Context-Aware Review Helpfulness Rating Prediction

Jiliang Tang, Huiji Gao, Xia Hu and Huan Liu
Computer Science and Engineering, Arizona State University, Tempe, AZ, USA
{jiliang.tang, huiji.gao, xia.hu, huan.liu}@asu.edu

ABSTRACT
Online reviews play a vital role in the decision-making process for online users. Helpful reviews are usually buried in a large number of unhelpful reviews, and with the consistently increasing number of reviews, it becomes more and more difficult for online users to find helpful reviews. Therefore most online review websites allow online users to rate the helpfulness of a review and a global helpfulness score is computed for the review based on its available ratings. However, in reality, user-specified helpfulness ratings for reviews are very sparse - a few reviews attract large numbers of helpfulness ratings while most reviews obtain few or even no helpfulness ratings. The available helpfulness ratings are too sparse for online users to assess the helpfulness of reviews. Also the helpfulness of a review is not necessarily equally useful for all users and users with different background may treat the helpfulness of a review very differently. The user idiosyncracy of review helpfulness motivates us to study the problem of review helpfulness rating prediction in this paper. We first identify various types of context information, model them mathematically, and propose a context-aware review helpfulness rating prediction framework CAP. Experimental results demonstrate the effectiveness of the proposed framework and the importance of context awareness in solving the review helpfulness rating prediction problem.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Information filtering; J.4 [Computer Application]: Social and Behavioral Sciences

Keywords
Review Rating Prediction, Social Context, Review Recommendation, Content Context

1. INTRODUCTION
Reviews, providing experiences with and opinions about products or services from other users, play a crucial role in online communities such as e-commerce and product review sites, where users rely on reviews in their decision-making process. For example, users will select restaurants with good reviews in Yelp, and reviews about products in eBay are important sources of information for users to make purchases. However, helpful reviews are usually buried in large numbers of useless reviews [15], and with the availability of massive reviews, it becomes increasingly difficult for online users to find helpful reviews.

In an attempt to help online users identify helpful reviews, most online review websites implement a mechanism to allow users to rate the helpfulness of a review and then a global helpfulness score is computed for the review such as “20 out of 30 people found the following review helpful” in eBay and a score from 0 to 5 in Ciao. In reality, a large proportion of reviews obtain few or no helpfulness ratings, particularly the more recent ones and the available helpfulness ratings are too sparse for online users to assess the helpfulness of reviews [15]. For example, it is difficult for users to assess the helpfulness of a review in eBay with a score of “1 out of 1 people found review helpful”. There is recent work automatically predicting a global helpfulness score for a review [9, 13, 15]. However, a review is not necessarily equally useful for all users. For example, in eBay, a review’s helpfulness can have a score of “500 out of 1000 people found the following review helpful”. This indicates that the other half do not think the review helpful or are indifferent. This user idiosyncrasy of review helpfulness motivates us to study if we can predict review helpfulness rating for each user.

We choose a product review site, a classical type of online review websites, to investigate if review helpfulness rating prediction can help mitigate the problem caused by user idiosyncrasy. Figure 1(a) gives an overview of product review sites where users have four different behaviors - connecting to other users, writing reviews, rating the helpfulness of reviews, and rating items. Figure 1(b) depicts the user helpfulness rating behavior and there are two types of ratings including item ratings and review helpfulness ratings. The review helpfulness rating is fundamentally different from the item rating. The former indicates “how does a user X rate a review from another user Y?” while the latter denotes “how does a user X rate an item?” These differences not only are useful to differ our studied problem from item rating prediction problem, but also present unique opportunities for us to investigate the review helpfulness rating prediction problem. First, the texts of reviews can affect how users rate the helpfulness of reviews [13], thus provide content context about reviews. Second, users play two roles in review helpfulness rating: authors - users who write reviews, and raters - users who rate reviews. Both raters and authors can rate items,
and raters may be related to authors such as raters may connect to authors. Raters, authors and their relations provide rich social context about reviews. For example, authors with high reputations are likely to write helpful reviews [4], and raters might think of reviews from their connected authors more helpful. Therefore the dual roles of users expand the horizon of social context in helpfulness rating, providing a new perspective to exploit social context.

The availability of content context and social context provides unique opportunities but also brings about new challenges: (1) what types of social context can be extracted from the dual roles of users in helpfulness rating, and (2) how to model content context and various types of social context mathematically for prediction. Addressing these two challenges, we propose a framework for the helpfulness rating prediction problem by exploiting context information. Our contributions are summarized next.

- Defining the problem of review helpfulness rating prediction with context awareness formally;
- Analyzing various types of social context and providing a way to formulate them mathematically;
- Proposing a Context-Aware helpfulness rating Prediction framework (CAP) that exploits content context with various types of social context; and
- Evaluating the framework in Ciao, a real-world product review site, to understand the working of CAP and the importance of context awareness in the problem of review helpfulness rating prediction.

The rest of this paper is organized as follows. Section 2 defines our problem formally. Section 3 describes the dataset and analyzes various types of social context. Section 4 introduces how to formulate context information mathematically and a context-aware helpfulness rating prediction framework. Section 5 presents experimental results and our observations. Section 6 briefly reviews related work. Section 7 concludes this study with future work.

2. PROBLEM STATEMENT

Typically there are three types of objects on product review sites. Let \( U = \{u_1, u_2, \ldots, u_n\} \) be the set of users, \( P = \{p_1, p_2, \ldots, p_m\} \) be the set of items, and \( R = \{r_1, r_2, \ldots, r_N\} \) be the set of reviews where \( n, m \) and \( N \) are the numbers of users, items and reviews, respectively. Users in product review websites have four behaviors. First, users can connect to each other and we use \( T \in \mathbb{R}^{n \times n} \) to represent their social relations where \( T_{ij} = 1 \) if \( u_i \) creates a connection to \( u_j \) and zero otherwise. Second, users can rate items and \( R \in \mathbb{R}^{n \times m} \) is introduced to denote item rating where \( R_{ij} \) is the item rating if \( u_i \) gives \( p_j \) a rating. Third, users can rate the helpfulness of reviews and \( H \in \mathbb{R}^{n \times N} \) is employed to represent helpfulness rating where \( H_{ij} \) is the helpfulness rating if \( u_i \) gives \( r_j \) a rating. For both \( R \) and \( H \), we adopt a symbol “?” to represent unknown ratings. Finally, users can write reviews and we use \( A \in \mathbb{R}^{n \times N} \) to denote the author-review relations where \( A_{ij} = 1 \) indicates that \( r_j \) is written by \( u_i \) and zero otherwise. For the \( j \)-th review \( r_j \), we use \( x_j \) to represent its textual feature vector. We use \( O = \{(u_i, r_j, u_k) | H_{ij} \neq ? \} \) and \( Q = \{(u_i, r_j, u_k) | H_{ij} = ? \} \) to denote the set of known and unknown helpfulness ratings respectively, where \( u_k \) is the author of \( r_j \). Although we choose product view sites to study the problem, the framework proposed in this paper can be applied to other online review websites.

With the notations above, our problem can be stated as: given the known helpfulness rating set \( O \) and its contextual awareness including the social network \( T \), the item ratings \( R \), the author-review relations \( A \), and the review content \( \{x_j\}_{j=1}^{N} \), we aim to predict unknown helpful ratings for triples \( (u_i, r_j, u_k) \in Q \) by exploiting the known helpfulness rating set \( O \) and context information \( \{T, R, A, \{x_j\}_{j=1}^{N}\} \).

3. SOCIAL CONTEXT ANALYSIS

The dual roles of users in the studied problem bring about unique challenges and opportunities to exploit social context. In this section, we conduct preliminary social context analysis to seek a solution to the first challenge - what types of social context can be extracted.

3.1 Dataset

For the purpose of this study, we crawled a data set from Ciao, a popular product review site. We started with a set of the most active users and then did breadth-first search until no new users could be found. For each user, we collect his/her profile, social networks and item rating entities. For each item rating entity, we collected the time point when this entity was created, item name, the category of the item, the rating score and the associated review. For each review, we collected its textual content, raters, their helpfulness ratings respectively, where \( u_k \) is the author of \( r_j \). Some statistics about the dataset are shown in Table 1 where the number of user, review, item, typical and helpfulness ratings are 43,666, 302,232, 112,804, 302,232 and 8,894,899 respectively. The average rating score per review is 3.9126.

<table>
<thead>
<tr>
<th>Table 1: Statistics of the Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Users</td>
</tr>
<tr>
<td># of Reviews</td>
</tr>
<tr>
<td># of Items</td>
</tr>
<tr>
<td># of Item Ratings</td>
</tr>
<tr>
<td># of Helpful Ratings</td>
</tr>
<tr>
<td>Ave Rating Score per Review</td>
</tr>
<tr>
<td># of Connections</td>
</tr>
</tbody>
</table>

In Ciao, users give scores from 0 to 5 to indicate the helpfulness of reviews, and we find that the majority (80.74%) of helpfulness ratings are 3 and 4. We compute the number of helpfulness ratings given by each user and the distribution suggests a power-law-like distribution: a few users con-
Authors and raters provide rich social context about reviews. For example, authors are likely to rate the helpfulness of a review similarly to how their trust networks do [17] and some users have a propensity to rate reviews higher or lower [5].

- **Rater Context.** Rater context is extracted to capture context information from raters. For example, raters are likely to rate the helpfulness of a review similarly to how their trust networks do [17] and some users have a propensity to rate reviews higher or lower [5].

- **Connection Context.** If raters connect to the author of a review, how will they think of the helpfulness of the review? Connection context is employed to capture context information about social relations between raters and authors.

- **Preference Context.** If raters have similar preferences with the author of a review, how will they think of the helpfulness of the review? Preference context is introduced to exploit the context information of raters and authors of reviews with similar preferences.

Most existing tasks exploiting social context only consider users as either authors [15, 12] or raters [10, 16]. For example, author context is exploited as the social features in review quality prediction problem [12], while rater context is widely exploited in the item rating prediction problem [10, 1]. Author context and rater context are seldom both investigated in one problem, while our problem naturally involves both of them. There are recent studies exploiting relations between users [11, 17]. Only considering one role of users, they utilize relations between two raters [11, 17] or two authors [15]. However, we consider relations between raters and authors in our problem, which is rarely studied before. In the following two subsections, we focus our attention on the analysis of connection context and preference context.

### 3.2 Various Types of Social Context

Authors, raters, and their relations provide rich social context about reviews. For example, authors are likely to rate the helpfulness of reviews from the authors with high reputation more helpful [4], and the helpfulness of reviews from the same authors is likely to be consistent [15].

- **Author Context.** Author context is defined to capture context information from the authors of reviews. For example, authors are likely to rate the helpfulness of reviews from authors with high reputation more helpful [4], and the helpfulness of reviews from the same authors is likely to be consistent [15].

- **Rater Context.** Rater context is extracted to capture context information from raters. For example, raters are likely to rate the helpfulness of reviews from the authors with high reputation more helpful [4], and the helpfulness of reviews from the same authors is likely to be consistent [15].

### 3.3 Connection Context Analysis

Given that raters connect to the author of a review, we investigate how the raters will rate the helpfulness of the review. For each review, we divide its raters into two groups: connection group $G_c$ containing raters who have connections with its author, and non-connection group $G_n$ containing raters without relations with its author. To analyze connection context, we select reviews with both $G_c$ and $G_n$, not null, including 74.45% of all reviews. This observation suggests that most reviews involve connection context. On average, 33.17% of helpfulness ratings for a review are from the connection group $G_c$.

For the $j$-th review $r_j$, we use connection rating $T_j$ and non-connection rating $R_j$ to represent the average helpfulness ratings from raters in $G_c$ and $G_n$, respectively. Assume that $t$ and $r$ are the vectors of all connection ratings ($T_j$) and non-connection ratings ($R_j$), respectively. We check the means of $t$ and $r$ and find that the mean of $t$ is larger than that of $r$. To study the significance, we also conduct a two-sample t-test on the vectors $t$ and $r$. The null hypothesis is $H_0: t = r$, and the alternative hypothesis is $H_1: t > r$. The result shows that there is strong evidence to reject the null hypothesis with significance level $α = 0.01$, indicating that $t$ is significantly larger than $r$. With the evidence from the means and t-test result, we conclude that raters are more likely to think of reviews from their connected authors more helpful.

Users may have heterogeneous connection strengths with their social networks [22]. For each review, we further divide the connection group $G_c$ into strong connection groups and weak connection groups based on the metrics introduced in [22]. We do similar analysis on these two groups to that on $G_c$ and $G_n$. To save space, we ignore the detailed results and directly give Observation 2 - the more strongly raters connect to an author, the more helpful raters consider the reviews from the author.

### 3.4 Preference Context Analysis

Given that raters have similar preferences to the author of a review, we investigate how the raters will think of the helpfulness of the review. To answer this question, we first need to define the measure of preference similarity between raters and authors. In the context of product review sites, user preferences can be extracted from the item rating behavior [23]. For example, if two users rate the same items similarly, they have similar preferences or opinions. Therefore, we use item rating similarity to measure preference similarity between the raters and authors. For the $i$-th user $u_i$, we first calculate his/her cosine item rating similarities with other users $s_{ik}$, $\forall k \neq i$, and then we choose $j$-th user $u_j$ as $u_i$’s preference similar user if $s_{ij} > \frac{1}{|\mathcal{K}|} \sum_{k \neq i} s_{ik}$, where $\frac{1}{|\mathcal{K}|} \sum_{k} s_{ik}$ is the average item similarity between $u_i$ and other users. Let $P \in \mathbb{R}^{n \times n}$ denote the preference relation.
matrix where $P_{i,j} = 1$ if $u_j$ is a preference similar user of $u_i$ and zero otherwise.

Similar to connection context analysis, for each review, we split its raters into two groups: preference group $G_u$, the group of raters who are the preference similar users of the author, and non-preference group $G_v$, the group of users who are not the preference similar users of the author. To conduct preference context analysis, we choose reviews with both $G_u$ and $G_v$ not null, including 89.08% of all reviews. On average, 48.93% of helpfulness ratings for a review are from the preference group $G_u$. These observations reveal that most reviews have preference context. For each review, we check the rater overlap between preference group $G_u$ and connection group $G_v$ in context connection analysis and find that the average overlap is 9.23%. This observation suggests that preference context is very different from connection context and it is necessary to study them separately.

For the $i$-th review, we use preference rating $O_i$ and non-preference rating $S_i$ to denote the average helpfulness ratings from $G_u$ and $G_v$, respectively. Let $\mathbf{o}$ and $\mathbf{s}$ be the set of preference ratings ($O_i$s) and non-preference ratings ($S_i$s) respectively, and the mean of $\mathbf{o}$ is $3.8440$ larger than that of $\mathbf{s}$, which is $3.7468$. We also conduct a two-sample t-test on the vectors $\mathbf{o}$ and $\mathbf{s}$. The null hypothesis is $H_0: \mathbf{o} \sim \mathbf{s}$, and the alternative hypothesis is $H_1: \mathbf{o} > \mathbf{s}$. The result show that there is strong evidence to reject the null hypothesis with significance level $\alpha = 0.01$. With the evidence from the means and t-test result, we conclude Observation 3 - raters are likely to consider the reviews from their preference similar users more helpful.

For each review, based on the item similarities between raters and its author, we further divide $G_u$ equally into high preference similar group and low preference similar group. By analyzing these two groups as similar to $G_v$, and $G_u$, we find Observation 4 - the more similar the preferences of raters and an author are, the more helpful raters consider the reviews from the author.

### 4. OUR FRAMEWORK

With social context analysis in the last section, we are ready to introduce our context-aware helpfulness rating prediction framework CAP to address the second challenge - how to model content context and various types of social context for the prediction problem.

#### 4.1 Modeling Context Awareness

We use $H_{ij}$ to denote the estimated helpfulness rating from $u_i$ to $r_j$. Before modeling contextual information, we first introduce our basic model based on probabilistic matrix factorization, and the helpfulness rating $H_{ij}$ can be estimated as,

$$H_{ij} \sim N(H_{ij}, \sigma^2_\beta), \quad H_{ij} = u_i^\top v_j,$$

where $u_i \in \mathbb{R}^K$ and $v_j \in \mathbb{R}^K$ are latent factors of $u_i$ and $r_j$ to capture the preference of $u_i$ and the characteristics of $r_j$ where $K$ is the number of latent factors.

**Modeling Content Context**: Different from the item rating prediction problem, in the review helpfulness rating prediction problem, the texts of reviews provide content context about reviews. We introduce a latent factor $\beta_j$ to exploit the review content of $r_j$ as,

$$\beta_j \sim N(\mathbf{g}^\top \mathbf{x}_j, \sigma_\beta^2),$$

where the latent factor $\beta_j$ has a linear relation with the textual features from review content [14] with coefficients $\mathbf{g}$ and we use similar textual features to these in [15]. Following the common assumption on the loss or error function, we further assume Gaussian error between $\beta_j$ and $\mathbf{g}^\top \mathbf{x}_j$.

**Modeling Individual Context**: Individual context contains author context and rater context. For the helpfulness rating of $u_i$ to $r_j$ from the author $u_i$, we introduce two latent factors $\xi_e$ and $\alpha_i$ to capture author context and rater context, respectively. According to the observations from [15], author context is usually related to the characteristics of authors such as their reputations and statuses. Assume that there is an author feature vector $\mathbf{z}_{ik}$ for $u_i$.

Similar to modeling review content, we consider a linear relationship between author context and rater context and we use similar textual features to these in [15]. Following the common assumption on the loss or error function, we further assume Gaussian error between $\xi_e$ and $\mathbf{g}^\top \mathbf{x}_j$, and then author context can be formulated as,

$$\xi_e \sim N(\mathbf{b}^\top \mathbf{z}_{ik}, \sigma^2_e).$$

A similar strategy can be applied to rater context. Assume that for the rater $u_i$, there is a rater feature vector $\mathbf{y}_j$, and then rater context is formally defined as,

$$\alpha_i \sim N(\mathbf{d}^\top \mathbf{y}_j, \sigma^2_\alpha).$$

**Modeling Relation Context**: Relation context includes connection context and preference context. Different from individual context, relation context only exists when raters and authors have connections or preference relations. Therefore for the helpfulness rating of $u_i$ to $r_j$ from the author $u_i$, we define two indicator functions: (1) $\delta_1$ is defined for preference context where $\delta_1(i, k) = 1$ if $u_i$ is the preference similar user of $u_k$. (2) $\delta_2$ is defined for connection context where $\delta_2(i, k) = 1$ if $u_i$ trusts $u_k$. $\mathbf{p}_{ik}$ and $\mathbf{q}_i$ to model them, respectively. Based on Observation 2, connection context is related to the connection strengths between raters and authors. Therefore for the pair of users $(u_i, u_k)$, we define a connection strength feature vector $\mathbf{q}_i$, and then the latent factor $\gamma_{ik}$ for connection context is formulated as,

$$\gamma_{ik} \sim N(f(\mathbf{h}^\top \mathbf{q}_i, \mathbf{p}_{ik}), \sigma^2_\gamma).$$

where we use an active function $f$ on $\mathbf{h}^\top \mathbf{q}_i$ to ensure that the effect from connection context is positive according to Observation 1 and we find that a sigmoid function works well in this paper. Similarly, based on Observation 3 and Observation 4, the latent factor $\gamma_{ik}$ is formally defined as,

$$\gamma_{ik} \sim N(f(\mathbf{r}^\top \mathbf{p}_i, \mathbf{p}_{ik}), \sigma^2_\gamma).$$

where $\mathbf{p}_i$ is a preference similar feature vector for the pair of users $(u_i, u_k)$.

We use similar author feature vector $\mathbf{z}_{ik}$ as in [15], and the definitions of the rater features for $\mathbf{y}_j$, the preference similar features for $\mathbf{p}_i$, and the connection strength features for $\mathbf{q}_i$ can be found in Appendix. Exploiting content context and various types of social context, our context-aware helpfulness rating prediction framework CAP will estimate the
helpfulness rating $H_{ij}$ as

$$H_{ij} \sim \mathcal{N}(H_{ij}, \sigma_H^2),$$

$$H_{ij} = u_i^T v_j + \beta + \alpha_i + \xi_i + \delta_1(i, k)\gamma_i^k + \delta_2(i, k)\lambda^k_i$$

where we also relate the latent factors of $u_i$ and $v_j$ to observed features of $u_i$ and $r_j$ with coefficients $W$ and $V$ respectively, and $\mathcal{MVN}$ denotes multivariate normal distribution.

Predicting Helpfulness Rating: After learning the latent factors $\Omega$ and prior parameters $\Theta$ from known ratings $O$, an unknown rating of $u_i$ to $r_j$ from $u_i$, $r_j$ can be predicted as,

$$\bar{H}_{ij} = u_i^T v_j + \beta + \alpha_i + \xi_i + \delta_1(i, k)\gamma_i^k + \delta_2(i, k)\lambda^k_i,$$

where $\xi_i \sim \mathcal{N}(0, \sigma_{\xi_i}^2)$, $\gamma_i^k \sim \mathcal{N}(f(r^T p^k_i), \sigma_{\gamma_i}^2)$, $\lambda^k_i \sim \mathcal{N}(f(h^T q_i^k), \sigma_{\lambda_i}^2)$, $u_i \sim \mathcal{MVN}(W_y, A_u)$, $v_j \sim \mathcal{MVN}(V_y, A_v)$.

4.2 Learning Parameters

In this paper, we adopt Monte Carlo EM algorithm [3] to learn latent factors and prior parameters $\Theta$ for CAP from the data automatically. The Monte Carlo EM algorithm maximizes the marginal log-likelihood by iterating through expectation (E) and maximization (M) steps until convergence. Let $\hat{\Theta}$ denote the $t$-th estimate of the set of prior parameters $\Theta$ at the $t$-th iteration.

E-step: We take the expectation of the complete data log likelihood with respect to the posterior of latent factors $\Omega$ conditional on the observation data $O$ as,

$$g_t(\Theta) = E_{\Omega \sim \mathcal{P}(\Omega | \Theta^t)}[L(\Omega, \Theta)],$$

where $L(\Omega, \Theta)$ is the data log-likelihood and the expectation is taken over the posterior distribution $P(\Omega | O, \Theta^t)$. The E-step in our model is not in close form due to the multiplicative terms $u_i^T v_j$, thus approximated by Monte Carlo mean.

We use a Gibbs sampler to draw $\ell$ samples of the latent factors and compute the Monte Carlo means and variances of latent factors.

Computing Monte Carlo mean and variance of $\alpha_i$: Now considering that all the other factors are given, we use Rest to denote all other factors except $\alpha_i$. Assume that $L(\alpha_i)$ is the function including terms involving $\alpha_i$ in $L(\Omega, \Theta)$. Let $L'(\alpha_i)$ and $L''(\alpha_i)$ denote the first-order and second-order derivatives of $L(\alpha_i)$, respectively. Suppose that $\hat{\alpha}_i$ is the minimizer of $L(\alpha_i)$ satisfying the equation $L'(\hat{\alpha}_i) = 0$. Then approximations of the mean and variance of $\alpha_i$ are $\hat{\alpha}_i$ and $L''(\hat{\alpha}_i)|_{\hat{\alpha}_i}$ respectively [3]. We first take the derivation of $L(\Omega, \Theta) w.r.t. \alpha_i$ as,

$$\frac{\partial L}{\partial \alpha_i} = \left( \sum_{j \in (\Omega \setminus \alpha_i)} \frac{\alpha_i}{\sigma_{\alpha_i}^2} + \frac{d^T y_j}{\sigma_{\gamma_j}^2} - \left( \sum_{j \in (\Omega \setminus \alpha_i)} \frac{1}{\sigma_{\alpha_i}^2} + \frac{1}{\sigma_{\gamma_j}^2} \right) \alpha_i \right),$$

where $\alpha_i = H_{ij} - \beta - \delta_1(i, k)\gamma_i^k - \delta_2(i, k)\lambda_i^k - u_i^T v_j - \xi_i$ and $I(\hat{\alpha})$ is the set of helpfulness ratings from $u_i$. Then given Rest, the Monte Carlo mean and variance of $\alpha_i$ are,

$$V[\alpha_i | \text{Rest}] = \left( \sum_{j \in (\Omega \setminus \alpha_i)} \frac{1}{\sigma_{\alpha_i}^2} + \frac{d^T y_j}{\sigma_{\gamma_j}^2} \right)^{-1} \left( \sum_{j \in (\Omega \setminus \alpha_i)} \frac{\alpha_i}{\sigma_{\alpha_i}^2} + \frac{d^T y_j}{\sigma_{\gamma_j}^2} \right).$$

Similar to $\alpha_i$, we can compute the Monte Carlo means and variances of $\beta_j$, $\xi_k$, $u_i$, $v_j$, $\gamma_i^k$ and $\lambda_i^k$ and we ignore the details to save space.

M-step: We maximize the expected complete log likelihood from the E-step to update $\Theta$ as,

$$\hat{\Theta}^{t+1} = \arg \max_{\Theta} g_t(\Theta),$$

where we consider $A_u = \sigma_u^2 I$ and $A_v = \sigma_v^2 I$.

Estimating $\sigma_{\alpha_i}^2$, $d$: Setting zeros to the derivatives of $E_{\Omega \sim \mathcal{P}(\Omega | \hat{\Theta}^t)}[L(\Omega, \Theta)]$ with respective to $\sigma_{\alpha_i}^2$ and $d$, we obtain,

$$\sigma_{\alpha_i}^2 = \sum_{j \in (\Omega \setminus \alpha_i)} (E[\alpha_i] - d^T y_j)^2 + V[\alpha_i],$$

$$d = (n + Y^T Y)^{-1} Y n,$$

where $Y = [y_1, \ldots, y_n]$, $n = |E[\alpha_1], \ldots, E[\alpha_n]|$.

Estimating the other prior parameters $\sigma_{\beta_j}^2, \sigma_{\xi_k}^2, \sigma_{\gamma_i}^2, \sigma_{\lambda_i}^2, \sigma_{W}^2, \sigma_{V}^2$ for $\beta_j$, $\xi_k$, $\gamma_i^k$, $\lambda_i^k$, $W$, and $V$ for $\Theta$ and $\Theta^t$, respectively. Similarly, we can obtain other prior parameters $\sigma_{h_i}^2, \sigma_{h_i}^2, \sigma_{h_i}^2, \sigma_{W}^2, \sigma_{V}^2$ for $h_i$, $\sigma_{W}^2$, and $\sigma_{V}^2$.

5. EXPERIMENTS

In this section, we conduct experiments to answer the following questions (1) Does context awareness help to improve the performance of helpfulness rating prediction as expected? (2) If it does, is it necessary to exploit every type of contextual information? To answer the first question, we
5.1 Experimental Setting

We first rank all helpfulness ratings according to the time points when they are published in chronological order, and then equally split the whole data set into two parts - 50% of them as the training set and 50% of them as the testing set. For the testing set, we further divide it into two parts: (1) cold-start - including helpfulness ratings where the raters or authors are newly added users or the reviews are newly written reviews; and (2) warm-start - containing ratings where the raters and the reviews exist in the training set. We examine the data and find that cold-start includes 44.72% of the testing set. A large proportion of review helpfulness ratings are cold-start ratings, further demonstrating the significance of finding a unique framework for the problem of review helpfulness rating prediction with both cold-start and warm-start settings. A common metric Root Mean Squared Error is used to evaluate performance.

5.2 Comparison of Different Predictors

We compare the proposed framework CAP to various baseline methods, which can be grouped into three categories.

Methods in the first category are totally based on simple statistics from the training set and they are: ST:Mean - predicting the helpfulness of a review as the mean of the helpfulness ratings in the training set; ST:Review - predicting the helpfulness of a review as the average helpfulness rating the review received in the training set; ST:Author - predicting the helpfulness of a review from an author as the average helpfulness rating of the reviews from the author in the training set and ST:Rater - predicting the helpfulness rating from a specific rater as the average rating given by the rater in the training set.

The second category covers the state-of-the-art predictors in the item rating prediction problem and they are: IRP:MF - performing a low-rank matrix factorization on helpfulness rating matrix $H$ [10, 26]; IRP:Neighbor - predicting the helpfulness rating of a review from a rater as the average weighted rating of the review from the rater’s trust terms context; and IRP:MF+Neighbor - this predictor exploits both rating and social information [17]. Note that we do not compare our framework CAP with methods based on tensor factorization [8, 18]: (1) the focus of this paper is to investigate whether exploiting context-aware information can improve the prediction performance and we can choose tensor factorization based methods instead of matrix factorization based methods as our basic models; and (2) the high time and space complexities of tensor factorization limit their applications to large-scale data sets. We also do not compare CAP with feature-based factorization methods [25] in this subsection since they are special cases of CAP, which is discussed in later subsection.

The methods in the third category are review quality predictors and they are: RQP:Text - performing linear regression on textual features from the review content; RQP:Author - performing linear regression on author features, referred as the social features of reviews in [15]; and RQP:Text+Author - performing linear regression on the features with both textual features and author features. The predictors in this category will compute a global helpfulness score for a review.

![Table 2: Performance of Different Predictors in terms of RMSE.](image)

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>warm-start</th>
<th>cold-start</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST:Mean</td>
<td>0.5227</td>
<td>0.7371</td>
</tr>
<tr>
<td>ST:Review</td>
<td>0.4335</td>
<td>0.7158</td>
</tr>
<tr>
<td>ST:Author</td>
<td>0.4603</td>
<td>0.7099</td>
</tr>
<tr>
<td>ST:Rater</td>
<td>0.4969</td>
<td>0.7144</td>
</tr>
<tr>
<td>IRP:MF</td>
<td>0.3616</td>
<td>N.A.</td>
</tr>
<tr>
<td>IRP:Neighbor</td>
<td>0.4102</td>
<td>0.5737</td>
</tr>
<tr>
<td>IRP:MF+Neighbor</td>
<td>0.3270</td>
<td>0.5737</td>
</tr>
<tr>
<td>RQP:Text</td>
<td>0.3758</td>
<td>0.5519</td>
</tr>
<tr>
<td>RQP:Author</td>
<td>0.5442</td>
<td>0.6922</td>
</tr>
<tr>
<td>RQP:Text+Author</td>
<td>0.3584</td>
<td>0.5004</td>
</tr>
<tr>
<td>CAP</td>
<td>0.2342</td>
<td>0.3165</td>
</tr>
</tbody>
</table>

In cold-start, we have similar observations. However, content context plays a more important role in the cold-start problem. The methods considering the review content always outperform those ignoring it. CAP also obtains the best performance, indicating the ability of the proposed framework CAP to solve the cold-start problem.

- Baseline methods based on simple statistics perform well and sometimes their performance is even better than IRP:Author. As reported in Section 2, more than 80% of helpfulness ratings are 3 or 4; therefore, the majority of helpfulness is close to the average helpfulness. ST:Rater obtains better performance than ST:Mean, suggesting that different raters may think of the helpfulness of the same review differently. The quality of reviews from the same author is likely to be consistent [15] and ST:Author obtains the best performance in this category.

- A combination of IRP:MF and IRP:Neighbor obtains the best performance in that category, indicating that users’ historical helpfulness ratings and ratings from their neighborhood are complementary to each other.

- RQP:Text+Author performs the best among review quality predictors, suggesting the importance of the effect from author context. However only considering the effect from author context is not enough to be a good predictor and sometimes the performance is even worse than those based on simple statistics.

- CAP always outperforms all baseline methods. Compared to the best performance of baseline methods, CAP obtains 28.38% relative improvement in terms of RMSE. There are two major reasons: (1) CAP incorporates observed features of review content, raters, authors and their relations; and (2) CAP considers the effects of content context and various types of social context. We will investigate the contributors to the improvement in the next subsection.
5.3 Impact of Different Components

Empirical study shows that CAP significantly helps improve the helpfulness rating prediction performance by exploiting context awareness. In this subsection, we investigate the effect of content context and various types of social context on CAP. Except the basic model from the item rating prediction problem \((\mathbf{u}_i, \mathbf{v}_j)\), our proposed framework CAP exploits content context \((\beta_j)\) and four types of social context \((\alpha_i, \xi_j, \gamma_i, \lambda_j)\). To answer the second question, we eliminate the effects of content context and four types of social context systematically from the proposed framework CAP by defining the following variants of CAP: (1) \(CAP_{\text{Author}}\) - eliminating the effect of author context from CAP; (2) \(CAP_{\text{Rater}}\) - eliminating the effect of rater context from CAP; (3) \(CAP_{\text{Connection}}\) - eliminating the effect of connection context from CAP; (4) \(CAP_{\text{Preference}}\) - eliminating the effect of preference context from CAP; (5) \(CAP_{\text{ARCP}}\) - eliminating the effects of four types of social context from CAP; (6) \(CAP_{\text{Text}}\) - eliminating the effect of content context of reviews from CAP; and (7) \(CAP_{\text{Text}, \text{ARCP}}\) - eliminating the effects from content context and four types of context information from CAP. This variant is a feature-based matrix factorization method [25].

The results are demonstrated in Figures 3(a) and 3(b) in warm-start and cold-start, respectively. When we eliminate each type of social context from CAP, the performance degrades. Compared to CAP, on average, the performance of \(CAP_{\text{Author}}, CAP_{\text{Rater}}, CAP_{\text{Connection}}\) and \(CAP_{\text{Preference}}\) relatively reduces 14.86% and 7.22% in terms of RMSE in warm-start and cold-start respectively, indicating that social context information can help improve the performance. When eliminating four types of social context simultaneously, the performance of \(CAP_{\text{ARCP}}\) is worse than that of variants which eliminate one type of social context, indicating that the four types of social context contain complementary information to each other. When we eliminate the effect of content context, the performance reduces more than 20% in both warm-start and cold-start data sets and the content of reviews plays an important role in the prediction process, especially for the cold-start problem. \(CAP_{\text{Text}}\) \(ARCP\) performs worst, suggesting the importance of content context and various types of social context information for CAP. However, \(CAP_{\text{Text}, \text{ARCP}}\) obtains performance improvement over IRP:MF. Both of them consider the interactions between the latent factors of raters and reviews, while \(CAP_{\text{Text}, \text{ARCP}}\) also incorporates observed features of raters and reviews, suggesting that observed features are very important and can help improve prediction performance. Besides the performance improvement, another advantage of \(CAP_{\text{Text}, \text{ARCP}}\) over IRP:MF is that it enables to solve the cold-start problem in the review helpfulness rating prediction problem.

With these observations, we can answer the second question that the contributors to the performance improvement in CAP include: (1) considering content context, (2) exploiting various types of social context, and (3) incorporating observed features of raters and reviews.
Latent factor models consider the interactions between latent features of users and items by projecting them to the same latent factor space. Matrix factorization methods are very competitive methods in this category [10, 7, 17]. They assume that a few latent patterns influence user rating behaviors and perform a low-rank matrix factorization on the user-item rating matrix. Recently, high-dimensional tensor factorization based methods were proposed to allow adding additional dimensions to the user-item matrix [8, 18]. However, the limitations of the high-dimensional tensor factorization based methods are their high time and space complexities.

7. CONCLUSION

In this paper we study the problem of review helpfulness rating prediction by exploiting context awareness to infer unknown helpfulness ratings automatically. We extract four types of social context, i.e., author context, rater context, connection context and preference context, formulate them mathematically, and propose a context-aware helpfulness prediction framework CAP which exploits context content and various types of social context. Experimental results demonstrate that our proposed framework outperforms the state-of-the-art baseline methods with both cold-start and warm-start settings, and further experiments are conducted to understand the importance of context awareness in the proposed framework.

There are several directions needing further investigation. First, we will study how to model context awareness in the tensor factorization based models. Second, user preferences are likely to change over time and it is interesting to exploit temporal effects in CAP. Finally, users may need helpful reviews associating with the recommended items to help them make decisions. Recommending items and the helpful reviews simultaneously will be a promising direction.

Acknowledgments

We thank the useful comments from Atish Das Sarma and anonymous reviewers. This work is, in part, supported by ARO (#025071) and NSF (#IIS-1217466).

8. REFERENCES


APPENDIX

Rater Features: # of trustors, # of trustees, pagerank score, average item rating, average item rating from the social network, average item rating from similar users, average helpfulness rating, average helpfulness rating from the trust network, average helpfulness rating from similar users

Preference Similar Features: # of commonly rated items, Jaccard’s coefficient on rated items, Cosine similarity of item ratings, Pearson similarity of item ratings, difference of average item rating scores, difference of minimal item rating scores, difference of maximal item rating scores

Connection Strength Features: Jaccard’s coefficient on common out-degrees, Jaccard’s coefficient on common in-degrees, Adamic/Adar score on common out-degrees, Adamic/Adar score on in-degrees, Katz score.