2008 Eighth IEEE International Conference on Data Mining

Modeling and Predicting the Helpfulness of Online Reviews

Yang Liu[1], Xiangji Huang[2], Aijun An[1] and Xiaohui Yu[2]
[1]Department of Computer Science and Engineering, York University, Toronto, Canada
{yliu, aan}@cse.yorku.ca
[2]School of Information Technology, York University, Toronto, Canada
{jhuang, xhyu}@yorku.ca

Abstract

Online reviews provide a valuable resource for potential customers to make purchase decisions. However, the sheer volume of available reviews as well as the large variations in the review quality present a big impediment to the effective use of the reviews, as the most helpful reviews may be buried in the large amount of low quality reviews. The goal of this paper is to develop models and algorithms for predicting the helpfulness of reviews, which provides the basis for discovering the most helpful reviews for given products. We first show that the helpfulness of a review depends on three important factors: the reviewer's expertise, the writing style of the review, and the timeliness of the review. Based on the analysis of those factors, we present a nonlinear regression model for helpfulness prediction. Our empirical study on the IMDB movie reviews dataset demonstrates that the proposed approach is highly effective.

1 Introduction

The increasing impact of the Internet has dramatically changed the way that people shop for goods. More and more people are now gravitating to reading products reviews prior to making purchasing decisions. Such reviews have become an indispensable component of e-commerce Websites such as Amazon (http://www.amzon.com), and they are also available through dedicated Websites such as CNET (http://www.cnet.com) and IMDB (http://www.imdb.com). While reading reviews can help the potential customers make informed decisions, in many cases the large quantity of reviews available for a product can be overwhelming and actually impede the customers' ability to evaluate the product. This is further aggravated by the fact that the quality of the online reviews tends to be very uneven, ranging from excellent detailed opinions to simple repetition of product specifications to (in the worst case) pure spams. As a consequence, potential consumers have to sift through a large number of reviews in order to form an unbiased judgment regarding the product.

To alleviate this problem, many Websites are now allowing readers of a review to indicate whether they think that review is helpful by voting for or against it, and a tally (or score) is provided in the form of "*100 out of 150 people found the following review helpful*". The reviews can be sorted according to their helpfulness using those scores. Although this is certainly an improvement in the right direction, there are still important issues to be addressed. For example,

- For newly posted reviews, most likely no vote or only a few votes have been cast, and therefore, identifying their helpfulness is difficult.
- Presenting the reviews ranked by their user-voted helpfulness scores may create situations of "monopoly" in that only the highest ranked reviews get viewed, leaving no opportunities for the newly published yet unvoted reviews to show up on users' radar.
- In some cases, reviews can be incorrectly labeled as helpful or not helpful due to spam voting [8].

In these scenarios, it will be highly desirable to have a way to predict the helpfulness of the given reviews. The predicted helpfulness scores can then be used to address the above problems either directly or indirectly, by combining with existing user votes (if there is any).

This paper is concerned with the problem of automatically evaluating the *helpfulness* of reviews and consequently identifying the most helpful reviews for a particular product. Previous research on review mining has focused on answering questions like "*What do people think of the product?*" [3, 16, 18], "*How would users' evaluation affect the sales of a certain product?*" [1, 5, 11], and "*How to understand and summarize the reviews with minimum human efforts?*" [7, 20], but few explicitly consider the problem of evaluating the quality of reviews, which is significantly

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.94

different from the well-studied problem of sentiment classification and opinion extraction.

In this paper, we take a principled approach to tackling this important problem by developing a novel model for predicting helpfulness of reviews. The model is based on a thorough analysis of some major factors that may affect the helpfulness of a review, including the areas of expertise of the reviewer, the writing styles, the timeliness of the reviews, the length of the reviews, etc. We provide a detailed analysis of those factors and explain their effects on the helpfulness of reviews. We then develop a non-linear regression model that takes all important factors into consideration, serving as a basis for helpfulness prediction. Extensive experiments were conducted on the IMDB dataset, demonstrating the effectiveness of the proposed approach.

To make our discussions and results more concrete, in this paper we use movie reviews in the past two years (2006-2007) collected from the IMDB Website as a case study. However, our approach is general enough to be easily adapted to handling other types of online reviews.

To summarize, we make the following contributions in this paper.

- We carefully analyze the possible factors that might affect the helpfulness of reviews, and identify three most influential ones, namely, reviewer expertise, writing style, and timeliness.
- We develop a mathematical model that is able to capture all of the three important factors for helpfulness prediction.
- We conduct extensive experiments on a movie dataset to verify the effectiveness of our approach.

The rest of the paper is organized as follows. In Section 2, we define the prediction problem, and provide a detailed analysis of the major factors that affect the helpfulness of reviews. In Section 3, we propose a regression model based on radial basis functions. Experimental results are presented in Section 4. Section 5 provides a review of related work. Section 6 concludes this paper and discusses directions for future work.

2 Problem Definition and Observations

In this section, we first formally define the problem of helpfulness prediction, and then analyze the factors that may affect the helpfulness of a review, which will provide the basis for the proposal of the model in the next section.

2.1 Problem definition

The goal of this research is to develop a model that can accurately predict the helpfulness of a review. For a given review, its "helpfulness" H is defined as the expected fraction of people who will find the review helpful. That is, H is a number falling in the range $[0, 1]$, and greater values of H imply higher helpfulness.

As in any prediction tasks, the prediction model will be obtained based on available training data, which consist of reviews and related product information. Let the set of reviewers (authors of the reviews) concerned be $\mathcal{U}$, the set of movies be $\mathcal{M}$, the set of reviews be $\mathcal{D}$, then each review can be represented as a quadruple $R = (u, d, m, t)$, where $u \in \mathcal{U}$ denotes the reviewer, $d \in \mathcal{D}$ represents the review, $m \in \mathcal{M}$ represents the movie for which the review is written, and t indicates the number of days elapsed from the movie release to the time the review is published. For each movie in $\mathcal{M}$, assume that the genres it falls in are also available.

The helpfulness of a review in the training data can be approximated by the tally attached to that review, which takes the form of "*x out of y people found the following review helpful*". That is, $H = \frac{x}{y}$. As an effective indicator of the public opinions, this evaluation metric has also been widely adopted in previous product review helpfulness studies [19]. To maintain the robustness of the prediction model, in this study, we only consider reviews with at least 10 votes, i.e., $y \geq 10$.

2.2 Observations

In order to develop an effective model for helpfulness prediction, we must carefully analyze the important factors that may affect a review's helpfulness rating. To this end, we have examined the reviews on several popular Websites, including CNET, Amazon, and IMDB, and conducted preliminary experiments to evaluate the various factors involved. Our efforts reveal that the following are among the most important factors.

1. **Reviewer Expertise**: Product reviews often involve personal experience, thoughts, and concerns. Also, it is common that different reviewers demonstrate expertise on different types of products. For example, reviewers fond of science fictions are likely to be familiar with and produce good reviews on sci-fi movies like *Star Wars* and *The Matrix*, but may be less proficient in writing reviews for *American Zeitgeist* and *An Inconvenient Truth*, which fall into the category of documentaries. Those preferences and expertise might be well reflected through reviews they compose, which we must take into consideration when building the prediction model.
2. **Writing Style**: Due to the large variation of the reviewers' background and language skills, the online reviews are of dramatically different qualities. Some

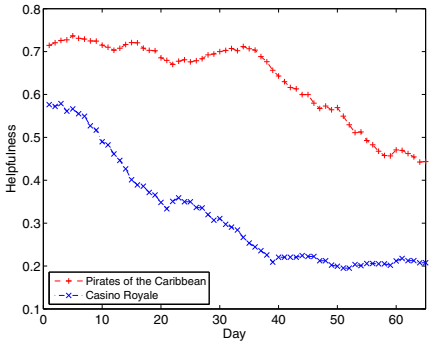


Figure 1. An example of review helpfulness vs. time of review.

reviews are highly readable and therefore tend to be more helpful, whereas some reviews are either lengthy but with few sentences containing author's opinions, or snappy but filled with insulting remarks. Simply ignoring such differences in readability and style may produce misleading estimates of the review quality. A proper representation of such difference must be identified and factored into the prediction model.

3. **Timeliness**: In addition to the available review content, most online reviews are also associated with a particular time stamp, which indicates when the review is posted. In general, the helpfulness of a review may significantly depend on when it is published. For instance, research shows that a quarter of a motion picture's total revenue comes from the first two weeks [4], which means a timely review might be especially valuable for users seeking opinions about the movie. As a concrete example, Figure 1 shows the average helpfulness of reviews versus the time the reviews are published (number of days since the release) for two movies, *Pirates of the Caribbean (Dead Man's Chest)* and *Casino Royal*. The average helpfulness numbers presented here are 14-day moving averages in order to smooth-out the short-term irregularities and show the overall trend. It is evident that the general trend is that the average helpfulness of movie reviews declines as time passes by. In addition, some previous studies made similar observations that product reviews written early tend to get more user attention on e-commerce websites, such as Amazon [8]. This further confirms our hypothesis that timely reviews are usually more helpful.

We have also considered other possible factors that may affect the helpfulness values, e.g., length of the review, polarity of the review, the average rating of all reviews on the movie, etc. However, none of them shows clear correlation with the value of helpfulness, and the detailed examination is available elsewhere [12]. Other factors, such as server-side weblogs indicating how many users read but did not respond to the helpfulness question, might also facilitate the prediction. However, they are not considered in our study due to the data availability issue.

3 Predicting Helpfulness

Based on the observations from the previous section, we propose a model that accounts for these three important factors. Once trained, this model can be used for predicting the helpfulness of a given review. In the following discussions, we will use the IMDB movie data as a case study, although the model can be easily applied to other types of review data.

Since radial basis functions (RBF) are used in the modeling of both the expertise factor and the writing style factor (in the next subsection), a brief introduction of RBF is in order. After that, we will analyze how to model each factor mentioned in the previous section, and then present the non-linear regression model with all those factors incorporated, followed by a description of the training algorithm.

3.1 Radial basis functions

Function approximation is an important component to solving prediction problems defined over both continuous and discrete spaces. A powerful function approximator will not only accurately represent a value for a state it has experienced, but also generalize values to nearby states it has not experienced before. The most common type of approximator is the linear approximator. It has the benefit of being straightforward and involving lower computational cost, but it is obviously unreliable if the true relation between the inputs and the output is nonlinear. One then has to rely on non-linear approximators, such as RBF.

Radial basis functions have the advantage of being much simpler than other popular function approximators, such as multilayer perceptron neural networks, but still serving as a universal function approximator. They are generally used when local properties of the functional relationship needs to be captured, as is the case in the modeling of reviewer expertise and writing style. Due to its high flexibility, radial basis functions have been widely used in many areas, including finance and image processing [2].

A radial basis function is a real-valued function whose value depends only on the distance of the input vector $\mathbf{x}$ from some *center* point $\boldsymbol{\mu}$. In the most general form, the RBF $\phi(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = f\left((\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$, where f is the function used (Gaussian, Cauchy, etc.) and $\boldsymbol{\Sigma}$ is the metric. The term $(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$ represents the distance between the input $\mathbf{x}$ and the center $\boldsymbol{\mu}$ in the metric defined

by $\mathbf{\Sigma}$. Here, we choose the distance metric to be Euclidean. In this case, $\mathbf{\Sigma} = \sigma^2\mathbf{I}$ for some scalar radius σ. Hence,

$$\phi(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Sigma}) = f\left(\frac{(\mathbf{x}-\boldsymbol{\mu})^T(\mathbf{x}-\boldsymbol{\mu})}{\sigma^2}\right). \quad (1)$$

The function f can take various forms. In this study, we choose the commonly used Gaussian RBF: $f(y) = e^{-y}$, and $y = -\frac{(\mathbf{x}-\boldsymbol{\mu})^T(\mathbf{x}-\boldsymbol{\mu})}{\sigma^2}$, where σ is also called the spread of the RBF. Intuitively, the further away $\mathbf{x}$ is from the center $\boldsymbol{\mu}$, the smaller the function value is, and the function peaks at the center when $\mathbf{x} = \boldsymbol{\mu}$. In addition, the value of the spread σ determines the "tightness" of the RBF, i.e., how fast the function value falls off when the input $\mathbf{x}$ gets further away from the center.

Multiple RBFs can be combined to build up function approximations of the form

$$g(\mathbf{x}) = \sum_{i=1}^{k} a_i \phi(\mathbf{x}|\boldsymbol{\mu}_i, \mathbf{\Sigma}_i), \quad (2)$$

where the approximation function $g(\mathbf{x})$ is represented as a weighted sum of k radial basis functions, each with a different center $\boldsymbol{\mu}_i$, a metric $\mathbf{\Sigma}_i$, and a weight a_i. Such function approximation models are sometimes referred to in the literature as radial basis function networks. Figure 2 shows an example of using 3 radial basis functions to approximate a function. In this example, one would like to fit a function to the scattered data points. Although in our model for helpfulness prediction, we will deal with multi-dimensional input, for illustration purpose, this example deals with one-dimensional input. The fitted function represented by the solid line can be obtained by taking the weighted sum of the three individual RBFs. Fitting the data with the function involves determining the centers and spreads of the RBFs as well as the weight of each RBF.

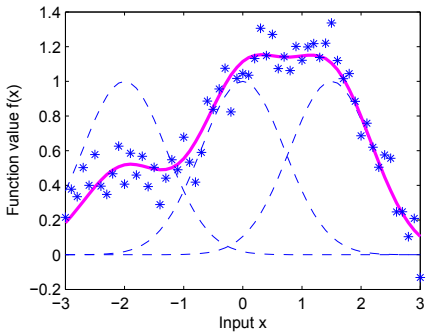


Figure 2. Using radial basis functions for function approximation. The solid line represents the fitted function, and the three RBFs are plotted with dotted lines.

3.2 Modeling expertise

As discussed in Section 2, the helpfulness of a review depends in part on the level of expertise of the reviewer on the product (movie) concerned. For example, for a given reviewer, if his past reviews on a certain set of movies (denoted by $\mathcal{A}$) are rated very high while his reviews on some other movies (denoted by $\mathcal{B}$) are very low, then we have reasons to expect that a new review by this reviewer will be considered more helpful if the movie concerned is more similar to the movies in $\mathcal{A}$ than to those in $\mathcal{B}$.

In order to quantify the "similarity", we first need to choose the right features to represent each movie. To this end, we use the genres provided by IMDB to represent each movie. As an example, the movie *Casino Royale* is labeled by IMDB as "Action", "Adventure", and "Thriller", which can be used to represent the movie for our purpose. Formally, each movie is represented by a m-dimensional vector $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, where m is the number of different genres available for all movies. Each dimension corresponds to one genre, and $x_i (1 \leq i \leq m)$ takes the value of $\frac{1}{l}$, if the movie belongs to the corresponding genre (where l is the number of genres the movie falls into), and 0 otherwise. Note that due to the normalization factor l, $\sum_{i=1}^{m} x_i = 1$.

The next step is to measure the similarity of a given movie to movies that have been reviewed by the same reviewer, and relate this measure to the helpfulness score. We choose to approximate the relationship using RBFs. If we were to predict the helpfulness of a review based solely on the reviewer expertise factor, then we would fit the following regression model on the training data.

$$\hat{H}_1 = \sum_{i=1}^{k_1} u_i \phi(\mathbf{x}|\boldsymbol{\mu}_i, \sigma_i), \quad (3)$$

where $\hat{H}_1$ is the estimated helpfulness score, $\mathbf{x}$ is the feature vector representing the movie, k_1 is the number of centers in the RBF network, $\boldsymbol{\mu}_i$ and σ_i are the center and spread of the i-th RBF respectively, and u_i is the weight of the i-th RBF.

Since we represent each movie using a feature vector based on its genres, each center can be considered as corresponding to one "cluster" of movies that are similar to each other in terms of their genres. The helpfulness of a given movie is thus the weighted sum of the distance between the movie to those centers. In this way, the reviewer's expertise on different clusters of movies can be naturally captured in that similar movies will have similar distances to the centers and therefore have similar helpfulness scores.

3.3 Modeling writing style

A previous study [19] has shown that the linguistic style can be a very good indicator of the utility of the review. In fact, shallow syntactical features like part-of-speech provide more predicting powers than deeper features at the lexical level. Thus, we choose to label the part-of-speech of the words contained in the reviews with a fixed set of tags using LingPipe[1], a suite of Java libraries for the linguistic analysis of natural language.

For each review, we parse it using the LingPipe tagger, and count the number of words with each tag. Those counts are further normalized by dividing them with the word count of the review. The resulting numbers form a vector, denoted by $\mathbf{y}$, with each number corresponding to one dimension. This vector $\mathbf{y}$ is used as a representation of the review for the purpose of modeling writing styles.

We again use a radial basis function network to model the relationship between the feature vector $\mathbf{y}$ and the helpfulness of the review, with each RBF explaining part of the functional relationship, and the weights indicating the contribution of each RBF. Formally, if we were to predict the helpfulness solely based on the writing style, the regression model we would like to use is

$$\hat{H}_2 = \sum_{i=1}^{k_2} v_i \psi(\mathbf{y}|\boldsymbol{\nu}_i, \xi_i), \tag{4}$$

where $\hat{H}_2$ is the estimated helpfulness, v_i, $\boldsymbol{\nu}_i$, and ξ_i are the weight, center, and the spread of the i-th RBF respectively, and k_2 is the number of RBFs.

Of course, the writing style is only one of the factors affecting the helpfulness of a review. Therefore, the model in Equation 4 will be combined with other factors in the complete model we propose.

3.4 Modeling timeliness

Our analysis in Section 2 has shown that there is a strong correlation between the helpfulness of a review and when it is published. Having observed the trend for a large number of movies, we hypothesize that the helpfulness of a movie review is subject to exponential decay with respect to time. Therefore, we propose the following model for movie reviews if the prediction of helpfulness were to be done only based on the timeliness:

$$\hat{H}_3 = e^{-\beta(t-t_0)+d}, \tag{5}$$

where $\hat{H}_3$ is the estimated helpfulness, t_0 is the release time of the movie, t is the time when the review is published, and β and d are parameters in the model to be estimated. Intuitively, β controls the rate of decay in the helpfulness as we move further away in time from the movie release.

[1] http://alias-i.com/lingpipe/

3.5 The complete model

Now that we have built the regression model for each individual factor, we are ready to propose the complete model that incorporates all of the above factors. The idea is to consider the helpfulness score a weighted sum of the three individual models, as shown below:

$$\hat{H} = p\sum_{i=1}^{k_1} u_i \phi(\mathbf{x}|\boldsymbol{\mu}_i, \sigma_i) + q\sum_{i=1}^{k_2} v_i \psi(\mathbf{y}|\boldsymbol{\nu}_i, \xi_i) + r\cdot e^{-\beta(t-t_0)+d},$$

where p, q, and r are the weights of the three components. Note that the above equation can be further simplified, as the weights p, q, and r can be "absorbed" by the individual components. For example, $p\sum_{i=1}^{k_1} u_i \phi(\mathbf{x}|\boldsymbol{\mu}_i, \sigma_i)$ can be rewritten as $\sum_{i=1}^{k_1} u'_i \phi(\mathbf{x}|\boldsymbol{\mu}_i, \sigma_i)$, where $u'_i = p \cdot u_i$, and $r \cdot e^{-\beta(t-t_0)+d}$ can be rewritten as $w \cdot e^{-\beta(t-t_0)}$, where $w = r \cdot e^d$. Therefore, the model can be written in a more concise form:

$$\hat{H} = \sum_{i=1}^{k_1} u_i \phi(\mathbf{x}|\boldsymbol{\mu}_i, \sigma_i) + \sum_{i=1}^{k_2} v_i \psi(\mathbf{y}|\boldsymbol{\nu}_i, \xi_i) + w\cdot e^{-\beta(t-t_0)} \tag{6}$$

where the notations u_i and v_i are overloaded for the sake of brevity, with them actually referring to u'_i and v'_i as defined above.

The model given in Equation 6 makes it possible to capture all of the factors discussed in this section, with the weights $\{u_i\}_{i=1}^{k_1}$, $\{v_i\}_{i=1}^{k_2}$ and w controlling the "contribution" of each factor to the helpfulness score.

3.6 Parameter estimation

We now develop the algorithm that can be used to estimate the model parameters based on the training data (movie reviews). Assume that the training data consists of N reviews, and for each review j $(1 \le j \le N)$, $\mathbf{x}_j$, $\mathbf{y}_j$, and t_j can be obtained, as well as the true helpfulness score H_j. The set of parameters in the model include

1. the weights $\{u_i\}_{i=1}^{k_1}$, $\{v_i\}_{i=1}^{k_2}$, and w;
2. the centers $\{\boldsymbol{\mu}_i\}_{i=1}^{k_1}$ and $\{\boldsymbol{\nu}_i\}_{i=1}^{k_2}$,
3. the spreads $\{\sigma_i\}_{i=1}^{k_1}$ and $\{\xi_i\}_{i=1}^{k_2}$, and
4. the decay rate β.

The values of k_1 and k_2 are supplied by the user.

The goal of training is to estimate the parameters such that the sum of squared error (SSE) between the true values and the model output values is minimized, i.e., we would like to minimize

$$\varepsilon = \frac{1}{2}\sum_{j=1}^{N} \delta^2, \tag{7}$$

where $\delta = H_j - \hat{H}_j$. The optimization can be done through the method of steepest descent. By computing the partial derivatives of Equation 7, we can apply the following rules to iteratively update the values of the parameters as follows.

Let $\{\eta_u, \eta_v, \eta_w \dots\}$ be the user-defined learning rate for parameters $\{u_i, v_i, w \dots\}$ in the model.

1. For the weights, we have

$$u_i^{new} = u_i^{old} - \eta_u \frac{\partial \varepsilon}{\partial u_i} = u_i^{old} - \eta_u \sum_{j=1}^{N} \delta \phi(\mathbf{x}_j | \boldsymbol{\mu}_i^{old}, \sigma_i^{old}),$$

$$v_i^{new} = v_i^{old} - \eta_v \frac{\partial \varepsilon}{\partial v_i} = v_i^{old} - \eta_v \sum_{j=1}^{N} \delta \psi(\mathbf{y}_j | \boldsymbol{\nu}_i^{old}, \xi_i^{old}),$$

$$w^{new} = w^{old} - \eta_w \frac{\partial \varepsilon}{\partial w} = w^{old} - \eta_w \sum_{j=1}^{N} \delta e^{-\beta^{old}(t_j - t_{0,j})};$$

2. For the centers, we have

$$\begin{aligned}
\boldsymbol{\mu}^{new} &= \boldsymbol{\mu}^{old} - \eta_{\boldsymbol{\mu}} \frac{\partial \varepsilon}{\partial \boldsymbol{\mu}} \\
&= \boldsymbol{\mu}^{old} - 2\eta_{\boldsymbol{\mu}} \boldsymbol{\mu}^{old} \sum_{j=1}^{N} \delta \phi(\mathbf{x}_j | \boldsymbol{\mu}_i^{old}, \sigma_i^{old}) \frac{\mathbf{x}_j - \boldsymbol{\mu}_i^{old}}{(\sigma^{old})^2}, \\
\boldsymbol{\nu}^{new} &= \boldsymbol{\nu}^{old} - \eta_{\boldsymbol{\nu}} \frac{\partial \varepsilon}{\partial \boldsymbol{\nu}} \\
&= \boldsymbol{\nu}^{old} - 2\eta_{\boldsymbol{\nu}} \boldsymbol{\nu}^{old} \sum_{j=1}^{N} \delta \psi(\mathbf{y}_j | \boldsymbol{\nu}_i^{old}, \xi_i^{old}) \frac{\mathbf{y}_j - \boldsymbol{\nu}_i^{old}}{(\xi^{old})^2};
\end{aligned}$$

3. For the spreads, let $\omega = \frac{1}{\sigma^2}$, and $\zeta = \frac{1}{\xi^2}$, and we have

$$\begin{aligned}
\omega^{new} =& \omega^{old} - \eta_\omega \frac{\partial \varepsilon}{\partial \omega} \\
=& \omega^{old} + \eta_\omega u_i^{old} \sum_{j=1}^{N} \delta \phi(\mathbf{x}_j | \boldsymbol{\mu}_i^{old}, \omega_i^{old}) \\
& \cdot (\mathbf{x}_j - \boldsymbol{\mu}_i^{old})^T (\mathbf{x}_j - \boldsymbol{\mu}_i^{old}) \\
\zeta^{new} =& \zeta^{old} - \eta_\zeta \frac{\partial \varepsilon}{\partial \zeta} \\
=& \zeta^{old} + \eta_\zeta v_i^{old} \sum_{j=1}^{N} \delta \psi(\mathbf{y}_j | \boldsymbol{\nu}_i^{old}, \zeta_i^{old}) \\
& \cdot (\mathbf{y}_j - \boldsymbol{\nu}_i^{old})^T (\mathbf{y}_j - \boldsymbol{\nu}_i^{old})
\end{aligned}$$

4. Finally, for the decay rate β, we have

$$\beta^{new} = \beta^{old} - \eta_\beta \frac{\partial \varepsilon}{\partial \beta} = \beta^{old} + \eta_\beta w^{old} \sum_{j=1}^{N} (t_j - t_{0,j}) e^{-\beta(t_j - t_{0,j})}$$

3.7 Prolific vs non-prolific reviewers

Recall that in modeling the reviewer expertise as described in Section 3.2, we rely on the genres of the movies the reviewer has commented and the corresponding helpfulness scores. This requires sufficient past reviews of the reviewer in order to achieve meaningful results. In reality, some reviewers may have written only a few or no reviews, or the reviews a reviewer has written may not be present due to data availability issues. We therefore make the distinction between prolific reviewers and non-prolific reviewers and revise the model correspondingly. We call a reviewer a prolific reviewer if the number of reviews authored by him/her in the data set exceeds a certain threshold T, and non-prolific otherwise. For prolific users, we simply use the model described in Section 3.5, whereas for non-prolific ones, we need to drop the first term regarding reviewer expertise in the model, as we do not have sufficient grounds to make meaningful inference in that regard. In that case, the model becomes

$$\hat{H} = \sum_{i=1}^{k_2} v_i \psi(\mathbf{y} | \boldsymbol{\nu}_i, \xi_i) + w \cdot e^{-\beta(t - t_0)}. \tag{8}$$

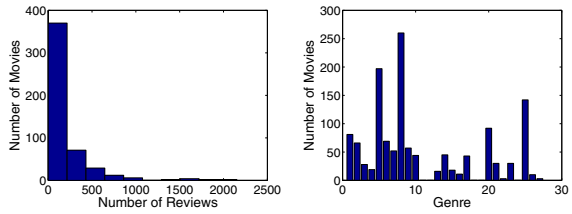
Note that since the above model does not involve any information regarding individual reviewers, a common model can be trained for all of the non-prolific reviewers. The parameter estimation can be done using the update formulae presented in Section 3.6. It is worth pointing out that the distinction between prolific and non-prolific reviewers is due to data availability; we do not assume that the reviews written by prolific reviewers are more helpful than those written by non-prolific reviewers.

4 Empirical Study

We conducted extensive experiments on the IMDB data set to evaluate the effectiveness of the proposed prediction model and study the behavior of the model as we change the user-tunable parameters.

4.1 Experiment settings

The movie review data set was obtained from the publicly accessible IMDB Website. Specifically, we collected the reviews for 504 movies released in the United States during the period from January 6, 2006 to November 21, 2007. We intentionally selected the time that is not very close to the present time in the hope that the voting of helpfulness has stabilized, as less and less reviews are expected to appear as time increases across the whole time span. To model reviewer expertise, we also collected the genre labels for each movie. In total, $94,919$ reviews were collected, and the number of review entries collected for each movie ranges from $2,152$ (for *Superman Returns* [2006]) to 2 (for *Absolute Wilson* [2006]). Those reviews were posted by 56,588 different reviewers. Note that we only collected reviews posted by reviewers from the US as it helps to ensure the consistency in the release time (it is common for a movie to be released on different dates in different countries). The total number of genres involved are 27.



(a) Distribution of the number of reviews

(b) Distribution of the number of movies per genre

Figure 3. The distribution properties of the data

Figure 3(a) and (b) show the distributions of the number of reviews available for movies, and the number of movies per genre respectively. To ensure the robustness of the prediction model, we only use the reviews with at least 10 votes. Also, for the purpose of training and testing, only the reviews with a helpfulness score available (i.e., reviews with a label of the form "*x out of y people found this review helpful*") are used. The number of such movie reviews is 22,819. The movie information (genres for each movie) and the review data are indexed using Apache Lucene[2]. For each review, its feature vectors are obtained as described in Section 3, and we use 10-fold cross validation to evaluate our approach.

4.2 Evaluation

We evaluate the effectiveness of the proposed model using two metrics as we anticipate that the model will be used in different ways. First, the model can be used to predict the helpfulness of reviews directly, so we would like to measure the deviation of the predicted value from the true value. We call this a prediction problem. Second, the model can be also used to help retrieve only those reviews that are considered helpful, i.e., the reviews having a predicted helpfulness higher than a certain threshold. We call this a classification (or retrieval) problem.

Two metrics, which were used in previous literatures [19, 5], are adopted to evaluate the predication accuracy in those two scenarios respectively, namely, the *Mean Squared Error* (MSE) (for the prediction problem) and the *F-measure* (for the classification problem). Specifically, for each review in the test set, we make a prediction for its helpfulness and compute the squared deviation between the predicated value and the true helpfulness. MSE is defined as the sum of all the deviations divided by the total number of predictions. That is,

[2]http://lucene.apache.org

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(H_i - \hat{H}_i)^2, \quad (9)$$

where n is the number of reviews in the test set. Note that lower MSE values indicate higher prediction accuracy. To measure the performance using F-measure, we consider a review as helpful if its helpfulness score is greater than a given threshold θ. In our experiments, we set $\theta = 0.5$.

4.3 Parameter selections

In the prediction model, there are several user-chosen parameters that provide the flexibility to fine tune the model for optimal performance. They include the threshold T to separate prolific users and non-prolific users, the number of RBFs in the RBF network k_1 and k_2, and a threshold θ determining whether a review is helpful. We now study how the choice of these parameter values affects the prediction accuracy.

4.3.1 Effect of T

Recall that in Section 3, we use a threshold T to distinguish a prolific reviewer from a non-prolific reviewer, based on how many reviews in the data are authored by that reviewer. We train different models for the two types of reviewers as discussed in Section 3.

With fixed values of k_1 and k_2 ($k_1 = 3$, and $k_2 = 10$), we vary T, and observe the changes in accuracy. Similar trends can be observed for other values of k_1 and k_2. As shown in Table 1, as T increases from 10 to 30, the prediction performance improves in both F-measure (for the classification problem) and MSE (for the prediction problem) , and at $T = 30$, it achieves the best accuracy with F-measure=0.7116 and MSE=0.0332. This implies that accumulating more reviews for a given author allows our model to better capture the effects that influence the helpfulness, which leads to more accurate predictions. In addition, the accuracy for prolific reviewers is much superior to that for non-prolific reviewers across different values of T, indicating the effectiveness of the "reviewer expertise" factor in the model for prolific reviewers.

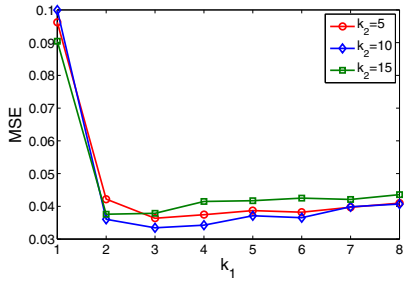
4.3.2 Effects of k_1 and k_2

We then vary the values of k_1 and k_2, with T fixed at 30, to study how the number of RBFs affects the prediction accuracy on prolific reviewers. We do not consider non-prolific reviewers in this experiment as the features describing reviewer expertise are not available for non-prolific reviewers, and thus k_1 is not required for the corresponding model. The effect of k_2 is similar on the two types of reviewers, and therefore only the results on the prolific reviewers are presented here.

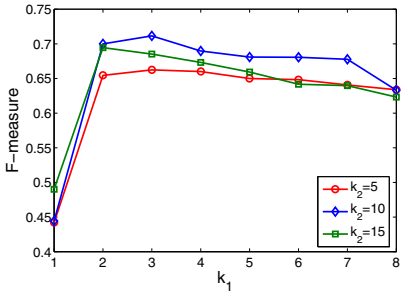
T		MSE	F-measure	# of Reviews	# of Reviewers
10	P	0.0486	0.6717	2378	109
	N	0.0768	0.4307	20441	17266
15	P	0.0392	0.6748	1814	65
	N	0.0632	0.4307	21005	17310
20	P	0.0386	0.6886	1258	33
	N	0.0668	0.4364	21561	17342
25	P	0.0354	0.6989	1079	25
	N	0.0661	0.4363	21740	17350
30	P	**0.0332**	**0.7116**	912	19
	N	0.0658	0.4365	21907	17356

Table 1. Effect of T. N and P refer to non-prolific and prolific reviewers respectively.

We first vary the value of k_1, and observe from Figure 4(a) and (b) that there is a large improvement in accuracy when k_1 increases from 1 to 2, and the model achieves its best performance with $k_1 = 3$. This implies that introducing multiple components to analyze the reviewer expertise can greatly improve the prediction accurary. However, after k_1 past a thresold, the accuracy tends to decrease. This might be due to over-fitting the training data with more RBFs. Nonetheless, the accuracy remains stable for a wide range of k_1 values, indicating the insensitivity of the model with respect to the choice of k_1 values. It is also worth noting that the trend in accuracy remains the same regardless of the choice of k_2.



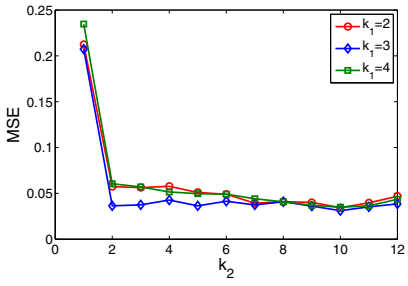
(a) MSE



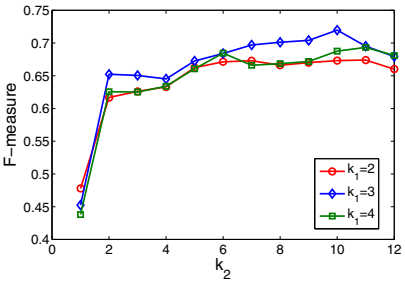
(b) F-measure

Figure 4. Effect of k_1 on prolific reviewers

Similarly, we fix the values of T and k_1, and vary k_2 from 1 to 12. As shown in Figure 5 (a) and (b), there is also an optimal choice of k_2, which is 10. Similar to the case of



(a) MSE



(b) F-measure

Figure 5. Effect of k_2 on prolific reviewers

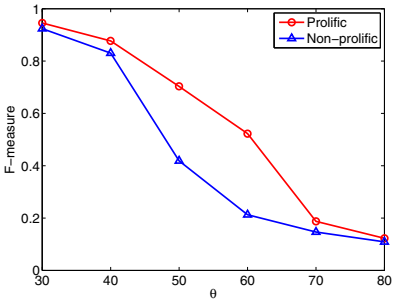


Figure 6. Effect of θ

k_1, the accuracy remains quite stable over a wide range of k_2, which again demonstrates that the model is not sensitive to the choice of parameter values.

4.3.3 Effect of θ

In classifying a review as helpful or not helpful, we use a threshold θ. Figure 6 shows the effect of the value of θ on the accuracy. Clearly, smaller θ values tend to lead to better accuracy. This is as expected, because a larger θ means less reviews can be classified as helpful, and is therefore more restrictive, making accurate classifications more difficult.

4.4 Effects of individual factors

In our study, three factors that may affect the review helpfulness are considered, and we propose a non-linear re-

gression model to incorporate them into one model. Here, we study how the three factors affect the prediction of helpfulness individually. That is, how would the model perform if we choose to use only one of the factors for prediction? In Section 3, we discussed three models (Equations 3, 4, and 5) corresponding to the three factors. For the experiments, we train the three individual models as presented in Section 3 with the corresponding feature vectors and measure the accuracy of each one. In particular, we let $k_1 = 3$ and $k_2 = 10$ in this experiment, and the results are shown in Table 2.

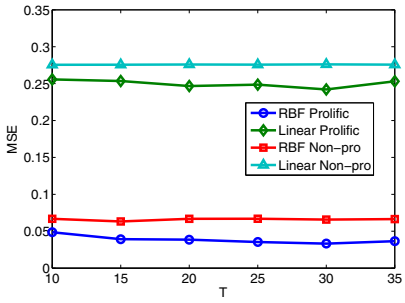
Component	MSE	F-measure
reviewer expertise	0.1912	0.5179
writing style	0.1937	0.2433
timeliness	0.1004	0.6482
all of the three	0.0332	0.7116

Table 2. Individual effects on prediction

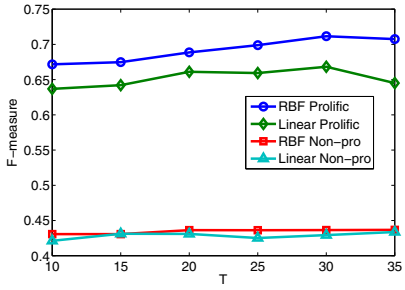
Apparently, considering the timeliness factor only yields the best results in both MSE and F-measure among the three factors. This coincides with our intuition that a timely review can be very helpful for customers to evaluate the product of interest. Only considering the writing style gives the worst performance of the three, implying that it has less predictive power compared with reviewer expertise and timeliness. Note that the results obtained by considering only one factor are not as good as considering all the factors together.

4.5 Comparison with alternative method

To demonstrate the effectiveness of our proposed model, we compare it against a baseline model that use linear regression (LR). For each review, we obtain the feature vectors $(\mathbf{x}, \mathbf{y}, t)$ corresponding to each factor in the same way as described in Section 3 and concatenate them together to form one vector $\mathbf{r}$. Then the linear regression model can be written as $\hat{H}_l = \boldsymbol{\beta}^T \mathbf{r} + b$, where β is the coefficient vector and b is the intercept. This model can be fit to the training data using standard linear least squares method. We conducted a series of experiments with different T values and compare the performance of the LR model with our proposed method. As shown in Figure 7, it is clear that our proposed method is much more accurate than the LR model for both prolific and non-prolific reviewers.



(a) MSE



(b) F-measure

Figure 7. Comparison with linear regression model

5 Related Work

5.1 Review mining

With the rapid growth of online reviews, automatic review mining has attracted a lot of research attention. Early work in this area was primarily focused on determining the semantic orientation of reviews. Among them, some of the studies attempt to learn a positive/negative classifier at the document level [16, 15], while others work at a finer level and use words as the classification subject [18].

Pushing further from the explicit two-class classification problem, Liu et al. [7] build a framework to compare consumer opinions of competing products using multiple feature dimensions. Liu et al. [11] assume that sentiment consists of multiple hidden aspects, and use a probability model to quantitatively measure the relationship between sentiment aspects and reviews.

Our method departs from classic review mining approaches in that, ultimately, we want to examine the importance of these opinions, which is a new and important research problem. In some sense, determining the sentiment and helpfulness of reviews are orthogonal to each other and could be modeled independently. One recent work that is closely related to our study attempts to examine the economic impact of the online reviews [5]. That approach mainly focuses on quantifying the extent of which the textual content, especially the subjectivity of each review, affects product sales on a market such as Amazon, while our method aims to build a more fundamental model for review helpfulness prediction. Another relevant study in this field analyzes spams that exist in online reviews [8]. In particular, their work presents a categorization of review spams, and proposes some novel strategies to detect different types

of spams. Our work can be considered complimentary to that work in that the spam filtering model can be used as a preprocessing step in our approach.

5.2 Authority and importance mining

Identifying the quality of Web documents has received a lot of attention, particularly because of its application to search engines. PageRank and HITS are two popular link-based ranking algorithms to determine the importance of web pages [10, 14]. The HITS algorithm is based on the observation that a good hub usually points to good authorities and a good authority usually points to good hubs. The Pagerank algorithm doesn't distinguish hub and authority pages. Instead, it estimates the importance of the web page's neighbours, and the authority of the page is considered proportional to this value. Motivated by this idea, various algorithms have been proposed to discover the authorities or leaders in the Web domain [9, 17, 13].

Note that our approach is different from above methods in that we use semantic information of web document rather than link structures for evaluating the helpfulness of online reviews.

6 Conclusions and Future Work

In this paper, we have considered the important problem of predicting the helpfulness of reviews. We provided a detailed analysis of the major factors affecting the helpfulness of a review, and proposed a nonlinear model based on radial basis functions for helpfulness prediction. Extensive experiments on the IMDB data set have confirmed the effectiveness of the proposed model.

Our study in this paper has focused on the movie reviews, but our approach is general enough to be easily adapted to other domains as well. For example, if we would like to handle product reviews on Amazon or CNET, we can simply replace the genres of movies with the categories of products, and the writing style and timeliness can still be modeled in the similar way as described in Section 3.

This study presents the first step in modeling the helpfulness of reviews. For future work, we plan to study the related ranking problem, i.e., how do we rank the reviews based on the helpfulness? One way to do this is to rank the reviews based on their predicted helpfulness, but we can also develop a model to directly predict the set of most helpful reviews. Another possible direction for future work is to incorporate existing votes as an indicator of the future helpfulness, and build an adaptive model which can automatically update the predication value of helpfulness as new reviews come in. Besides, we also plan to incorporate collaborative filtering methods, such as [6], to help build a personalized helpfulness prediction model.

References

[1] N. Archak, A. Ghose, and P. G. Ipeirotis. Show me the money!: deriving the pricing power of product features by mining consumer reviews. In *KDD*, pages 56–65, 2007.

[2] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *SIGGRAPH '01*, pages 67–76, 2001.

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

[4] C. Dellarocas, N. F. Awad, and X. M. Zhang. Exploring the value of online reviews to organizations: Implications for revenue forecasting and planning. In *ICIS*, pages 379–386, 2004.

[5] A. Ghose and P. G. Ipeirotis. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *ICEC*, pages 303–310, 2007.

[6] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *IJCAI*, pages 688–693, 1999.

[7] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, pages 168–177, 2004.

[8] N. Jindal and B. Liu. Opinion spam and analysis. In *WSDM*, pages 219–230, 2008.

[9] P. Jurczyk and E. Agichtein. Discovering authorities in question answer communities by using link analysis. In *CIKM*, pages 919–922, 2007.

[10] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[11] Y. Liu, X. Huang, A. An, and X. Yu. ARSA: a sentiment-aware model for predicting sales performance using blogs. In *SIGIR*, pages 607–614, 2007.

[12] Y. Liu, X. Huang, A. An, and X. Yu. Mining helpfulness of online reviews. Technical Report CSE-2008-05, Department of Computer Science and Engineering, York University, 2008.

[13] S. Nakajima, J. Tatemura, Y. Hino, Y. Hara, and K. Tanaka. Discovering important bloggers based on analyzing blog threads. In *WWW*, 2005.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[15] B. Pang and L. Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*, page 271, 2004.

[16] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *EMNLP*, 2002.

[17] X. Song, Y. Chi, K. Hino, and B. Tseng. Identifying opinion leaders in the blogosphere. In *CIKM*, pages 971–974, 2007.

[18] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL*, pages 417–424, 2001.

[19] Z. Zhang and B. Varadarajan. Utility scoring of product reviews. In *CIKM*, pages 51–57, 2006.

[20] L. Zhuang, F. Jing, and X.-Y. Zhu. Movie review mining and summarization. In *CIKM*, pages 43–50, 2006.