SVR-O**. Besides, as an upper-bound, we also test a fully supervised algorithm **SVR-A**, i.e. SVR model fed with the aspect ratings in the ground-truth for training, and compare it with what LRR can achieve without such supervision. We use RBF kernel with default parameters implemented in the libsvm package [4] for both SVR-O and SVR-A. All the models are evaluated on the same data set: for LRR and the two methods in [17], we use the whole data set for both training and testing; for SVR-based models, we perform 4-fold cross validation and report the mean value of performance.

Measures: We use four different measures to quantitatively evaluate different methods, including (1) mean square error on aspect rating prediction (Δ^2_{aspect}), (2) aspect correlation inside reviews (ρ_{aspect}), (3) aspect correlation across reviews (ρ_{review}) and (4) Mean Average Precision (MAP) [11], a frequently used measure in information retrieval for evaluating ranking accuracy.

Formally, suppose $\mathbf{s}^*_{di}$ is the ground-truth rating for aspect A_i. Δ^2_{aspect} directly measures the difference between the predicted aspect rating $\mathbf{s}_{di}$ and $\mathbf{s}^*_{di}$, and is defined as:

$$\Delta^2_{aspect} = \sum_{d=1}^{|D|}\sum_{i=1}^{k}(\mathbf{s}_{di} - \mathbf{s}^*_{di})^2/(k \times |D|)$$

ρ_{aspect} aims to measure how well the predicted aspect ratings can preserve the relative order of aspects within a review given by their ground-truth ratings. For example, in a review, the reviewer may have liked the location better than cleanliness, and ρ_{aspect} would assess whether the predicted ratings would give the same preference order. ρ_{aspect} is defined as:

$$\rho_{aspect} = \sum_{d=1}^{|D|}\rho_{\mathbf{s}_d,\mathbf{s}^*_d}/|D|$$

where $\rho_{\mathbf{s}_d,\mathbf{s}^*_d}$ is the Pearson correlation between two vectors $\mathbf{s}_d$ and $\mathbf{s}^*_d$.

Similarly, ρ_{review} is defined as the following Pearson correlation:

$$\rho_{review} = \sum_{i=1}^{k}\rho(\vec{\mathbf{s}_i}, \vec{\mathbf{s}^*_i})/k$$

where $\vec{\mathbf{s}_i}$ and $\vec{\mathbf{s}^*_i}$ are the predicted and ground-truth rating vectors for aspect A_i across all the reviews. It tells us whether the predicted ratings and the ground-truth ratings for aspect A_i would give a similar ranking of all the reviews in this aspect. Such ranking can answer questions such as "Which hotel has the best service?".

However, ρ_{review} puts equal emphasis on all items and does not reflect the quality of the top ranked ones, which intuitively is more important from a user's perspective. Therefore, we also use MAP to evaluate the model's ranking accuracy of reviews. More specifically, we treat the top 10 reviews ranked by the ground-truth aspect ratings as the relevant reviews, and see whether we would be able to rank these top 10 reviews on the top, if we use predicted aspect ratings to rank the reviews. We rank all the hotels according to each of the 7 aspects and calculate MAP at the cutoff of 10 reviews.

Result Analysis: We report the performance of all five algorithms measured by four metrics in Table 6. Since SVR-A is fully supervised while others are not, we list it separately on the last line as an upper bound. We also highlight the best performance in each measure for all the non SVR-A models in bold.

Table 6: Comparison with other models

Method	Δ^2_{aspect}	ρ_{aspect}	ρ_{review}	MAP@10
Local prediction	**0.588**	0.136	**0.783**	0.131
Global prediction	0.997	0.279	0.584	0.000
SVR-O	0.591	0.294	0.581	0.358
LRR	0.896	**0.464**	0.618	**0.379**
SVR-A	0.306	0.557	0.673	0.473

A general observation is that LRR performs much better than all other non SVR-A models on ρ_{aspect} and MAP@10, but it does not perform the best on Δ^2_{aspect} and ρ_{review}. High ρ_{aspect} means that LRR can better distinguish the ratings of different aspects within a review. Note that such information about the relative preferences on different aspects cannot be obtained with only an overall rating. In addition, the high MAP@10 values show that LRR also can better retrieve the top 10 hotels based on each aspect rating than other methods, leading to more useful ranking results from a user's perspective since it is the top ranked results that would affect user satisfaction most.

Note that Δ^2_{aspect} measures the deviation of each predicted aspect rating and ground-truth rating independently, thus it does not reflect how well the relative order of aspects is preserved. Consider, e.g., there are only three aspects. One review has an overall rating of 4 and ground-truth aspect ratings of $(3, 4, 5)$. A naive prediction of $(4, 4, 4)$, which cannot differentiate different aspects, would have $\Delta^2_{aspect} = 0.67$, but another prediction $(2, 3, 4)$, which can tell the real difference between aspects, would have $\Delta^2_{aspect} = 1$, which is higher (thus worse). Indeed, it can be observed that the Local prediction method achieves the best Δ^2_{aspect} of 0.588, but it also under-performs other methods by having the lowest ρ_{aspect}, which is actually a more important factor in applications.

By further investigating the ranking accuracy of reviews based on predicted aspect ratings, we can see that the two measures ρ_{review} and MAP@10 generate different conclusions. This is expected because ρ_{review} measures the overall correlation of all the 1850 *h-reviews* while MAP@10 only cares about top 10. Local prediction does score the highest in ρ_{review} but it scores poorly in terms of MAP@10. This indicates that it outperforms LRR at lower rankings instead of the top ones, which users usually care about most.

Note that SVR fed with overall ratings did not achieve desirable performance, which to some extent confirms our assumption that there are differences between the aspect ratings and overall ratings. As a result, looking at only the overall ratings is not sufficient. Finally, not surprisingly, LRR does not perform as well as SVR-A, which was trained with ground-truth aspect ratings. However, LRR does not require any annotated aspect ratings for training, and can thus be applied to more application scenarios than SVR-A.

Computational Complexity: Efficiency of mining algorithms is an important issue in real applications. The major computational overhead of LRR is in solving the nonlinear optimization problems in Eq (6) and (11). The convergence rate of training procedure for LRR depends on the size of model parameter $\mathbf{\Theta}$, number of reviews |D| and iteration step limit l. The complexity is roughly estimated as $O(k(n + k + 1)|D|l)$, which is linear with the number of reviews. For our data set, the algorithm finishes in less than 3 minutes on a Pentium 4 2.8G cpu/2GB memory desktop.

5.4 Applications

The detailed understanding of opinions obtained using LRR can be potentially useful for many applications. Here we present three sample applications.

Aspect-Based Summarization:

Since LRR can infer the aspect ratings $\mathbf{s}_d$ for each review d, we can easily aggregate the aspect ratings of all reviews about the same hotel to generate one numerical rating for each aspect of the hotel (e.g. $\frac{1}{|D|}\sum_{d\in D}\mathbf{s}_d$). Such aspect ratings for a hotel can be regarded as a concise aspect-based opinion summary for the hotel. On top of that, we can also select the sentences in each review about the given hotel by calculating the aspect scores according to Eq (1), and selecting the highest and lowest scored sentences for each aspect to help users better understand the opinions on different aspects.

We show a sample aspect-based summary generated in this way in Table 7. We can see that reviewers agree that Hotel Max's price is excellent when considering its great location in Seattle. However, there is also room for improvement: poor heating system and the charge for Internet access. This kind of detailed information would be very useful to the users for digesting the essential opinions in a large number of reviews.

User Rating Behavior Analysis:

By inferring hidden aspect weights α_d for each individual review, we can know the relative emphasis placed by a reviewer on different aspects, which can be regarded as knowledge about a user's rating behavior. One potential application of analyzing the reviewers' rating behavior is to discover what factors have most influence on reviewers' judgment when they make such evaluations. To look into this, we selected two groups of hotels with different price ranges: one group have prices over $800, which would be called "expensive hotels," while the other group have prices below $100 and would be called "cheap hotels." For each group, we then selected top 10 and bottom 10 hotels based on their average overall ratings, resulting in four subgroups of hotels. We show the average aspect weights α_d of these four different subgroups of hotels in Table 8. (In the group of "expensive hotels," the lowest overall rating is 3 stars.)

Table 8: User behavior analysis

	Expensive Hotel		Cheap Hotel	
Aspect	5 Stars	3 Stars	5 Stars	1 Star
Value	0.134	0.148	0.171	0.093
Room	0.098	0.162	0.126	0.121
Location	0.171	0.074	0.161	0.082
Cleanliness	0.081	0.163	0.116	0.294
Service	0.251	0.101	0.101	0.049

It is interesting to note that reviewers give the "expensive hotels" high ratings mainly due to their nice services and locations, while they give low ratings because of undesirable room condition and overprice. In contrast, reviewer give the "cheap" hotels high ratings mostly because of the good price/value and good location, while giving low ratings for its poor cleanliness.

Additionally, those numerical ratings may contain different

Table 7: Aspect-based Comparative Summarization (Hotel Max in Seattle)

Aspect	Summary	Rating
Value	Truly unique character and a great location at a reasonable price Hotel Max was an excellent choice for our recent three night stay in Seattle.	3.1
	Overall not a negative experience, however considering that the hotel industry is very much in the impressing business there was a lot of room for improvement.	1.7
Room	We chose this hotel because there was a Travelzoo deal where the Queen of Art room was $139.00/night.	3.7
	Heating system is a window AC unit that has to be shut off at night or guests will roast.	1.2
Location	The location ,a short walk to downtown and Pike Place market , made the hotel a good choice.	3.5
	when you visit a big metropolitan city, be prepared to hear a little traffic outside!	2.1
Business Service	You can pay for wireless by the day or use the complimentary Internet in the business center behind the lobby though.	2.7
	My only complaint is the daily charge for internet access when you can pretty much connect to wireless on the streets anymore.	0.9

meanings across various reviewers: users with a low budget might give a cheaper hotel 5 stars rating of "*value*", while some others seeking for better service might also give a more expensive hotel 5 stars rating for its "*value*". Only predicting the ratings for each aspect is not enough to reveal such subtle differences between the users, but the inferred aspect weights can be useful for understanding such differences as a user with a low budget presumably is more likely to place more weight on "*value*" than on "*service*". To look into this, we selected the hotels with the same 5-star ratings for the *"value"* aspect from 4 different cities: *Amsterdam*, *Barcelona*, *Florence* and *San Francisco*, which have the largest numbers of hotels in our corpus. We then rank these hotels according to their ratios of value/location weight, value/room weight and value/service weight, respectively, and for each ratio, we calculate the average real price of the top-10 hotels and bottom-10 hotels, respectively. The results are shown in Table 9.

We find that hotels with relatively higher *"value"* weights tend to have a lower price while the hotels with higher *"location"*, *"room"* and *"service"* weights tend to have a higher price, suggesting that even though these hotels all have the same ratings on the "value" aspect, people who place more weight on value than on other aspects (i.e., who really cares about price) would prefer a cheaper hotel, while those who placed a higher weight on another aspect such as "location" or "service" (than on "price") would accept a higher price. Thus the inferred aspect weights α_d can be very useful for revealing users' rating behavior.

Personalized ranking: Ranking hotels based on their inferred ratings on each different aspect is already very useful to users. Here we show that the learned weights on different aspects at the level of each individual review would enable us to further personalize such ranking by selectively using only the reviews written by reviewers whose rating behavior is similar to a current user. Specifically, given a specific user's weighting preference as a query, we can select the reviewers whose weighting preference is similar, and rank the hotels only based on the reviews written by the subset of reviewers with a similar preference.

To show the effectiveness of LRR in supporting such personalized ranking, consider a sample query: Query = {value weight:0.9, others:0.016}, which indicates that the user cares most about the "value" and does not care about other aspects. We use two different ranking approaches: 1) approach 1 would simply rank the hotels by the predicted aspect rating without considering the input query; 2) approach 2 would select the top 10% reviewers who have the closet aspect weights (i.e. α_d) to the query and predict the associated hotels aspect ratings only based on those selected reviews. Using both approaches, we rank hotels based on the weighted sum of the predicted aspect ratings for all the aspects with the weight defined in the query (thus the rating on "value" would contribute most to scoring), and show the top-5 returned hotels using each approach in Table 10.

Table 9: Subgroups of hotels with 5-star on "value"

City	AvgPrice	Group	Val/Loc	Val/Rm	Val/Ser
Amsterdam	241.6	top-10	190.7	214.9	221.1
		bot-10	270.8	333.9	236.2
Barcelona	280.8	top-10	270.2	196.9	263.4
		bot-10	330.7	266.0	203.0
San Fran.	261.3	top-10	214.5	249.0	225.3
		bot-10	321.1	311.1	311.4
Florence	272.1	top-10	269.4	248.9	220.3
		bot-10	298.9	293.4	292.6

Table 10: Personalized Hotel Ranking

	Hotel	Overall Rating	Price	Location
Approach 1	Majestic Colonial	5.0	339	Punta Cana
	Agua Resort	5.0	753	Punta Cana
	Majestic Elegance	5.0	537	Punta Cana
	Grand Palladium	5.0	277	Punta Cana
	Iberostar	5.0	157	Punta Cana
Approach 2	Elan Hotel Modern	5.0	216	Los Angeles
	Marriott San Juan Resort	4.0	354	San Juan
	Punta Cana Club	5.0	409	Punta Cana
	Comfort Inn	5.0	155	Boston
	Hotel Commonwealth	4.5	313	Boston

It is interesting to see that although the top-5 results from approach 1 all have 5-star overall ratings (presumably they also have high ratings on "value" since the ranking is based on weights specified in the query), their prices tend to be much higher than the top-5 results returned from approach 2; indeed, the average price of the top-5 hotels from approach 1 is $412.6, while that of the top-5 hotels from approach 2 is only $289.4, which is much lower. (The average price of all the hotels in the data set is $334.3). Intuitively, for this sample query, the results of approach 2 are more useful to the user. This means that due to the selective use of reviews

from reviewers who have a similar weight preference to the query, approach 2 is able to personalize the ranking and correctly place more weight on the "value" aspect to ensure that the top-ranked hotels really have relatively low prices.

6. CONCLUSIONS

In this paper, we defined a novel text mining problem named Latent Aspect Rating Analysis (LARA) to analyze opinions expressed in online reviews at the level of topical aspects. LARA takes a set of review texts with overall ratings and a specification of aspects as input, and discovers each individual reviewer's latent ratings on the given aspects and the relative emphasis a reviewer has placed on different aspects. To solve this problem, we proposed a novel Latent Rating Regression (LRR) model. Our empirical experiments on a hotel review data set show that the proposed LRR model can effectively solve the problem of LARA, revealing interesting differences in aspect ratings even when the overall ratings are the same as well as differences in user's rating behavior. The results also show that the detailed analysis of opinions at the level of topical aspects enabled by the proposed model can support multiple application tasks, including aspect opinion summarization, ranking of entities based on aspect ratings, and analysis of reviewers rating behavior.

Our work opens up a novel direction in text mining where the focus is on analyzing latent ratings in opinionated text. There are many interesting future research directions to further explore. For example, although we defined LARA based on reviews, LARA is clearly also applicable to any set of opinionated text (e.g. weblogs) documents with overall ratings to achieve detailed understandings of opinions. It would be interesting to explore other possible application scenarios. Besides, our LRR model is not strictly limited to word features, other kinds of features could be easily embedded into this model. Also, in our definition of LARA, we assumed that the aspects are specified in the form of a few keywords. While this is feasible and gives users control over the aspects to be analyzed, there may also be situations where such keywords are not available. It would be very interesting to further explore LARA in such a setting where we would aim at discovering the latent rating aspects in addition to the latent ratings and weights.

7. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their useful comments. This paper is based upon work supported in part by an IBM Faculty Award, an Alfred P. Sloan Research Fellowship, and by the National Science Foundation under grants IIS-0347933, IIS-0713581, IIS-0713571, and CNS-0834709.

8. REFERENCES

[1] Onix text retrieval toolkit stopword list. `http://www.lextek.com/manuals/onix/stopwords1.html`.

[2] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] H. Cui, V. Mittal, and M. Datar. Comparative experiments on sentiment classification for online product reviews. In *Twenty-First National Conference on Artificial Intelligence*, volume 21, page 1265, 2006.

[6] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03*, pages 519–528, 2003.

[7] A. Devitt and K. Ahmad. Sentiment polarity identification in financial news: A cohesion-based approach. In *Proceedings of ACL'07*, pages 984–991, 2007.

[8] A. Esuli and F. Sebastiani. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, 2006.

[9] A. Goldberg and X. Zhu. Seeing stars when there arenąt many stars: Graph-based semi-supervised learning for sentiment categorization. In *HLT-NAACL 2006 Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing*, 2006.

[10] M. Hu and B. Liu. Mining and summarizing customer reviews. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *KDD*, pages 168–177. ACM, 2004.

[11] K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of SIGIR'00*, pages 41–48. ACM, 2000.

[12] N. Jindal and B. Liu. Identifying comparative sentences in text documents. In *Proceedings of SIGIR '06*, pages 244–251, New York, NY, USA, 2006. ACM.

[13] H. Kim and C. Zhai. Generating Comparative Summaries of Contradictory Opinions in Text. In *Proceedings of CIKM'09*, pages 385–394, 2009.

[14] S. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of COLING*, volume 4, pages 1367–1373, 2004.

[15] K. Lerman, S. Blair-Goldensohn, and R. T. McDonald. Sentiment summarization: Evaluating and learning user preferences. In *EACL*, pages 514–522, 2009.

[16] B. Liu, M. Hu, and J. Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *WWW '05*, pages 342–351, 2005.

[17] Y. Lu, C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In *Proceedings of WWW'09*, pages 131–140, 2009.

[18] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima. Mining product reputations on the web. In *KDD '02*, pages 341–349, 2002.

[19] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, pages 115–124, 2005.

[20] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *EMNLP 2002*, pages 79–86, 2002.

[21] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of HLT '05*, pages 339–346, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[22] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130 – 137, 1980.

[23] B. Snyder and R. Barzilay. Multiple aspect ranking using the good grief algorithm. In *Proceedings of NAACL HLT*, pages 300–307, 2007.

[24] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL '08*, pages 308–316.

[25] Y. Yang and J. O.Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of ICML'97*, pages 412 – 420, 1997.

[26] L. Zhuang, F. Jing, and X. Zhu. Movie review mining and summarization. In *Proceedings of CIKM 2006*, page 50. ACM, 2006.